

Feature Addition #2: Team Features

Team features I want to add:

- Previous trad_season Ranking --DONE
- Previous trad_season win total --DONE
- Team Diff in win ranking --DONE
- Last trad_season Champion? -- DONE
- Injuries by game -- SEMI DONE? Could do more - estimate injury impact by usage
- Home or Away -- DONE
- Team Salary - DONE
- Rest Days - DONE
- Difference in Team Salary Cap year/over/year (assuming less money spent = trying to win less) - DONE

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import os
import random
import shutil
import plotly
os.chdir('C:\\\\Users\\\\Travis\\\\OneDrive\\\\Data Science\\\\Personal_Projects\\\\Sports\\\\NBA')
```

```
In [ ]: df = pd.read_csv('data/team/aggregates/Boxes_GameDifferences_and_RunningseasonAvera
```

```
In [ ]: df.head()
```

```
Out[ ]:   Unnamed: 0.3  Unnamed: 0.2  Unnamed: 0.1  Unnamed: 0  trad_unnamed: 0  trad_team  trad_matchup  trad_ga
          0           0           0       33065     29928         19      CHI    CHI @ NYK     202
          1           1           1       33050     29926         17      NYK    NYK vs. CHI     202
          2           2           2       33038     29924         15      TOR    TOR @ CLE     202
          3           3           3       33039     29937         28      PHI    PHI vs. LAC     202
          4           4           4       33040     29936         27      MIA    MIA vs. IND     202
```

5 rows × 686 columns

```
In [ ]: games_df = pd.read_csv('data/team/aggregates/both_team_boxscores_ALL.csv')
games_df.head()
```

	Unnamed: 0	trad_unnamed: 0	trad_team	trad_matchup	trad_gamedate	trad_w/l	trad_min	trad_l
0	1513	1514	LAL	LAL vs. SAC	2010-01-01	W	48.0	10
1	1512	1513	ATL	ATL vs. NYK	2010-01-01	L	53.0	10
2	1511	1512	ORL	ORL @ MIN	2010-01-01	W	48.0	10
3	1510	1511	NYK	NYK @ ATL	2010-01-01	W	53.0	11
4	1509	1510	MIN	MIN vs. ORL	2010-01-01	L	48.0	9

5 rows × 211 columns

Previous trad_season win total/ranking

```
In [ ]: games_df = games_df[games_df['trad_season'] >= 2012]
games_df = games_df[games_df['trad_season_type'] == 'Regular']
```

```
In [ ]: games_df['win?'] = np.where(games_df['trad_w/l'] == 'W', 1, 0)
```

```
In [ ]: df['win?'] = np.where(df['trad_w/l'] == 'W', 1, 0)
```

```
In [ ]: #groupby team and trad_season to get win totals per year
df_wins = games_df.groupby(['trad_team', 'trad_season'])['win?'].sum()
```

```
In [ ]: dfw = pd.DataFrame(df_wins)
dfw
```

Out[]: win?

	trad_team	trad_season	
ATL	2012	44	
	2013	38	
	2014	60	
	2015	48	
	2016	43	
WAS
	2018	32	
	2019	25	
	2020	34	
	2021	35	
	2022	13	

330 rows × 1 columns

```
In [ ]: dfw1 = dfw.loc[('ATL', 2019)]
dfw1
```

```
Out[ ]: win?    20
Name: (ATL, 2019), dtype: int32
```

```
In [ ]: def get_previous_trad_season_win_total(team, trad_season):
        return dfw.loc[(team, trad_season-1)]
```

```
In [ ]: df['team_wins_previous_season'] = df.apply(lambda row: get_previous_trad_season_win_total(row['team'], row['trad_season']), axis=1)
```

```
In [ ]: df['team_win%_previous_season'] = df['team_wins_previous_season'] / 82
```

```
In [ ]: df['team_2_wins_previous_season'] = df.apply(lambda row: get_previous_trad_season_win_total(row['team_2'], row['trad_season']), axis=1)
```

```
In [ ]: df['team2_win%_previous_season'] = df['team_2_wins_previous_season'] / 82
```

```
In [ ]: df['team1_prevwins_minus_team2_prevwins'] = df['team_wins_previous_season'] - df['team_2_wins_previous_season']
```

Home or Away

```
In [ ]: df['home_or_away'] = np.where(df['trad_matchup'].str.contains('@'), 'away', 'home')
```

```
Out[ ]:
```

	Unnamed: 0.3	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	trad_unnamed: 0	trad_team	trad_matchup	tr
0	0	0	33065	29928	19	CHI	CHI @ NYK	
1	1	1	33050	29926	17	NYK	NYK vs. CHI	
2	2	2	33038	29924	15	TOR	TOR @ CLE	
3	3	3	33039	29937	28	PHI	PHI vs. LAC	
4	4	4	33040	29936	27	MIA	MIA vs. IND	
...
16093	16093	16093	16968	18251	2460	NYK	NYK @ CLE	
16094	16094	16094	16969	18250	2459	GSW	GSW vs. SAS	
16095	16095	16095	16973	18246	2455	UTA	UTA @ POR	
16096	16096	16096	16971	18248	2457	CLE	CLE vs. NYK	
16097	16097	16097	16972	18247	2456	POR	POR vs. UTA	

16098 rows × 693 columns

NBA Champion by Year

```
In [ ]: champs = pd.read_csv('data/team/cleaned_finals_results.csv')
champs.head()
```

```
Out[ ]:
```

	Unnamed: 0	Yr	Series	Winning_team	Losing_team	Favorite	Underdog	Winner_abv	Loser_a
0	0	2021	Finals	Milwaukee Bucks	Phoenix Suns	PHO	MIL	MIL	N
1	1	2020	Finals	Los Angeles Lakers	Miami Heat	LAL	MIA	LAL	M
2	2	2019	Finals	Toronto Raptors	Golden State Warriors	GSW	TOR	TOR	G
3	3	2018	Finals	Golden State Warriors	Cleveland Cavaliers	GSW	CLE	GSW	C
4	4	2017	Finals	Golden State Warriors	Cleveland Cavaliers	GSW	CLE	GSW	C

```
In [ ]: def champ_winner(team, trad_season):
    sz_to_check = trad_season - 1
    df = champs[champs['Yr'] == sz_to_check]
    if team in df['Winner_abv'].values:
        return 1
    else:
        return 0
```

```
In [ ]: df['prev_season_champ_winner'] = df.apply(lambda row: champ_winner(row['trad_team'])
```

```
In [ ]: df['team2_prev_season_champ_winner'] = df.apply(lambda row: champ_winner(row['team_
```

```
In [ ]: df.to_csv('data/team/aggregates/All_Things.csv')
```

Team Salary

```
In [ ]: df = pd.read_csv('data/team/aggregates/All_Things.csv')
df
```

Out[]:

	Unnamed: 0.4	Unnamed: 0.3	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	trad_unnamed: 0	trad_team	trad_i
0	0	0	0	33065	29928	19	CHI	CF
1	1	1	1	33050	29926	17	NYK	NY
2	2	2	2	33038	29924	15	TOR	TC
3	3	3	3	33039	29937	28	PHI	PH
4	4	4	4	33040	29936	27	MIA	MI
...
16093	16093	16093	16093	16968	18251	2460	NYK	NY
16094	16094	16094	16094	16969	18250	2459	GSW	GSV
16095	16095	16095	16095	16973	18246	2455	UTA	UT
16096	16096	16096	16096	16971	18248	2457	CLE	CLI
16097	16097	16097	16097	16972	18247	2456	POR	POI

16098 rows × 696 columns

In []: df

Out[]:

	Unnamed: 0.4	Unnamed: 0.3	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	trad_unnamed: 0	trad_team	trad_i
0	0	0	0	33065	29928	19	CHI	CF
1	1	1	1	33050	29926	17	NYK	NY
2	2	2	2	33038	29924	15	TOR	TC
3	3	3	3	33039	29937	28	PHI	PH
4	4	4	4	33040	29936	27	MIA	MI
...
16093	16093	16093	16093	16968	18251	2460	NYK	NY
16094	16094	16094	16094	16969	18250	2459	GSW	GSV
16095	16095	16095	16095	16973	18246	2455	UTA	UT
16096	16096	16096	16096	16971	18248	2457	CLE	CLI
16097	16097	16097	16097	16972	18247	2456	POR	POI

16098 rows × 696 columns

```
In [ ]: # # append all team_contracts
# team_contracts = pd.DataFrame()
# i = 1

# for file in os.listdir('data/team/team_contracts'):
#     # Load it
#     df = pd.read_csv('data/team/team_contracts/' + file, encoding='Latin1', o
#     # drop unnamed cols
#     unnamed = df.columns[df.columns.str.contains('Unnamed')]
#     df = df.drop(unnamed, axis=1)
#     # if i=1, then make it the first df
#     if i == 1:
#         team_contracts = df
#         i = i + 1
#     else:
#         # merge it with the existing dataframe on team
#         team_contracts = team_contracts.merge(df, on='Team', how='Left')

# unnamed = team_contracts.columns[team_contracts.columns.str.contains('Unnamed')]
# team_contracts.drop(columns = unnamed, inplace=True)
# team_contracts

C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\3545278788.py:17: FutureWarning:
Passing 'suffixes' which cause duplicate columns {'2009/10(*)_y', '2003/04(*)_y', '2007/08_y', '2016/17(*)_y', '2005/06(*)_y', '2008/09(*)_y', '2004/05(*)_y', '2002/03(*)_y', '2001/02_y', '2003/04_y', '2019/20(*)_y', '2011/12(*)_y', '2021/22(*)_y', '2011/12_y', '2015/16(*)_y', '2002/03_y', '2014/15(*)_y', '2009/10_y', '2018/19(*)_y', '2020/21(*)_y', '2017/18(*)_y', '2008/09_y', '2000/01(*)_y', '2006/07_y', '2004/05_y', '2000/01_y', '2007/08(*)_y', '2012/13(*)_y', '2013/14(*)_y', '2001/02(*)_y', '2005/06_y', '2006/07(*)_y'} in the result is deprecated and will raise a MergeError in a future version.
    team_contracts = team_contracts.merge(df, on='Team', how='left')
```

Out[]:	Team	2000/01_x	2000/01(*)_x	2001/02_x	2001/02(*)_x	2002/03_x	2002/03(*)_x
0	Portland	\$87,395,140	\$150,209,627	\$84,292,850	\$140,319,647	\$105,595,657	\$173,925,253
1	New York	\$74,007,738	\$127,200,145	\$85,993,039	\$143,149,897	\$93,452,379	\$153,924,217
2	Miami	\$73,472,329	\$126,279,916	\$53,351,186	\$88,812,030	\$57,378,326	\$94,507,103
3	Brooklyn	\$68,977,578	\$118,554,603	\$75,193,712	\$125,172,596	\$60,412,919	\$99,505,342
4	Washington	\$59,085,969	\$101,553,484	\$54,776,087	\$91,184,019	\$46,691,190	\$76,904,456
5	LA Lakers	\$58,968,213	\$101,351,093	\$54,262,017	\$90,328,263	\$63,151,182	\$104,015,499
6	Milwaukee	\$57,819,318	\$99,376,441	\$56,224,266	\$93,594,760	\$57,995,152	\$95,523,069
7	San Antonio	\$57,215,938	\$98,339,386	\$45,844,814	\$76,316,410	\$53,182,559	\$87,596,308
8	Indiana	\$55,086,140	\$94,678,814	\$53,398,621	\$88,890,997	\$53,472,323	\$88,073,576
9	Phoenix	\$53,906,347	\$92,651,057	\$56,596,369	\$94,214,187	\$55,225,080	\$90,960,518
10	Utah	\$53,893,860	\$92,629,595	\$52,642,362	\$87,632,078	\$50,934,629	\$83,893,772
11	Dallas	\$52,274,842	\$89,846,921	\$57,960,080	\$96,484,312	\$72,940,316	\$120,139,056
12	Denver	\$52,189,220	\$89,699,760	\$54,460,475	\$90,658,629	\$49,109,209	\$80,887,143
13	Oklahoma City	\$51,538,658	\$88,581,612	\$45,556,733	\$75,836,850	\$53,167,479	\$87,571,471
14	Boston	\$51,478,707	\$88,478,570	\$47,515,677	\$79,097,845	\$52,548,195	\$86,551,455
15	Philadelphia	\$51,085,458	\$87,802,677	\$58,073,366	\$96,672,896	\$64,822,487	\$106,768,285
16	Cleveland	\$49,780,549	\$85,559,875	\$45,649,974	\$75,992,069	\$49,610,448	\$81,712,729
17	Houston	\$49,283,309	\$84,705,250	\$49,168,169	\$81,848,694	\$50,181,239	\$82,652,870
18	Memphis	\$48,899,840	\$84,046,166	\$50,920,631	\$84,765,963	\$61,009,172	\$100,487,421
19	Minnesota	\$47,488,500	\$81,620,437	\$55,093,782	\$91,712,876	\$59,518,801	\$98,032,650
20	Charlotte	\$46,441,558	\$79,821,014	\$49,999,445	\$83,232,496	NaN	NaN
21	Sacramento	\$46,266,289	\$79,519,774	\$54,921,012	\$91,425,274	\$70,416,596	\$115,982,270
22	Golden State	\$42,363,018	\$72,811,059	\$47,681,152	\$79,373,305	\$47,737,715	\$78,628,174
23	Detroit	\$40,475,338	\$69,566,624	\$42,417,677	\$70,611,365	\$47,777,888	\$78,694,343
24	Atlanta	\$39,297,129	\$67,541,590	\$51,525,207	\$85,772,381	\$55,711,549	\$91,761,773
25	Toronto	\$38,313,749	\$65,851,414	\$52,657,161	\$87,656,710	\$55,408,207	\$91,262,143
26	Orlando	\$37,459,860	\$64,383,801	\$45,984,001	\$76,548,112	\$50,435,346	\$83,071,410
27	Chicago	\$29,691,907	\$51,032,701	\$42,592,857	\$70,902,978	\$44,531,095	\$73,346,593
28	LA Clippers	\$29,606,189	\$50,885,371	\$33,849,932	\$56,348,909	\$42,955,777	\$70,751,903

29 rows × 431 columns

```
In [ ]: # def name_to_abbrev(name):
#         if name == 'Atlanta':
#             return 'ATL'
#         elif name == 'Boston':
#             return 'BOS'
#         elif name == 'Brooklyn':
#             return 'BKN'
#         elif name == 'Charlotte':
#             return 'CHA'
#         elif name == 'Chicago':
#             return 'CHI'
#         elif name == 'Cleveland':
#             return 'CLE'
#         elif name == 'Dallas':
#             return 'DAL'
#         elif name == 'Denver':
#             return 'DEN'
#         elif name == 'Detroit':
#             return 'DET'
#         elif name == 'Golden State':
#             return 'GSW'
#         elif name == 'Houston':
#             return 'HOU'
#         elif name == 'Indiana':
#             return 'IND'
#         elif name == 'LA Clippers':
#             return 'LAC'
#         elif name == 'LA Lakers':
#             return 'LAL'
#         elif name == 'Memphis':
#             return 'MEM'
#         elif name == 'Miami':
#             return 'MIA'
#         elif name == 'Milwaukee':
#             return 'MIL'
#         elif name == 'Minnesota':
#             return 'MIN'
#         elif name == 'New Orleans':
#             return 'NOP'
#         elif name == 'New York':
#             return 'NYK'
#         elif name == 'Oklahoma City':
#             return 'OKC'
#         elif name == 'Orlando':
#             return 'ORL'
#         elif name == 'Philadelphia':
#             return 'PHI'
#         elif name == 'Phoenix':
#             return 'PHX'
#         elif name == 'Portland':
#             return 'POR'
#         elif name == 'Sacramento':
#             return 'SAC'
#         elif name == 'San Antonio':
#             return 'SAS'
#         elif name == 'Toronto':
```

```
#         return 'TOR'
#     elif name == 'Utah':
#         return 'UTA'
#     elif name == 'Washington':
#         return 'WAS'
```

```
In [ ]: #team_contracts['Team_Abbrev'] = team_contracts.apply(Lambda row: name_to_abbrev(ro
#team_contracts
```

```
In [ ]: #team_contracts = team_contracts.rename(columns = {'2012/13': '2012', '2013/14': '2
#
#team_contracts.head(3)
```

```
In [ ]: #team_contracts.to_csv('data/team/team_contracts/All_team_contracts.csv')
```

Load Point

```
In [ ]: team_contracts = pd.read_csv('data/team/team_contracts/All_team_contracts.csv')
```

```
In [ ]: team_contracts.head(5)
```

```
Out[ ]:   Unnamed:
          0      Team  2000/01  2000/01(*)  2001/02  2001/02(*)  2002/03  2002/03(*)  2003/04  2003/04(*)
          0    Portland  $87,395,140  $150,209,627  $84,292,850  $140,319,647  $105,595,657  $173,911  $153,911  $153,911
          1    New York  $74,007,738  $127,200,145  $85,993,039  $143,149,897  $93,452,379  $153,911  $153,911  $153,911
          2      Miami  $73,472,329  $126,279,916  $53,351,186  $88,812,030  $57,378,326  $94,511  $94,511  $94,511
          3    Brooklyn  $68,977,578  $118,554,603  $75,193,712  $125,172,596  $60,412,919  $99,511  $99,511  $99,511
          4  Washington  $59,085,969  $101,553,484  $54,776,087  $91,184,019  $46,691,190  $76,911  $76,911  $76,911
```

5 rows × 390 columns

```
In [ ]: colz = list(team_contracts.columns)
colz= colz[3:]
```

```
In [ ]: colz = colz[:-1]
colz = [n for n in colz if 'yoy' not in n]
colz.remove('Team_Abbrev')
colz
```

```
Out[ ]: ['2000/01(*)',
 '2001/02',
 '2001/02(*)',
 '2002/03',
 '2002/03(*)',
 '2003/04',
 '2003/04(*)',
 '2004/05',
 '2004/05(*)',
 '2005/06',
 '2005/06(*)',
 '2006/07',
 '2006/07(*)',
 '2007/08',
 '2007/08(*)',
 '2008/09',
 '2008/09(*)',
 '2009/10',
 '2009/10(*)',
 '2011/12',
 '2011/12(*)',
 '2012',
 '2012/13(*)',
 '2013',
 '2013/14(*)',
 '2014',
 '2014/15(*)',
 '2015',
 '2015/16(*)',
 '2016',
 '2016/17(*)',
 '2017',
 '2017/18(*)',
 '2018',
 '2018/19(*)',
 '2019',
 '2019/20(*)',
 '2020',
 '2020/21(*)',
 '2021',
 '2021/22(*)',
 '2000/01_x',
 '2000/01(*)_x',
 '2001/02_x',
 '2001/02(*)_x',
 '2002/03_x',
 '2002/03(*)_x',
 '2003/04_x',
 '2003/04(*)_x',
 '2004/05_x',
 '2004/05(*)_x',
 '2005/06_x',
 '2005/06(*)_x',
 '2006/07_x',
 '2006/07(*)_x',
 '2007/08_x',
```

'2007/08(*)_x',
'2008/09_x',
'2008/09(*)_x',
'2009/10_x',
'2009/10(*)_x',
'2011/12_x',
'2011/12(*)_x',
'2012/13_x',
'2012/13(*)_x',
'2013/14_x',
'2013/14(*)_x',
'2014/15_x',
'2014/15(*)_x',
'2015/16_x',
'2015/16(*)_x',
'2016/17_x',
'2016/17(*)_x',
'2017/18_x',
'2017/18(*)_x',
'2018/19_x',
'2018/19(*)_x',
'2019/20_x',
'2019/20(*)_x',
'2020/21_x',
'2020/21(*)_x',
'2021/22_x',
'2021/22(*)_x',
'2000/01_y',
'2000/01(*)_y',
'2001/02_y',
'2001/02(*)_y',
'2002/03_y',
'2002/03(*)_y',
'2003/04_y',
'2003/04(*)_y',
'2004/05_y',
'2004/05(*)_y',
'2005/06_y',
'2005/06(*)_y',
'2006/07_y',
'2006/07(*)_y',
'2007/08_y',
'2007/08(*)_y',
'2008/09_y',
'2008/09(*)_y',
'2009/10_y',
'2009/10(*)_y',
'2011/12_y',
'2011/12(*)_y',
'2012/13_y',
'2012/13(*)_y',
'2013/14_y',
'2013/14(*)_y',
'2014/15_y',
'2014/15(*)_y',
'2015/16_y',

'2015/16(*)_y',
'2016/17_y',
'2016/17(*)_y',
'2017/18_y',
'2017/18(*)_y',
'2018/19_y',
'2018/19(*)_y',
'2019/20_y',
'2019/20(*)_y',
'2020/21_y',
'2020/21(*)_y',
'2021/22_y',
'2021/22(*)_y',
'2000/01_x.3',
'2000/01(*)_x.3',
'2001/02_x.3',
'2001/02(*)_x.3',
'2002/03_x.3',
'2002/03(*)_x.3',
'2003/04_x.3',
'2003/04(*)_x.3',
'2004/05_x.3',
'2004/05(*)_x.3',
'2005/06_x.3',
'2005/06(*)_x.3',
'2006/07_x.3',
'2006/07(*)_x.3',
'2007/08_x.3',
'2007/08(*)_x.3',
'2008/09_x.3',
'2008/09(*)_x.3',
'2009/10_x.3',
'2009/10(*)_x.3',
'2011/12_x.3',
'2011/12(*)_x.3',
'2012.4',
'2012/13(*)_x.3',
'2013.4',
'2013/14(*)_x.3',
'2014.4',
'2014/15(*)_x.3',
'2015.4',
'2015/16(*)_x.3',
'2016.4',
'2016/17(*)_x.3',
'2017.4',
'2017/18(*)_x.3',
'2018.4',
'2018/19(*)_x.3',
'2019.4',
'2019/20(*)_x.3',
'2020.4',
'2020/21(*)_x.3',
'2021.4',
'2021/22(*)_x.3',
'2000/01_y.3',

'2000/01(*)_y.3',
'2001/02_y.3',
'2001/02(*)_y.3',
'2002/03_y.3',
'2002/03(*)_y.3',
'2003/04_y.3',
'2003/04(*)_y.3',
'2004/05_y.3',
'2004/05(*)_y.3',
'2005/06_y.3',
'2005/06(*)_y.3',
'2006/07_y.3',
'2006/07(*)_y.3',
'2007/08_y.3',
'2007/08(*)_y.3',
'2008/09_y.3',
'2008/09(*)_y.3',
'2009/10_y.3',
'2009/10(*)_y.3',
'2011/12_y.3',
'2011/12(*)_y.3',
'2012/13_x.1',
'2012/13(*)_y.3',
'2013/14_x.1',
'2013/14(*)_y.3',
'2014/15_x.1',
'2014/15(*)_y.3',
'2015/16_x.1',
'2015/16(*)_y.3',
'2016/17_x.1',
'2016/17(*)_y.3',
'2017/18_x.1',
'2017/18(*)_y.3',
'2018/19_x.1',
'2018/19(*)_y.3',
'2019/20_x.1',
'2019/20(*)_y.3',
'2020/21_x.1',
'2020/21(*)_y.3',
'2021/22_x.1',
'2021/22(*)_y.3',
'2000/01_x.2',
'2000/01(*)_x.2',
'2001/02_x.2',
'2001/02(*)_x.2',
'2002/03_x.2',
'2002/03(*)_x.2',
'2003/04_x.2',
'2003/04(*)_x.2',
'2004/05_x.2',
'2004/05(*)_x.2',
'2005/06_x.2',
'2005/06(*)_x.2',
'2006/07_x.2',
'2006/07(*)_x.2',
'2007/08_x.2',

'2007/08(*)_x.2',
'2008/09_x.2',
'2008/09(*)_x.2',
'2009/10_x.2',
'2009/10(*)_x.2',
'2011/12_x.2',
'2011/12(*)_x.2',
'2012/13_y.1',
'2012/13(*)_x.2',
'2013/14_y.1',
'2013/14(*)_x.2',
'2014/15_y.1',
'2014/15(*)_x.2',
'2015/16_y.1',
'2015/16(*)_x.2',
'2016/17_y.1',
'2016/17(*)_x.2',
'2017/18_y.1',
'2017/18(*)_x.2',
'2018/19_y.1',
'2018/19(*)_x.2',
'2019/20_y.1',
'2019/20(*)_x.2',
'2020/21_y.1',
'2020/21(*)_x.2',
'2021/22_y.1',
'2021/22(*)_x.2',
'2000/01_y.2',
'2000/01(*)_y.2',
'2001/02_y.2',
'2001/02(*)_y.2',
'2002/03_y.2',
'2002/03(*)_y.2',
'2003/04_y.2',
'2003/04(*)_y.2',
'2004/05_y.2',
'2004/05(*)_y.2',
'2005/06_y.2',
'2005/06(*)_y.2',
'2006/07_y.2',
'2006/07(*)_y.2',
'2007/08_y.2',
'2007/08(*)_y.2',
'2008/09_y.2',
'2008/09(*)_y.2',
'2009/10_y.2',
'2009/10(*)_y.2',
'2011/12_y.2',
'2011/12(*)_y.2',
'2012.3',
'2012/13(*)_y.2',
'2013.3',
'2013/14(*)_y.2',
'2014.3',
'2014/15(*)_y.2',
'2015.3',

'2015/16(*)_y.2',
'2016.3',
'2016/17(*)_y.2',
'2017.3',
'2017/18(*)_y.2',
'2018.3',
'2018/19(*)_y.2',
'2019.3',
'2019/20(*)_y.2',
'2020.3',
'2020/21(*)_y.2',
'2021.3',
'2021/22(*)_y.2',
'2000/01_x.1',
'2000/01(*)_x.1',
'2001/02_x.1',
'2001/02(*)_x.1',
'2002/03_x.1',
'2002/03(*)_x.1',
'2003/04_x.1',
'2003/04(*)_x.1',
'2004/05_x.1',
'2004/05(*)_x.1',
'2005/06_x.1',
'2005/06(*)_x.1',
'2006/07_x.1',
'2006/07(*)_x.1',
'2007/08_x.1',
'2007/08(*)_x.1',
'2008/09_x.1',
'2008/09(*)_x.1',
'2009/10_x.1',
'2009/10(*)_x.1',
'2011/12_x.1',
'2011/12(*)_x.1',
'2012.2',
'2012/13(*)_x.1',
'2013.2',
'2013/14(*)_x.1',
'2014.2',
'2014/15(*)_x.1',
'2015.2',
'2015/16(*)_x.1',
'2016.2',
'2016/17(*)_x.1',
'2017.2',
'2017/18(*)_x.1',
'2018.2',
'2018/19(*)_x.1',
'2019.2',
'2019/20(*)_x.1',
'2020.2',
'2020/21(*)_x.1',
'2021.2',
'2021/22(*)_x.1',
'2000/01_y.1',

```
'2000/01(*)_y.1',
'2001/02_y.1',
'2001/02(*)_y.1',
'2002/03_y.1',
'2002/03(*)_y.1',
'2003/04_y.1',
'2003/04(*)_y.1',
'2004/05_y.1',
'2004/05(*)_y.1',
'2005/06_y.1',
'2005/06(*)_y.1',
'2006/07_y.1',
'2006/07(*)_y.1',
'2007/08_y.1',
'2007/08(*)_y.1',
'2008/09_y.1',
'2008/09(*)_y.1',
'2009/10_y.1',
'2009/10(*)_y.1',
'2011/12_y.1',
'2011/12(*)_y.1',
'2012.1',
'2012/13(*)_y.1',
'2013.1',
'2013/14(*)_y.1',
'2014.1',
'2014/15(*)_y.1',
'2015.1',
'2015/16(*)_y.1',
'2016.1',
'2016/17(*)_y.1',
'2017.1',
'2017/18(*)_y.1',
'2018.1',
'2018/19(*)_y.1',
'2019.1',
'2019/20(*)_y.1',
'2020.1',
'2020/21(*)_y.1',
'2021.1',
'2021/22(*)_y.1']
```

```
In [ ]: for c in colz:
    team_contracts[c] = team_contracts[c].astype(str).str.replace(',', ' ')
    team_contracts[c] = team_contracts[c].astype(str).str.replace('$', '')
    team_contracts[c] = team_contracts[c].astype(float)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\833158621.py:3: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.
```

```
    team_contracts[c] = team_contracts[c].astype(str).str.replace('$', '')
```

```
In [ ]: def get_salaries(team, trad_season):
    try:
        sz = str(trad_season)
        data = team_contracts[team_contracts['Team_Abbrev'] == team]
        val = data[sz].values[0]
        # get value for trad_season
        return val
    except:
        return np.nan
```

```
In [ ]: get_salaries('POR', '2019')
```

```
Out[ ]: 131979953.0
```

```
In [ ]: # compare team_abbr in team contracts to teams in df, and find the ones that don't exist in both

# get list of teams in df
teams = df['trad_team'].unique()

# get list of teams in team contracts
teams_contracts = team_contracts['Team_Abbrev'].unique()

# get list of teams that are in df but not in team contracts
teams_not_in_contracts = [x for x in teams if x not in teams_contracts]

# get list of teams that are in team contracts but not in df
teams_not_in_df = [x for x in teams_contracts if x not in teams]

print(teams_not_in_contracts)
print(teams_not_in_df)

['NOP']
[]
```

```
In [ ]: df
```

Out[]:

	Unnamed: 0.4	Unnamed: 0.3	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	trad_unnamed: 0	trad_team	trad_i
0	0	0	0	33065	29928	19	CHI	CF
1	1	1	1	33050	29926	17	NYK	NY
2	2	2	2	33038	29924	15	TOR	TC
3	3	3	3	33039	29937	28	PHI	PH
4	4	4	4	33040	29936	27	MIA	MIL
...
16093	16093	16093	16093	16968	18251	2460	NYK	NY
16094	16094	16094	16094	16969	18250	2459	GSW	GSV
16095	16095	16095	16095	16973	18246	2455	UTA	UT
16096	16096	16096	16096	16971	18248	2457	CLE	CLI
16097	16097	16097	16097	16972	18247	2456	POR	POI

16098 rows × 696 columns

In []: `df['team_salary'] = df.apply(lambda row: get_salaries(row['trad_team'], row['trad_s`

YOY CHANGE IN TEAM SALARY

In []: `team_contracts.head()`

Out[]:

	Unnamed: 0	Team	2000/01	2000/01(*)	2001/02	2001/02(*)	2002/03	2002/03(*)
0	0	Portland	\$87,395,140	150209627.0	84292850.0	140319647.0	105595657.0	1739252
1	1	New York	\$74,007,738	127200145.0	85993039.0	143149897.0	93452379.0	1539242
2	2	Miami	\$73,472,329	126279916.0	53351186.0	88812030.0	57378326.0	945071
3	3	Brooklyn	\$68,977,578	118554603.0	75193712.0	125172596.0	60412919.0	995053
4	4	Washington	\$59,085,969	101553484.0	54776087.0	91184019.0	46691190.0	769042

5 rows × 390 columns

In []: `team_contracts['yo_19_20'] = team_contracts.apply(lambda row: row['2020'] - row['2019'])`
`team_contracts['yo_20_21'] = team_contracts.apply(lambda row: row['2021'] - row['2020'])`
`team_contracts['yo_18_19'] = team_contracts.apply(lambda row: row['2019'] - row['2018'])`
`team_contracts['yo_17_18'] = team_contracts.apply(lambda row: row['2018'] - row['2017'])`
`team_contracts['yo_16_17'] = team_contracts.apply(lambda row: row['2017'] - row['2016'])`
`team_contracts['yo_15_16'] = team_contracts.apply(lambda row: row['2016'] - row['2015'])`
`team_contracts['yo_14_15'] = team_contracts.apply(lambda row: row['2015'] - row['2014'])`
`team_contracts['yo_13_14'] = team_contracts.apply(lambda row: row['2014'] - row['2013'])`
`team_contracts['yo_12_13'] = team_contracts.apply(lambda row: row['2013'] - row['2012'])`

```
In [ ]: team_contracts.head()
```

```
Out[ ]:   Unnamed:  
          0      Team  2000/01  2000/01(*)  2001/02  2001/02(*)  2002/03  2002/03  
          0      Portland $87,395,140 150209627.0 84292850.0 140319647.0 105595657.0 1739252  
          1      New York $74,007,738 127200145.0 85993039.0 143149897.0 93452379.0 1539242  
          2      Miami    $73,472,329 126279916.0 53351186.0 88812030.0 57378326.0 945071  
          3      Brooklyn $68,977,578 118554603.0 75193712.0 125172596.0 60412919.0 995053  
          4      Washington $59,085,969 101553484.0 54776087.0 91184019.0 46691190.0 769042
```

5 rows × 390 columns

```
In [ ]: team_contracts.to_csv('data/team/team_contracts/All_team_contracts.csv')
```

```
def get_yoy(team, trad_season):  
    try:  
        z = int(trad_season)  
        sz = str(trad_season)  
        sz1 = z - 1  
        sz2 = str(sz1)  
        colz = 'yoY_' + sz2[-2:] + '_' + sz[-2:]  
        data = team_contracts[team_contracts['Team_Abbrev'] == team]  
        val = data[colz].values[0]  
        # get value for trad_season  
        return val  
    except:  
        return np.nan
```

```
In [ ]: get_yoy('ATL', '2019')
```

```
Out[ ]: 31522537.0
```

```
In [ ]: df['team_yoy_salary_change'] = df.apply(lambda row: get_yoy(row['trad_team'], row['
```

```
In [ ]: df['team2_yoy_salary_change'] = df.apply(lambda row: get_yoy(row['team_2'], row['tr
```

```
In [ ]: df.to_csv('data/team/All_Things_2.csv')
```

Rest Days

```
In [ ]: def get_rest_days(team, trad_season, gamedate):  
  
    # rest days are days between game dates  
  
    data = df[(df['trad_team'] == team) & (df['trad_season'] == trad_season)]  
    data = df[df['trad_gamedate'] <= gamedate]  
    data = data.sort_values(by=['trad_gamedate'], ascending=False)  
    gameday1 = data.iloc[0]['trad_gamedate']  
    gameday1 = pd.to_datetime(gameday1)  
    gameday2 = data.iloc[2]['trad_gamedate']  
    gameday2 = pd.to_datetime(gameday2)  
    val = gameday1 - gameday2  
    val = val.days  
    # get value for trad_season  
    return val
```

```
In [ ]: get_rest_days('ATL', '2019', '2019-08-13')
```

```
Out[ ]: 3
```

```
In [ ]: df['rest_days'] = df.apply(lambda row: get_rest_days(row['trad_team'], row['trad_se
```

```
In [ ]: df['team2_rest_days'] = df.apply(lambda row: get_rest_days(row['team_2'], row['trad
```

```
In [ ]: df.to_csv('data/team/aggregates/All_Things_3.csv')
```

```
In [ ]: df
```

Out[]:

	trad_unnamed: 0	trad_team	trad_matchup	trad_gamedate	trad_w/l	trad_min	trad_pts	tra
0	19	CHI	CHI @ NYK	2022-12-23	W	48.0	118.0	
1	17	NYK	NYK vs. CHI	2022-12-23	L	48.0	117.0	
2	15	TOR	TOR @ CLE	2022-12-23	W	48.0	118.0	
3	28	PHI	PHI vs. LAC	2022-12-23	W	48.0	119.0	
4	27	MIA	MIA vs. IND	2022-12-23	L	48.0	108.0	
...
16093	2460	NYK	NYK @ CLE	2016-10-25	L	48.0	88.0	
16094	2459	GSW	GSW vs. SAS	2016-10-25	L	48.0	100.0	
16095	2455	UTA	UTA @ POR	2016-10-25	L	48.0	104.0	
16096	2457	CLE	CLE vs. NYK	2016-10-25	W	48.0	117.0	
16097	2456	POR	POR vs. UTA	2016-10-25	W	48.0	113.0	

16098 rows × 697 columns

Injuries by Game

(Originally in Features 4, for some reason)

```
In [ ]: player_trad_boxscores = pd.read_csv('data/player/aggregates/Trad&Adv_box_scores_Gam
player_trad_boxscores.head()
```

Out[]:

	Unnamed: 0	trad_unnamed: 0	trad_player	trad_team	trad_match up	trad_game date	trad_w/l	trad_min
0	7	1	Damian Lillard	POR	POR vs. LAL	10/31/2012	W	35.0
1	8	2	Kobe Bryant	LAL	LAL @ POR	10/31/2012	L	38.0
2	9	3	Nicolas Batum	POR	POR vs. LAL	10/31/2012	W	40.0
3	10	4	Al Jefferson	UTA	UTA vs. DAL	10/31/2012	W	29.0
4	11	5	Paul Millsap	UTA	UTA vs. DAL	10/31/2012	W	33.0

5 rows × 53 columns

In []:

```
# Part 1
cols = []
data = player_trad_boxscores
data['team_player'] = data['trad_team'] + ' ' + data['trad_player']
home_games = data.loc[data['trad_match up'].astype(str).str.contains('vs')]
away_games = data.loc[data['trad_match up'].astype(str).str.contains('@')]

# Fix so all games have the same format
home_games['trad_match up'] = home_games['trad_match up'].astype(str).str[-3:] + '@' + home_games['trad_match up'].astype(str).str[0:3]
print(f' The shape of the DF: {home_games.shape}')
home_games.head(2)
```

The shape of the DF: (131059, 54)

C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\3684823594.py:9: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
home_games['trad_match up'] = home_games['trad_match up'].astype(str).str[-3:] + '@' + home_games['trad_match up'].astype(str).str[0:3]
```

Out[]:

	Unnamed: 0	trad_unnamed: 0	trad_player	trad_team	trad_match up	trad_game date	trad_w/l	trad_min
0	7	1	Damian Lillard	POR	LAL @ POR	10/31/2012	W	35.0
2	9	3	Nicolas Batum	POR	LAL @ POR	10/31/2012	W	40.0

2 rows × 54 columns

In []:

```
data2 = pd.concat([home_games, away_games])
# Add matchup_gamedate to data
data2['Matchup_GameDate'] = data2['trad_match up'].astype(str) + ' ' + data2['trad_'
data2.head(2)
```

	Unnamed: 0	trad_unnamed: 0	trad_player	trad_team	trad_match up	trad_game date	trad_w/l	trad_min
0	7	1	Damian Lillard	POR	LAL @ POR	10/31/2012	W	35.0
2	9	3	Nicolas Batum	POR	LAL @ POR	10/31/2012	W	40.0

2 rows × 55 columns

```
In [ ]: # Part 2
# Get unique player names
player_team_namez = data2['team_player'].unique()

# Add players as columns
for player_team in player_team_namez:
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['Min'], 0)
    cols2 = str(player_team) + '_min'
    cols.append(cols2)
grouped = data2.groupby('Matchup_GameDate')[cols].sum()
grouped.to_csv('data/player/aggregates/players_minutes_played_GBG.csv')

grouped
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:7: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_played'] = np.where( data2['team_player'] == str(player_team), 1, 0)
C:\Users\Travis\AppData\Local\Temp\ipykernel_8124\1054112233.py:8: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use `newframe = frame.copy()`
    data2[str(player_team) + '_min'] = np.where( data2['team_player'] == str(player_team), data2['trad_min'], 0)
```

Out[]:

	POR_Damian Lillard_min	POR_Nicolas Batum_min	UTA_AI Jefferson_min	UTA_Paul Millsap_min	UTA_Marvin Williams_min	UTA_Ra Foye_
Matchup_GameDate						
ATL @ BKN_01_01_2021	0.0	0.0	0.0	0.0	0.0	0.0
ATL @ BKN_01_06_2014	0.0	0.0	0.0	0.0	0.0	0.0
ATL @ BKN_01_09_2019	0.0	0.0	0.0	0.0	0.0	0.0
ATL @ BKN_01_10_2017	0.0	0.0	0.0	0.0	0.0	0.0
ATL @ BKN_01_12_2020	0.0	0.0	0.0	0.0	0.0	0.0
...
WAS @ UTA_03_31_2017	0.0	0.0	0.0	0.0	0.0	0.0
WAS @ UTA_04_12_2021	0.0	0.0	0.0	0.0	0.0	0.0
WAS @ UTA_12_04_2017	0.0	0.0	0.0	0.0	0.0	0.0
WAS @ UTA_12_18_2021	0.0	0.0	0.0	0.0	0.0	0.0
WAS @ UTA_12_22_2022	0.0	0.0	0.0	0.0	0.0	0.0

12382 rows × 3493 columns

```
In [ ]: grouped = grouped.reset_index()
grouped = grouped.loc[:,~grouped.columns.duplicated()].copy()

In [ ]: grouped.to_csv('data/player/aggregates/players_minutes_played_GBG.csv') # This is t

In [ ]: # define a function to grab it
# do it by output that trad_season by that player?
minutes_df = pd.read_csv('data/player/aggregates/players_minutes_played_GBG.csv')
minutes_df.head()
```

Out[]:

	Unnamed: 0	Matchup_GameDate	POR_Damian Lillard_min	POR_Nicolas Batum_min	UTA_AI Jefferson_min	UTA_Paul Millsap_min	UTA_N Willian
0	0	ATL @ BKN_01_01_2021	0.0	0.0	0.0	0.0	0.0
1	1	ATL @ BKN_01_06_2014	0.0	0.0	0.0	0.0	0.0
2	2	ATL @ BKN_01_09_2019	0.0	0.0	0.0	0.0	0.0
3	3	ATL @ BKN_01_10_2017	0.0	0.0	0.0	0.0	0.0
4	4	ATL @ BKN_01_12_2020	0.0	0.0	0.0	0.0	0.0

5 rows × 3495 columns

```
In [ ]: to_drop = [c for c in df.columns if 'Unnamed' in c]
to_drop

Out[ ]: []

In [ ]: df = df.drop(columns = to_drop, axis=1)

In [ ]: df['Matchup_GameDate'] = df['trad_matchup'] + '_' + df['trad_gamedate']

In [ ]: df['Matchup_GameDate'] = df['Matchup_GameDate'].str.replace('-', '_')

In [ ]: df.head()
```

Out[]: trad_unnamed:

	trad_team	trad_matchup	trad_gamedate	trad_w/l	trad_min	trad_pts	trad_fg%
0	CHI	CHI @ NYK	2022-12-23	W	48.0	118.0	45
1	NYK	NYK vs. CHI	2022-12-23	L	48.0	117.0	43
2	TOR	TOR @ CLE	2022-12-23	W	48.0	118.0	42
3	PHI	PHI vs. LAC	2022-12-23	W	48.0	119.0	47
4	MIA	MIA vs. IND	2022-12-23	L	48.0	108.0	41

5 rows × 697 columns

```
In [ ]: def get_matchup_gamedate(matchup, gamedate):
    if 'vs.' in matchup:
        matchup = matchup.replace('vs.', ' @ ')
        matchup = matchup.replace('-', '_')
        gamedate = gamedate.replace('-', '_')
        tm1 = matchup.split(' @ ')[0]
        tm2 = matchup.split(' @ ')[1]
        tm2 = tm2.replace(' ', '')
        tm1 = tm1.replace(' ', '')
        new_matchup = tm2 + ' @ ' + tm1 + '_' + gamedate
    else:
        gamedate = gamedate.replace('-', '_')
        new_matchup = matchup + '_' + gamedate
    return new_matchup
```

```
In [ ]: get_matchup_gamedate('GSW vs. BOS', '2022-06-13')
```

```
Out[ ]: 'BOS @ GSW_2022_06_13'
```

```
In [ ]: df['Matchup_GameDate'] = df.apply(lambda row: get_matchup_gamedate(row['trad_matchup'], row['trad_gamedate']), axis=1)
```

```
In [ ]: df
```

Out[]:

trad_unnamed: 0	trad_team	trad_matchup	trad_gamedate	trad_w/l	trad_min	trad_pts	tra
0	19	CHI	CHI @ NYK	2022-12-23	W	48.0	118.0
1	17	NYK	NYK vs. CHI	2022-12-23	L	48.0	117.0
2	15	TOR	TOR @ CLE	2022-12-23	W	48.0	118.0
3	28	PHI	PHI vs. LAC	2022-12-23	W	48.0	119.0
4	27	MIA	MIA vs. IND	2022-12-23	L	48.0	108.0
...
16093	2460	NYK	NYK @ CLE	2016-10-25	L	48.0	88.0
16094	2459	GSW	GSW vs. SAS	2016-10-25	L	48.0	100.0
16095	2455	UTA	UTA @ POR	2016-10-25	L	48.0	104.0
16096	2457	CLE	CLE vs. NYK	2016-10-25	W	48.0	117.0
16097	2456	POR	POR vs. UTA	2016-10-25	W	48.0	113.0

16098 rows × 697 columns

In []:

```
# add minutes by game to df
df2 = df.merge(minutes_df, on='Matchup_GameDate', how='left')
print(df2.shape)
df2.head()
```

(16098, 4191)

Out[]:

trad_unnamed: 0	trad_team	trad_matchup	trad_gamedate	trad_w/l	trad_min	trad_pts	trad_fg%
0	19	CHI	CHI @ NYK	2022-12-23	W	48.0	118.0
1	17	NYK	NYK vs. CHI	2022-12-23	L	48.0	117.0
2	15	TOR	TOR @ CLE	2022-12-23	W	48.0	118.0
3	28	PHI	PHI vs. LAC	2022-12-23	W	48.0	119.0
4	27	MIA	MIA vs. IND	2022-12-23	L	48.0	108.0

5 rows × 4191 columns

```
In [ ]: df2.to_csv('data/team/aggregates/Team_Box_Data_With_Player_Minutes.csv')
```

```
In [ ]:
```