

Game Features - Odds and Refs

To get the odds and refs, I had to purchase some data from bigdataball.com.

I hypothetically could have scraped the information from <https://www.scoresandodds.com/nba>, but, as the project has already gone many iterations over my initial time budget, I wanted to save some time.

This data is in 'data/team/bigdataball_odds/2018-22 NBA Box Score Team Stats w. Odds.xlsx'

This data should be messed with in Excel, just because that's how I have been doing it.

```
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\\\\bigdataball\\\\2018-2022_BDB_Boxes_with_Odds.csv')
df.head(2)
```

```
Out[ ]:      DATASET    GAME-ID      DATE      TEAM VENUE 1Q 2Q 3Q 4Q OT1 ... BOX SCORE\\nURL
0   NBA 2018-2019 Regular Season 21800001 10/16/2018 Philadelphia R 21 21 24 21 NaN ... NaN
1   NBA 2018-2019 Regular Season 21800001 10/16/2018 Boston H 21 26 30 28 NaN ... BOX SCORE
```

2 rows × 65 columns

```
In [ ]: # load this seasons BDB data
df2 = pd.read_csv('data/team/aggregates/bigdataball_team_boxes.csv')
df2.head(2)
```

```
Out[ ]:
```

	BIGDATABALL\nDATASET	GAME-ID	DATE	TEAM	VENUE	1Q	2Q	3Q	4Q	OT1	..
0	NBA 2022-2023 Regular Season	22200001	10/18/2022	Philadelphia	R	29	34	25	29	NaN	..
1	NBA 2022-2023 Regular Season	22200001	10/18/2022	Boston	H	24	39	35	28	NaN	..

2 rows × 65 columns

```
In [ ]: df.columns
```

```
Out[ ]: Index(['DATASET', 'GAME-ID', 'DATE', 'TEAM', 'VENUE', '1Q', '2Q', '3Q', '4Q',
       'OT1', 'OT2', 'OT3', 'OT4', 'OT5', 'F', 'MIN', 'FG', 'FGA', '3P', '3PA',
       'FT', 'FTA', 'OR', 'DR', 'TOT', 'A', 'PF', 'ST', 'TO', 'TO\nTO', 'BL',
       'PTS', 'POSS', 'PACE', 'OEFF', 'DEFF', 'TEAM\nREST DAYS',
       'STARTING LINEUPS', 'Unnamed: 38', 'Unnamed: 39', 'Unnamed: 40',
       'Unnamed: 41', 'MAIN REF', 'CREW', 'OPENING ODDS', 'OPENING SPREAD',
       'OPENING TOTAL', 'LINE \nMOVEMENT #1', 'LINE \nMOVEMENT #2',
       'LINE \nMOVEMENT #3', 'CLOSING\nODDS', 'CLOSING SPREAD',
       'CLOSING TOTAL', 'MONEYLINE', 'HALFTIME', 'BOX SCORE\nURL', 'ODDS\nURL',
       'Team_Abbrev', 'Nba_com_team_Abbrev', 'Opp_abbr',
       'Nba_com_team_Abbrev.1', 'Date_underscore', 'HomeTeam', 'AwayTeam',
       'Matchup_GameDate'],
      dtype='object')
```

```
In [ ]: df2.columns
```

```
Out[ ]: Index(['BIGDATABALL\nDATASET', 'GAME-ID', 'DATE', 'TEAM', 'VENUE', '1Q', '2Q',
       '3Q', '4Q', 'OT1', 'OT2', 'OT3', 'OT4', 'OT5', 'F', 'MIN', 'FG', 'FGA',
       '3P', '3PA', 'FT', 'FTA', 'OR', 'DR', 'TOT', 'A', 'PF', 'ST', 'TO',
       'TO\nTO', 'BL', 'PTS', 'POSS', 'PACE', 'OEFF', 'DEFF',
       'TEAM\nREST DAYS', 'STARTING LINEUPS', 'Unnamed: 38', 'Unnamed: 39',
       'Unnamed: 40', 'Unnamed: 41', 'CREW CHIEF', 'REFEREE\n&\nUMPIRE',
       'OPENING ODDS', 'OPENING SPREAD', 'OPENING TOTAL', 'LINE \nMOVEMENT #1',
       'LINE \nMOVEMENT #2', 'LINE \nMOVEMENT #3', 'CLOSING\nODDS',
       'CLOSING SPREAD', 'CLOSING TOTAL', 'MONEYLINE', 'HALFTIME',
       'BOX SCORE\nURL', 'FULL GAME\nODDS URL', 'Team_Abbrev',
       'Nba_com_team_Abbrev', 'Opp_Abbrev', 'Nba_com_team_2_abbr',
       'Date_underscore', 'HomeTeam', 'AwayTeam', 'Matchup_GameDate'],
      dtype='object')
```

```
In [ ]: df2.rename(columns={'BIGDATABALL\nDATASET': 'DATASET', 'Opp_Abbrev' : 'Opp_abbr',
```

```
In [ ]: # get the differences in the column names
set(df.columns) - set(df2.columns)
```

```
Out[ ]: {'CREW', 'MAIN REF', 'ODDS\nURL'}
```

```
In [ ]: # append this seasons data to the historical data
df = df.append(df2)
df
```

```
C:\Users\Travis\AppData\Local\Temp\ipykernel_47744\1804925609.py:2: FutureWarning:
The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
df = df.append(df2)
```

Out[]:

	DATASET	GAME-ID	DATE	TEAM	VENUE	1Q	2Q	3Q	4Q	OT1	...	Nba_com
0	NBA 2018-2019 Regular Season	21800001	10/16/2018	Philadelphia	R	21	21	24	21	NaN	...	
1	NBA 2018-2019 Regular Season	21800001	10/16/2018	Boston	H	21	26	30	28	NaN	...	
2	NBA 2018-2019 Regular Season	21800002	10/16/2018	Oklahoma City	R	23	24	32	21	NaN	...	
3	NBA 2018-2019 Regular Season	21800002	10/16/2018	Golden State	H	31	26	26	25	NaN	...	
4	NBA 2018-2019 Regular Season	21800003	10/17/2018	Milwaukee	R	36	31	26	20	NaN	...	
...
1145	NBA 2022-2023 Regular Season	22200573	01/04/2023	Golden State	H	26	30	32	31	NaN	...	
1146	NBA 2022-2023 Regular Season	22200574	01/04/2023	Miami	R	23	25	30	31	NaN	...	
1147	NBA 2022-2023 Regular Season	22200574	01/04/2023	LA Lakers	H	32	20	27	33	NaN	...	
1148	NBA 2022-2023 Regular Season	22200575	01/04/2023	Atlanta	R	30	31	30	29	NaN	...	
1149	NBA 2022-2023 Regular Season	22200575	01/04/2023	Sacramento	H	30	28	30	29	NaN	...	

19356 rows × 68 columns

Set up GameID to connect with main testing df

bigdataball uses a different form of gameids than Nba.com or BBallRef.

```
In [ ]: # get opposing team abbrev
def get_opp(df, game_id, team_abbrv):
    data = df[df['GAME-ID'] == game_id]
    opp_abbrv = data[data['Nba_com_team_Abbrev'] != team_abbrv]
    return opp_abbrv['Nba_com_team_Abbrev'].values[0]

In [ ]: df['opponent'] = df.apply(lambda row: get_opp(df, row['GAME-ID'], row['Nba_com_team_Abbrev']), axis=1)
df.head()
```

Out[]:

	DATASET	GAME-ID	DATE	TEAM	VENUE	1Q	2Q	3Q	4Q	OT1	...	Opp_abbrv	
0	NBA 2018-2019 Regular Season	21800001	10/16/2018	Philadelphia		R	21	21	24	21	NaN	...	Bos
1	NBA 2018-2019 Regular Season	21800001	10/16/2018	Boston		H	21	26	30	28	NaN	...	Phi
2	NBA 2018-2019 Regular Season	21800002	10/16/2018	Oklahoma City		R	23	24	32	21	NaN	...	Gol
3	NBA 2018-2019 Regular Season	21800002	10/16/2018	Golden State		H	31	26	26	25	NaN	...	Okc
4	NBA 2018-2019 Regular Season	21800003	10/17/2018	Milwaukee		R	36	31	26	20	NaN	...	Cha

5 rows × 69 columns

```
In [ ]: df['date'] = df['DATE'].astype(str).str.replace('/', '_')
df.head()
```

Out[]:

	DATASET	GAME-ID	DATE	TEAM	VENUE	1Q	2Q	3Q	4Q	OT1	...	Nba_com_team_Abbrev
0	NBA 2018-2019 Regular Season	21800001	10/16/2018	Philadelphia	R	21	21	24	21	NaN	...	
1	NBA 2018-2019 Regular Season	21800001	10/16/2018	Boston	H	21	26	30	28	NaN	...	
2	NBA 2018-2019 Regular Season	21800002	10/16/2018	Oklahoma City	R	23	24	32	21	NaN	...	
3	NBA 2018-2019 Regular Season	21800002	10/16/2018	Golden State	H	31	26	26	25	NaN	...	
4	NBA 2018-2019 Regular Season	21800003	10/17/2018	Milwaukee	R	36	31	26	20	NaN	...	

5 rows × 70 columns

```
In [ ]: def make_gameid(df, gameid, team_abv, home_or_road, date):
    data = df[df['GAME-ID'] == gameid]
    home_team_data = data[data['VENUE'] == 'H']
    home_team = home_team_data['Nba_com_team_Abbrev'].values[0]
    away_team_data = data[data['VENUE'] == 'R']
    away_team = away_team_data['Nba_com_team_Abbrev'].values[0]
    home_gameid = home_team + ' vs. ' + away_team + '_' + date
    away_gameid = away_team + '@' + home_team + '_' + date
    if team_abv == home_team:
        return home_gameid
    elif team_abv == away_team:
        return away_gameid
```

```
In [ ]: df['Matchup_GameDate'] = df.apply(lambda row: make_gameid(df, row['GAME-ID'], row['Nba_com_team_Abbrev'], row['HomeOrAway'], row['Date']), axis=1)
df.head()
```

```
Out[ ]:
```

	DATASET	GAME-ID	DATE	TEAM	VENUE	1Q	2Q	3Q	4Q	OT1	...	Nba_com_team_Abbrev
0	NBA 2018-2019 Regular Season	21800001	10/16/2018	Philadelphia		R	21	21	24	21	NaN	...
1	NBA 2018-2019 Regular Season	21800001	10/16/2018	Boston		H	21	26	30	28	NaN	...
2	NBA 2018-2019 Regular Season	21800002	10/16/2018	Oklahoma City		R	23	24	32	21	NaN	...
3	NBA 2018-2019 Regular Season	21800002	10/16/2018	Golden State		H	31	26	26	25	NaN	...
4	NBA 2018-2019 Regular Season	21800003	10/17/2018	Milwaukee		R	36	31	26	20	NaN	...

5 rows × 70 columns

```
In [ ]: df = df.drop(columns = ['BOX SCORE\nURL', 'ODDS\nURL', 'HALFTIME'])
```

```
In [ ]: df = df.drop(columns = ['1Q', '2Q', '3Q', '4Q',
                               'OT1', 'OT2', 'OT3', 'OT4', 'OT5', 'F', 'MIN', 'FG', 'FGA', '3P', '3PA',
                               'FT', 'FTA', 'OR', 'DR', 'TOT', 'A', 'PF', 'ST', 'TO', 'TO\nTO', 'BL',
                               'PTS', 'POSS', 'PACE', 'OEFF', 'DEFF'])
```

```
In [ ]: df.columns
```

```
Out[ ]: Index(['DATASET', 'GAME-ID', 'DATE', 'TEAM', 'VENUE', 'TEAM\nREST DAYS',
               'STARTING LINEUPS', 'Unnamed: 38', 'Unnamed: 39', 'Unnamed: 40',
               'Unnamed: 41', 'MAIN REF', 'CREW', 'OPENING ODDS', 'OPENING SPREAD',
               'OPENING TOTAL', 'LINE \nMOVEMENT #1', 'LINE \nMOVEMENT #2',
               'LINE \nMOVEMENT #3', 'CLOSING\nODDS', 'CLOSING SPREAD',
               'CLOSING TOTAL', 'MONEYLINE', 'Team_Abbrev', 'Nba_com_team_Abbrev',
               'Opp_abbr', 'Nba_com_team_Abbrev.1', 'Date_underscore', 'HomeTeam',
               'AwayTeam', 'Matchup_GameDate', 'CREW CHIEF', 'REFEREE\n&\nUMPIRE',
               'FULL GAME\nODDS URL', 'opponent', 'date'],
              dtype='object')
```

```
In [ ]: df.to_csv('data\\team\\bigdataball\\2018-2023_BDB_Boxes_with_Odds.csv')
```

```
In [ ]: main_df = pd.read_csv('data/team/All_Things_8.csv')
main_df.head()
```

```
Out[ ]:   Unnamed: 0.3  Unnamed: 0.2  Unnamed: 0.1  Unnamed: 0  trad_team  trad_matchup  trad_gamedate  trad_w,
```

	Unnamed: 0.3	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	trad_team	trad_matchup	trad_gamedate	trad_w,
0	0	0	0	0	CHI	CHI @ NYK	2022-12-23	\
1	1	1	1	15	DET	DET @ ATL	2022-12-23	
2	2	2	2	27	LAC	LAC @ PHI	2022-12-23	
3	3	3	3	26	ATL	ATL vs. DET	2022-12-23	\
4	4	4	4	25	MEM	MEM @ PHX	2022-12-23	\

5 rows × 807 columns

```
In [ ]: main_df.Matchup_GameDate
```

```
Out[ ]: 0           CHI @ NYK_12_23_2022
1           DET @ ATL_12_23_2022
2           LAC @ PHI_12_23_2022
3           ATL vs. DET_12_23_2022
4           MEM @ PHX_12_23_2022
...
15097       NYK @ CLE_10_25_2016
15098       GSW vs. SAS_10_25_2016
15099       UTA @ POR_10_25_2016
15100       CLE vs. NYK_10_25_2016
15101       POR vs. UTA_10_25_2016
Name: Matchup_GameDate, Length: 15102, dtype: object
```

```
In [ ]: df.dtypes
```

```
Out[ ]: DATASET          object
        GAME-ID         int64
        DATE            object
        TEAM            object
        VENUE           object
        TEAM\nREST DAYS object
        STARTING LINEUPS object
        Unnamed: 38      object
        Unnamed: 39      object
        Unnamed: 40      object
        Unnamed: 41      object
        MAIN REF         object
        CREW             object
        OPENING ODDS     object
        OPENING SPREAD    float64
        OPENING TOTAL    float64
        LINE \nMOVEMENT #1 object
        LINE \nMOVEMENT #2 object
        LINE \nMOVEMENT #3 object
        CLOSING\nODDS    object
        CLOSING SPREAD   float64
        CLOSING TOTAL    float64
        MONEYLINE         object
        Team_Abbrev       object
        Nba_com_team_Abbrev object
        Opp_abbrev         object
        Nba_com_team_Abbrev.1 object
        Date_underscore   object
        HomeTeam          object
        AwayTeam          object
        Matchup_GameDate  object
        CREW CHIEF        object
        REFEREE\n&\nUMPIRE   object
        FULL GAME\nODDS URL object
        opponent          object
        date              object
        dtype: object
```

```
In [ ]: df.DATASET = df.DATASET.astype(str)
        df.DATE = df.DATE.astype(str)
        df['GAME-ID'] = df['GAME-ID'].astype(str)
        df.Nba_com_team_Abbrev = df.Nba_com_team_Abbrev.astype(str)
```

```
In [ ]: # merge the two

new_df = pd.merge(main_df, df, on='Matchup_GameDate', how='left')
new_df.head()
```

Out[]:

	Unnamed: 0.3	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	trad_team	trad_matchup	trad_gamedate	trad_w,
0	0	0	0	0	CHI	CHI @ NYK	2022-12-23	\
1	0	0	0	0	CHI	CHI @ NYK	2022-12-23	\
2	1	1	1	15	DET	DET @ ATL	2022-12-23	
3	1	1	1	15	DET	DET @ ATL	2022-12-23	
4	2	2	2	27	LAC	LAC @ PHI	2022-12-23	

5 rows × 842 columns

In []: `new_df.to_csv('data/team/All_Things_9.csv')`