

```

import numpy as np
import pandas as pd
import json
import ast
import re

In [2]:
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]:
player_dict = players.get_players()

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

In [6]:
steph

Out[6]:
[{'full_name': 'Stephen Curry',
  'first_name': 'Stephen',
  'last_name': 'Curry',
  'team_id': 1610612744}]

In [7]:
steph_id = steph[0]['id']

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

In [10]:
gsw = team [team in team_dict if team['abbreviation'] == "GSW"]

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

In [13]:
warriors_g1 = teamgamelog.TeamGameLog(gsw_id)

In [14]:
warriors_20_df = warriors_g1.get_data_frames()[0]

In [15]:
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  FG3A
10  1610612744  0022000378  FEB 09, 2021  GSW @ SAS  W  12  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]:
gamelog_steph = playergamelog.PlayerGameLog(steph_id)

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

In [18]:
gamelog_steph_df.head()

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]:
gamelog_steph_df['PAR'] = gamelog_steph_df['PTS'] +
gamelog_steph_df['AST'] + gamelog_steph_df['REB']

In [20]:
gamelog_steph_df.head()

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]:
gamelog_steph_df['BPM'].mean()

Out[21]:
1.2311111111111111

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

In [23]:
brogdon_id = brogdon[0]['id']

In [24]:
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]:
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]:
brog_17to20_df = pd.concat([gamelog_brogdon_17_df, gamelog_brogdon_18_df,
                             gamelog_brogdon_19_df, gamelog_brogdon_20_df])

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

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

In [29]:
brog_17to20_df.head()

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]:
brog_17to20_df['PAR'] = brog_17to20_df['PTS'] + brog_17to20_df['AST'] +
brog_17to20_df['REB']

In [31]:
brog_17to20_df.head()

In [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]:
brog_vs_bkn =
brog_17to20_df[brog_17to20_df['MATCHUP'].str.contains('BRN')]

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

In [36]:
def oppFinder(row):
    return row.split('@')[1]

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

Out[37]:
IND

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

In [40]:
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]:
   AVG_PLUSMINUS  AVG_REB  AVG_STL  AVG_PTS  AVG_AST  AVG_PAR
OPPONENT
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      4.33     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]:
with open('json_nba.json') as f:
    data = ast.literal_eval(f)

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

In [45]:
json = json_data[0]

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

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

In [51]:
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]:
competitors = pd.json_normalize(df['competitors'][0])
#competitors of the current game matchup

In [54]:
competitors

Out[54]:
   id  name  home
0  8301268-204  San Antonio Spurs  True
1  8301268-1175731  Golden State Warriors  False

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

In [56]:
displayGroups

Out[56]:
   id  description  defaultType  alternateType  markets  order
0  100-97  Game Lines  True  False  [{"id": "144044853", "descriptionKey": "Head T...  1
1  100-101  Period/Alternate Lines  False  False  [{"id": "G-2W-HCAP Handicap - Asian209", "des...  4
2  100-104  Game Props  False  False  [{"id": "14404484
```