In [1]:
```python
import numpy as np
import pandas as pd
import json
import ast
import re
```

In [2]:
```python
from nba_api.stats.static import players
from nba_api.stats.static import teams
from nba_api.stats.endpoints import playergamelog, teamgamelog
from nba_api.stats.endpoints import BoxScoreDefensive
from nba_api.stats.library import data
from nba_api.stats.endpoints import commonallplayers
from nba_api.stats.endpoints import boxscorematchups

pd.set_option('display.max_columns', None)
```

In [3]:
```python
player_dict = players.get_players()
```

In [5]:
```python
steph = [player for player in player_dict if player['full_name'] == 'Stephen Curry'][(
```

In [6]:
```python
steph
```

Out[6]:
```
{'id': 201939,
 'full_name': 'Stephen Curry',
 'first_name': 'Stephen',
 'last_name': 'Curry',
 'is_active': True}
```

In [7]:
```python
steph_id = steph['id']
```

In [8]:
```python
team_dict = teams.get_teams()
```

In [10]:
```python
gsw = [team for team in team_dict if team['abbreviation'] == "GSW"][0]
```

In [11]:
```python
gsw_id = gsw['id']
```

In [13]:
```python
warriors_gl = teamgamelog.TeamGameLog(gsw_id)
```

In [14]:
```python
warriors_20_df = warriors_gl.get_data_frames()[0]
```

In [15]:
```python
warriors_20_df[warriors_20_df['MATCHUP'].str.contains('SAS')]
```

Out[15]:

	Team_ID	Game_ID	GAME_DATE	MATCHUP	WL	W	L	W_PCT	MIN	FGM	FGA	FG_PCT	FG3M
10	1610612744	0022000378	FEB 09, 2021	GSW @ SAS	W	13	12	0.520	240	42	91	0.462	17
11	1610612744	0022000371	FEB 08, 2021	GSW @ SAS	L	12	12	0.500	240	38	86	0.442	13
21	1610612744	0022000223	JAN 20, 2021	GSW vs. SAS	W	8	6	0.571	240	46	91	0.505	15

In [16]:
```python
gamelog_steph = playergamelog.PlayerGameLog(steph_id)
```

In [17]:
```python
gamelog_steph_df = gamelog_steph.get_data_frames()[0]
```

In [18]:
```python
gamelog_steph_df.head(2)
```

Out[18]:

	SEASON_ID	Player_ID	Game_ID	GAME_DATE	MATCHUP	WL	MIN	FGM	FGA	FG_PCT	FG3M	FG3A
0	22020	201939	0022000527	FEB 28, 2021	GSW @ LAL	L	26	5	13	0.385	2	7
1	22020	201939	0022000511	FEB 26, 2021	GSW vs. CHA	W	36	8	15	0.533	3	8

In [19]:
```python
gamelog_steph_df['PAR'] = gamelog_steph_df['PTS'] + gamelog_steph_df['AST'] + gamelog_
```

In [20]:
```python
gamelog_steph_df.head(2)
```

Out[20]:

	SEASON_ID	Player_ID	Game_ID	GAME_DATE	MATCHUP	WL	MIN	FGM	FGA	FG_PCT	FG3M	FG3A
0	22020	201939	0022000527	FEB 28, 2021	GSW @ LAL	L	26	5	13	0.385	2	7
1	22020	201939	0022000511	FEB 26, 2021	GSW vs. CHA	W	36	8	15	0.533	3	8

In [21]:
```python
gamelog_steph_df['PAR'].mean()
```

Out[21]:
```
41.294117647058826
```

In [22]:
```python
brogdon = [player for player in player_dict if player['full_name'] == 'Malcolm Brogdon
```

In [23]:
```python
brogdon_id = brogdon['id']
```

In [24]:
```python
gamelog_brogdon_20 = playergamelog.PlayerGameLog(brogdon_id, '2020')
gamelog_brogdon_19 = playergamelog.PlayerGameLog(brogdon_id, '2019')
gamelog_brogdon_18 = playergamelog.PlayerGameLog(brogdon_id, '2018')
gamelog_brogdon_17 = playergamelog.PlayerGameLog(brogdon_id, '2017')
```

In [25]:
```python
gamelog_brogdon_20_df = gamelog_brogdon_20.get_data_frames()[0]
gamelog_brogdon_19_df = gamelog_brogdon_19.get_data_frames()[0]
gamelog_brogdon_18_df = gamelog_brogdon_18.get_data_frames()[0]
gamelog_brogdon_17_df = gamelog_brogdon_17.get_data_frames()[0]
```

In [26]:
```python
brog_17to20_df = pd.concat([gamelog_brogdon_17_df, gamelog_brogdon_18_df, gamelog_broc
```

In [27]:
```python
brog_17to20_df['GAME_DATE'] = pd.to_datetime(brog_17to20_df['GAME_DATE'])
```

In [28]:
```python
brog_17to20_df.sort_values(by='GAME_DATE', ascending=False,inplace=True)
```

In [29]:
```python
brog_17to20_df.head(2)
```

Out[29]:

	SEASON_ID	Player_ID	Game_ID	GAME_DATE	MATCHUP	WL	MIN	FGM	FGA	FG_PCT	FG3M	FG3A
0	22020	1627763	0022000530	2021-03-01	IND @ PHI	L	29	9	17	0.529	0	3
1	22020	1627763	0022000505	2021-02-26	IND @ BOS	L	33	5	17	0.294	2	7

In [30]:
```python
brog_17to20_df['PAR'] = brog_17to20_df['PTS'] + brog_17to20_df['AST'] + brog_17to20_di
```

In [31]:
```python
brog_17to20_df.head(2)
```

Out[31]:

	SEASON_ID	Player_ID	Game_ID	GAME_DATE	MATCHUP	WL	MIN	FGM	FGA	FG_PCT	FG3M	FG3A
0	22020	1627763	0022000530	2021-03-01	IND @ PHI	L	29	9	17	0.529	0	3
1	22020	1627763	0022000505	2021-02-26	IND @ BOS	L	33	5	17	0.294	2	7

In [32]:
```python
brog_vs_bkn = brog_17to20_df[brog_17to20_df['MATCHUP'].str.contains('BKN')]
```

In [33]:
```python
li_brog_vs_bkn = brog_vs_bkn['Game_ID'].to_list()
```

In [36]:
```python
def oppFinder(row):
    return row.split()[-1]
```

In [37]:
```python
oppFinder('IND vs. UTA')
```

Out[37]:
```
'UTA'
```

In [38]:
```python
brog_17to20_df['OPPONENT'] = brog_17to20_df['MATCHUP'].apply(oppFinder)
```

In [40]:
```python
brog_17to20_df.groupby('OPPONENT').agg(
    AVG_PLUSMINUS = ('PLUS_MINUS', 'mean'),
    AVG_REB = ('REB', 'mean'),
    AVG_STL = ('STL', 'mean'),
    AVG_PTS = ('PTS', 'mean'),
    AVG_AST = ('AST', 'mean'),
    AVG_PAR = ('PAR', 'mean')
 ).round(2)
```

Out[40]:

OPPONENT	AVG_PLUSMINUS	AVG_REB	AVG_STL	AVG_PTS	AVG_AST	AVG_PAR
ATL	5.83	4.67	1.00	17.00	6.17	27.83
BKN	5.00	6.20	1.20	15.20	5.00	26.40
BOS	1.00	4.60	0.50	17.00	4.80	26.40
CHA	2.75	4.08	0.75	17.08	5.42	26.58
CHI	5.80	5.40	0.60	17.80	4.80	28.00
CLE	4.50	3.75	1.12	20.75	5.25	29.75
DAL	-3.33	3.83	1.50	15.83	3.33	23.00
DEN	5.00	5.67	0.67	18.33	5.33	29.33
DET	3.60	5.40	0.90	15.70	5.70	26.80
GSW	-5.60	3.40	1.40	17.80	3.80	25.00
HOU	7.50	3.00	1.25	20.25	4.00	27.25
IND	9.33	3.83	0.67	12.83	3.83	20.50
LAC	-14.00	2.00	1.00	19.67	4.33	26.00
LAL	4.25	3.75	1.00	18.75	4.75	27.25
MEM	3.00	4.33	0.33	13.83	4.83	23.00
MIA	-3.67	4.00	0.50	11.17	2.50	17.67
MIL	-12.50	3.25	1.00	10.75	7.00	21.00
MIN	12.00	4.43	0.43	15.86	4.71	25.00
NOP	-2.17	4.50	1.33	16.00	4.67	25.17
NYK	-1.25	5.62	1.25	17.25	5.75	28.62
OKC	4.00	3.80	1.00	12.00	4.40	20.20
ORL	9.38	4.12	0.25	14.75	4.38	23.25
PHI	-5.50	3.62	0.38	15.00	4.25	22.88
PHX	-1.29	4.14	1.00	19.86	3.86	27.86
POR	6.17	3.17	1.33	14.17	5.17	22.50
SAC	0.17	4.33	0.33	16.17	5.00	25.50
SAS	1.75	4.25	0.50	15.75	3.25	23.25
TOR	2.73	4.82	0.73	17.45	4.55	26.82
UTA	-4.67	3.83	0.50	16.33	4.83	25.00
WAS	10.62	5.00	0.75	13.25	5.38	23.62

In [43]:
```python
with open('json_nba.json') as f:
    data = json.load(f)
```

In [44]:
```python
json_data = ast.literal_eval(data)
```

In [45]:
```python
json = json_data[0]
```

In [48]:
```python
events = json['events'][0]
```

In [50]:
```python
df = pd.json_normalize(events, sep='_')
```

In [51]:
```python
df
```

Out[51]:

	id	description	type	link	status	sport	startTime	live	awayTeamFir
0	8301268	Golden State Warriors @ San Antonio Spurs	GAMEEVENT	/basketball/nba/golden-state-warriors-san-anto...	U	BASK	1612920900000	False	Tru

In [53]:
```python
competitors = pd.json_normalize(df['competitors'][0])
#competitors of the current game matchup
```

In [54]:
```python
competitors
```

Out[54]:

	id	name	home
0	8301268-204	San Antonio Spurs	True
1	8301268-11757531	Golden State Warriors	False

In [55]:
```python
displayGroups = pd.json_normalize(df['displayGroups'][0])
#types of betting (lines, game&player props, periods)
```

In [56]:
```python
displayGroups
```

Out[56]:

	id	description	defaultType	alternateType	markets	order
0	100-97	Game Lines	True	False	[{'id': '144048853', 'descriptionKey': 'Head T...	1
1	100-101	Period/Alternate Lines	False	False	[{'id': 'G-2W-HCAP.Handicap - Asian.209', 'des...	4
2	100-104	Game Props	False	False	[{'id': '144048848', 'descriptionKey': 'First ...	6
3	100-105	Player Props	False	False	[{'id': '144144463', 'descriptionKey': 'Will {...	7

In [58]:
```python
player_props = pd.json_normalize(displayGroups[displayGroups['description'] == 'Playe:
```

In [60]:
```python
player_props.drop(['status','singleOnly','notes', 'period.abbreviation','period.live']
```

In [61]:
```python
player_props.head(2)
```

Out[61]:

	id	descriptionKey	description	key	marketTypeId	outcomes	period.id	period.description	p
0	144144463	Will {Dejounte Murray (SAS)} record a Double-D...	Will Dejounte Murray (SAS) record a Double-Dou...	GAME-PROP-12	-1093	[{'id': '773861571', 'description': 'Yes', 'st...	209	Game	
1	144144458	Will {DeMar DeRozan (SAS)} record a Double-Dou...	Will DeMar DeRozan (SAS) record a Double-Double?	GAME-PROP-12	-1093	[{'id': '773861561', 'description': 'Yes', 'st...	209	Game	

In [64]:
```python
pd.json_normalize(player_props['outcomes'][0])
```

Out[64]:

	id	description	status	type	price.id	price.american	price.decimal	price.fractional	price.malay
0	773861571	Yes	O	X	7077383005	+235	3.350	47/20	-0.43
1	773861572	No	O	X	7077383006	-330	1.30303	10/33	0.30

In [65]:
```python
player_props.head(1)
```

Out[65]:

	id	descriptionKey	description	key	marketTypeId	outcomes	period.id	period.description	p
0	144144463	Will {Dejounte Murray (SAS)} record a Double-D...	Will Dejounte Murray (SAS) record a Double-Dou...	GAME-PROP-12	-1093	[{'id': '773861571', 'description': 'Yes', 'st...	209	Game	

In [67]:
```python
player_props['outcome_ids'] = player_props['outcomes'].apply(lambda x: "|".join(i['id
```

In [68]:
```python
player_props['player_prop_odds'] = player_props['outcomes'].apply(lambda x: "|".join(
```

In [70]:
```python
player_props['yes'] = player_props['player_prop_odds'].apply(lambda x: x.split('|')[0]
player_props['no'] = player_props['player_prop_odds'].apply(lambda x: x.split('|')[1])
```

In [72]:
```python
player_props['outcome_yes_id'] = player_props['outcome_ids'].apply(lambda x: x.split(
player_props['outcome_no_id'] = player_props['outcome_ids'].apply(lambda x: x.split('
```

In [74]:
```python
player_props.drop(['outcome_ids', 'player_prop_odds'],axis=1,inplace=True)
```

In [77]:
```python
dejounte = [player for player in player_dict if player['full_name'] == 'Dejounte Murra
demar = [player for player in player_dict if player['full_name'] == 'DeMar DeRozan'][(
```

In [78]:
```python
player_props.iloc[25]['outcomes'][1]
```

Out[78]:
```
{'id': '773861620',
 'description': 'Under',
 'status': 'O',
 'type': 'U',
 'competitorId': '8301268-11766057',
 'price': {'id': '7077806507',
  'handicap': '42.5',
  'american': '-110',
  'decimal': '1.909091',
  'fractional': '10/11',
  'malay': '0.91',
  'indonesian': '-1.10',
  'hongkong': '0.91'}}
```

In [82]:
```python
def isHandicap(row):
    for i in row:
        try:
            return i['price']['handicap']
        except (KeyError):
            return np.nan
```

In [83]:
```python
player_props['handicap'] = player_props['outcomes'].apply(isHandicap)
```

In [84]:
```python
player_props['player_prop_odds'] = player_props['outcomes'].apply(lambda x: "|".join(:
```

In [86]:
```python
player_props.head(1)
```

Out[86]:

	id	descriptionKey	description	key	marketTypeId	outcomes	period.id	period.description	p
0	144144463	Will {Dejounte Murray (SAS)} record a Double-D...	Will Dejounte Murray (SAS) record a Double-Dou...	GAME-PROP-12	-1093	[{'id': '773861571', 'description': 'Yes', 'st...	209	Game	

In [87]:
```python
player_props['player_name'] = player_props['descriptionKey'].apply(lambda s: s.split(
```

In [88]:
```python
player_props['player_team'] = player_props['descriptionKey'].apply(lambda s: s.split(
lambda s: s.split('(',1)[1].split(')')[0])
```

In [90]:
```python
unique_team_names = list(player_props['player_team'].unique())
```

In [91]:
```python
unique_team_names
```

Out[91]:
```
['SAS', 'GSW']
```

In [92]:
```python
def getOpponent(row):
    for i in unique_team_names:
        if i == row:
            return unique_team_names[-1]
        else:
            return unique_team_names[0]
```

In [93]:
```python
player_props['opp_team'] = player_props['player_team'].apply(getOpponent)
```

In [94]:
```python
player_props.iloc[0]['descriptionKey']
```

Out[94]:
```
'Will {Dejounte Murray (SAS)} record a Double-Double?'
```

In [95]:
```python
player_props.iloc[-7]['descriptionKey'].split(' - {')
```

Out[95]:
```
['Total Points, Rebounds and Assists', 'Stephen Curry (GSW)} (NBA)']
```

In [96]:
```python
gsw_sas = teamgamelog.TeamGameLog(gsw_id)
```

In [97]:
```python
gsw_sas = gsw_sas.get_data_frames()[0]
```

In [98]:
```python
gsw_sas.head()
```

Out[98]:

	Team_ID	Game_ID	GAME_DATE	MATCHUP	WL	W	L	W_PCT	MIN	FGM	FGA	FG_PCT	FG3M
0	1610612744	0022000527	FEB 28, 2021	GSW @ LAL	L	19	16	0.543	240	35	86	0.407	8
1	1610612744	0022000511	FEB 26, 2021	GSW vs. CHA	W	19	15	0.559	240	48	79	0.608	11
2	1610612744	0022000489	FEB 24, 2021	GSW @ IND	W	18	15	0.545	240	39	83	0.470	5
3	1610612744	0022000483	FEB 23, 2021	GSW @ NYK	W	17	15	0.531	240	40	89	0.449	13
4	1610612744	0022000459	FEB 20, 2021	GSW @ CHA	L	16	15	0.516	240	37	85	0.435	13

In [ ]:

In [101...
```python
game_datetime = pd.to_datetime(re.findall(r'\d+', df['link'][0])[0][:8])
```

In [102...
```python
game_datetime
```

Out[102...
```
Timestamp('2021-02-09 00:00:00')
```

In [103...
```python
player_props['GAME_DATE'] = game_datetime
```

In [104...
```python
player_props.head(1)
```

Out[104...

	id	descriptionKey	description	key	marketTypeId	outcomes	period.id	period.description	p
0	144144463	Will {Dejounte Murray (SAS)} record a Double-D...	Will Dejounte Murray (SAS) record a Double-Dou...	GAME-PROP-12	-1093	[{'id': '773861571', 'description': 'Yes', 'st...	209	Game	

In [105...
```python
player_props['MATCHUP'] = player_props['player_team'] + ' @ ' + player_props['opp_tear
```

In [107...
```python
prop_action = ['Double-Double, Total Points', 'Total Rebounds and Assists',
               'Total Points, Rebounds and Assists', 'Total Made 3 Point Shots']
```

for i in player_props['descriptionKey']: if i in prop_action: print(prop_action) else: print('not here')

In [108...
```python
def getPlayerStats(player_name):
    gets_player_name = [player for player in player_dict if player['full_name'] == pla
    player_name_id = gets_player_name['id']
    df_player = playergamelog.PlayerGameLog(player_name_id).get_data_frames()[0]
    df_player['PAR'] = df_player['PTS'] + df_player['AST'] + df_player['REB']
    df_player['PA'] = df_player['PTS'] + df_player['AST']
    return df_player
```

In [110...
```python
players_df = pd.DataFrame(data.players)
```

In [111...
```python
players_df.columns = ['player_id','player_lastname','player_first_name','player_fullna
```

In [112...
```python
active_players_df = players_df[players_df['player_active'] == True]
```

In [114...
```python
teams_df = pd.DataFrame(data.teams)
```

In [115...
```python
teams_df.columns = ['team_id','team_abb','team_nickname','team_yr','team_city','team_
```

In [116...
```python
teams_df.drop('team_yr',inplace=True,axis=1)
```

In [119...
```python
teams_df[teams_df['team_abb'] == 'DAL']['team_id']
```

Out[119...
```
5    1610612742
Name: team_id, dtype: int64
```

In [120...
```python
def get_teams_id(teams_abb):
    teams_list = []
    for team in teams_abb:
        teams_list.append(teams_df[teams_df['team_abb'] == team]['team_id'].item())
    return teams_list

def get_teams_abb(teams_id):
    teams_list = []
    for team in teams_id:
        teams_list.append(teams_df[teams_df['team_id'] == team]['team_abb'].item())
    return teams_list
```

In [121...
```python
get_teams_id(['DAL', 'NOP'])
```

Out[121...
```
[1610612742, 1610612740]
```

In [122...
```python
get_teams_abb([1610612742, 1610612740])
```

Out[122...
```
['DAL', 'NOP']
```

In [123...
```python
df1 = teams_df[teams_df['team_abb'] == 'BOS']
df2 = teams_df[teams_df['team_abb'] == 'NOP']
```

In [124...
```python
pd.concat([df1,df2])
```

Out[124...

	team_id	team_abb	team_nickname	team_city	team_fullname	team_state
1	1610612738	BOS	Celtics	Boston	Boston Celtics	Massachusetts
3	1610612740	NOP	Pelicans	New Orleans	New Orleans Pelicans	Louisiana

In [125...
```python
def getTeamRecentGames(team_abb):
    gets_team_name = [team for team in team_dict if team['abbreviation'] == team_abb]
    team_name_id = gets_team_name['id']
    df_team = teamgamelog.TeamGameLog(team_name_id).get_data_frames()[0]
    return df_team
```

In [126...
```python
getTeamRecentGames('NOP').head(2)
```

Out[126...

	Team_ID	Game_ID	GAME_DATE	MATCHUP	WL	W	L	W_PCT	MIN	FGM	FGA	FG_PCT	FG3M
0	1610612740	0022000533	MAR 01, 2021	NOP vs. UTA	W	15	19	0.441	240	52	92	0.565	7
1	1610612740	0022000514	FEB 27, 2021	NOP @ SAS	L	14	19	0.424	240	40	91	0.440	9

In [127...
```python
getTeamRecentGames('DAL').head(2)
```

Out[127...

	Team_ID	Game_ID	GAME_DATE	MATCHUP	WL	W	L	W_PCT	MIN	FGM	FGA	FG_PCT	FG3M
0	1610612742	0022000529	MAR 01, 2021	DAL @ ORL	W	17	16	0.515	240	47	90	0.522	17
1	1610612742	0022000519	FEB 27, 2021	DAL @ BKN	W	16	16	0.500	240	45	86	0.523	14

In [2]:
```python
!jt -r
```

```
Reset css and font defaults in:
C:\Users\Shayan\.jupyter\custom &
C:\Users\Shayan\AppData\Roaming\jupyter\nbextensions
```

In [ ]:

In [ ]: