Players

March 25, 2018

```
In [1]: import numpy
        import pandas
```

```
In [2]: from nba_py.player import PlayerGeneralSplits, PlayerGameLogs, PlayerProfile, PlayerLis
        from nba_py.team import TeamList, TeamCommonRoster,TeamShootingSplits
```

```
In [3]: PlayerList(season='2016-17').info().info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 132 entries, 0 to 131
Data columns (total 13 columns):
PERSON_ID                   132 non-null int64
DISPLAY_LAST_COMMA_FIRST    132 non-null object
DISPLAY_FIRST_LAST          132 non-null object
ROSTERSTATUS                132 non-null int64
FROM_YEAR                   132 non-null object
TO_YEAR                     132 non-null object
PLAYERCODE                  132 non-null object
TEAM_ID                     132 non-null int64
TEAM_CITY                   132 non-null object
TEAM_NAME                   132 non-null object
TEAM_ABBREVIATION           132 non-null object
TEAM_CODE                   132 non-null object
GAMES_PLAYED_FLAG           132 non-null object
dtypes: int64(3), object(10)
memory usage: 8.3+ KB
```

```
In [4]: last_season = "2016-17"
        warriors = "1610612744"
        cavs = "1610612739"
        thunder = "1610612760"
        raps = "1610612761"
        player = 2544
        first_game = "1610612744"
```

```
In [5]: import time
```

```
In [6]: teams = TeamList().info().head(30)
```

```
In [10]: team_shooting = TeamShootingSplits('1610612766').overall()

In [57]:

Out[57]:   GROUP_SET GROUP_VALUE   FGM   FGA  FG_PCT  FG3M  FG3A  FG3_PCT  EFG_PCT  \
         0   Overall     2016-17  3093  7000   0.442   824  2347    0.351    0.501

            BLKA    ...     EFG_PCT_RANK  BLKA_RANK  PCT_AST_2PM_RANK  \
         0   450    ...                1          1                 1

            PCT_UAST_2PM_RANK  PCT_AST_3PM_RANK  PCT_UAST_3PM_RANK  PCT_AST_FGM_RANK  \
         0                  1                 1                  1                 1

            PCT_UAST_FGM_RANK  CFID  CFPARAMS
         0                  1   170   2016-17

         [1 rows x 32 columns]

In [11]: team_shooting.head()

Out[11]:   GROUP_SET GROUP_VALUE   FGM   FGA  FG_PCT  FG3M  FG3A  FG3_PCT  EFG_PCT  \
         0   Overall     2016-17  3093  7000   0.442   824  2347    0.351    0.501

            BLKA    ...     EFG_PCT_RANK  BLKA_RANK  PCT_AST_2PM_RANK  \
         0   450    ...                1          1                 1

            PCT_UAST_2PM_RANK  PCT_AST_3PM_RANK  PCT_UAST_3PM_RANK  PCT_AST_FGM_RANK  \
         0                  1                 1                  1                 1

            PCT_UAST_FGM_RANK  CFID  CFPARAMS
         0                  1   170   2016-17

         [1 rows x 32 columns]

In [12]: team_ids = teams['TEAM_ID']

In [13]: team_shooting_ovr = []
         for i in range (0,30):
             team_shooting_ovr.append(TeamShootingSplits(team_ids[i]).overall())

In [14]: team_shooting_df = pandas.concat(team_shooting_ovr)

In [15]: team_shooting_df.head()

Out[15]:   GROUP_SET GROUP_VALUE   FGM   FGA  FG_PCT  FG3M  FG3A  FG3_PCT  EFG_PCT  \
         0   Overall     2016-17  3123  6918   0.451   729  2137    0.341    0.504
         0   Overall     2016-17  3168  6978   0.454   985  2742    0.359    0.525
         0   Overall     2016-17  3275  6963   0.470  1067  2779    0.384    0.547
         0   Overall     2016-17  3210  7130   0.450   768  2196    0.350    0.504
```

```
0   Overall     2016-17  3169  7141   0.444   623  1831    0.340    0.487

   BLKA    ...     EFG_PCT_RANK  BLKA_RANK  PCT_AST_2PM_RANK  \
0   424    ...                1          1                 1
0   425    ...                1          1                 1
0   349    ...                1          1                 1
0   348    ...                1          1                 1
0   378    ...                1          1                 1

   PCT_UAST_2PM_RANK  PCT_AST_3PM_RANK  PCT_UAST_3PM_RANK  PCT_AST_FGM_RANK  \
0                  1                 1                  1                 1
0                  1                 1                  1                 1
0                  1                 1                  1                 1
0                  1                 1                  1                 1
0                  1                 1                  1                 1

   PCT_UAST_FGM_RANK  CFID  CFPARAMS
0                  1   170   2016-17
0                  1   170   2016-17
0                  1   170   2016-17
0                  1   170   2016-17
0                  1   170   2016-17

[5 rows x 32 columns]
```

```
In [16]: team_shooting_df['FG3_PCT'].mean()
```

```
Out[16]: 0.3571666666666667
```

```
In [7]: rosters = []
        for team in teams['TEAM_ID']:
            rosters.append(TeamCommonRoster(team, season=last_season).roster().drop(['SCHOOL',
```

```
In [18]: team_abrv = []
         abbrvs = teams['ABBREVIATION']
```

```
In [20]: for i in range(0,30):
             team_abrv.append(abbrvs.iloc[i])
```

```
In [21]: team_abrv[28]
```

```
Out[21]: 'DET'
```

```
In [12]: def compute_stats(player_id, season):
             stats = PlayerGameLogs(player_id, season=season).info().drop(['SEASON_ID','Player_
             wins_lose = stats['WL'].value_counts()
             #print(wins_lose.get('W'))
             games = stats.shape[0]
```

```
            stat = stats.drop(['WL'], axis = 1)
            avgs = stats.mean()
            if wins_lose.get('W'):
                avgs['WINS'] = wins_lose.get('W')
            else:
                avgs['WINS'] = 0
            if wins_lose.get('L'):
                avgs['LOSES'] = wins_lose.get('L')
            else:
                avgs['LOSES'] = 0
            avgs['GP'] = games
            return avgs
```

```
In [13]: def save_team(roster):
             stats = roster['PLAYER_ID'].apply(lambda player: compute_stats(player, last_seasor
             pandas.concat([roster, stats], axis=1).to_csv('team.csv')
```

```
In [18]: rosters[29].head()
```

Out[18]:

	TeamID	SEASON	PLAYER	NUM	POSITION	HEIGHT	WEIGHT	AGE	EXP	\
0	1610612766	2016	Briante Weber	0	G	6-2	165	24.0	1	
1	1610612766	2016	Marvin Williams	2	F	6-9	237	31.0	11	
2	1610612766	2016	Jeremy Lamb	3	G	6-5	185	25.0	4	
3	1610612766	2016	Nicolas Batum	5	G	6-8	200	28.0	8	
4	1610612766	2016	Ramon Sessions	7	G	6-3	190	31.0	9	

	PLAYER_ID
0	1627362
1	101107
2	203087
3	201587
4	201196

```
In [19]: save_team(rosters[3])
```

```
In [14]: for i in range(2,5):
             time.sleep(10)
             stats = rosters[i]['PLAYER_ID'].apply(lambda player: compute_stats(player, last_se
             pandas.concat([rosters[i], stats], axis=1).to_csv(team_abrv[i]+'.csv')
```

```
In [15]: for i in range(5,8):
             time.sleep(10)
             stats = rosters[i]['PLAYER_ID'].apply(lambda player: compute_stats(player, last_se
             pandas.concat([rosters[i], stats], axis=1).to_csv(team_abrv[i]+'.csv')
```

```
In [16]: for i in range(8,11):
             time.sleep(10)
             stats = rosters[i]['PLAYER_ID'].apply(lambda player: compute_stats(player, last_se
             pandas.concat([rosters[i], stats], axis=1).to_csv(team_abrv[i]+'.csv')
```

```
In [17]: for i in range(11,14):
             time.sleep(10)
             stats = rosters[i]['PLAYER_ID'].apply(lambda player: compute_stats(player, last_se
             pandas.concat([rosters[i], stats], axis=1).to_csv(team_abrv[i]+'.csv')

In [18]: for i in range(14,17):
             time.sleep(5)
             stats = rosters[i]['PLAYER_ID'].apply(lambda player: compute_stats(player, last_se
             pandas.concat([rosters[i], stats], axis=1).to_csv(team_abrv[i]+'.csv')

In [19]: for i in range(17,22):
             time.sleep(5)
             stats = rosters[i]['PLAYER_ID'].apply(lambda player: compute_stats(player, last_se
             pandas.concat([rosters[i], stats], axis=1).to_csv(team_abrv[i]+'.csv')

In [20]: for i in range(22,27):
             time.sleep(5)
             stats = rosters[i]['PLAYER_ID'].apply(lambda player: compute_stats(player, last_se
             pandas.concat([rosters[i], stats], axis=1).to_csv(team_abrv[i]+'.csv')

In [21]: for i in range(27,30):
             time.sleep(5)
             stats = rosters[i]['PLAYER_ID'].apply(lambda player: compute_stats(player, last_se
             pandas.concat([rosters[i], stats], axis=1).to_csv(team_abrv[i]+'.csv')

In [22]: all_teams = []
         for i in range(0,30):
             all_teams.append(pandas.read_csv(team_abrv[i]+'.csv'))

In [23]: totalPlayers = pandas.concat(all_teams)

In [32]: all_teams[0]['MIN']

Out[32]: 0     33.956522
         1     17.082192
         2     26.134146
         3     29.662162
         4     27.215190
         5     16.644068
         6     13.146341
         7     31.493671
         8     26.890411
         9     25.725806
         10     6.937500
         11    17.671429
         12    15.943396
         13    12.303571
         14     9.736842
         Name: MIN, dtype: float64
```

```
In [26]: totalPlayers.to_csv('all_players.csv')

In [29]: totalPlayers.head()
```

Out[29]:

	AGE	AST	BLK	DREB	EXP	FG3A	FG3M	FG3_PCT	\
0	32.0	3.652174	0.898551	6.115942	10	3.492754	1.086957	0.280072	
1	28.0	2.643836	0.013699	1.547945	R	1.506849	0.356164	0.151822	
2	30.0	1.719512	0.280488	4.463415	8	4.902439	1.731707	0.357598	
3	31.0	1.405405	1.243243	8.702703	12	0.027027	0.000000	0.000000	
4	25.0	2.303797	0.189873	2.392405	3	5.278481	1.886076	0.332190	

	FGA	FGM	...	PTS	REB	SEASON	STL	TOV	\
0	14.086957	6.231884	...	18.057971	7.724638	2016	1.304348	2.289855	
1	5.315068	1.986301	...	5.356164	1.684932	2016	0.534247	1.301370	
2	10.853659	4.682927	...	13.060976	5.865854	2016	0.707317	1.414634	
3	8.283784	5.243243	...	13.540541	12.702703	2016	0.864865	2.297297	
4	11.544304	5.253165	...	14.468354	2.835443	2016	0.696203	1.341772	

	TeamID	Unnamed: 0	WEIGHT	WINS	WL
0	1610612737	0	246	40.0	NaN
1	1610612737	1	190	37.0	NaN
2	1610612737	2	235	35.0	NaN
3	1610612737	3	265	37.0	NaN
4	1610612737	4	205	42.0	NaN

[5 rows x 35 columns]

```
In [27]: significantPlayers = totalPlayers[totalPlayers['MIN'] > 10.0]

In [28]: significantPlayers = significantPlayers[significantPlayers['GP'] > 40]

In [45]: significantPlayers['FG3_PCT_ADJ'] = significantPlayers['FG3_PCT'].apply(lambda x: x i

In [34]: significantPlayers.to_csv('sig_players.csv')

In [31]: tpm = significantPlayers['FG3M'].sum()

In [32]: tpmissed = significantPlayers['FG3A'].sum()

In [33]: tpm/tpmissed
```

Out[33]: 0.3598123338854981

```
In [ ]: cavs_lineup.iloc[1]['GROUP_NAME']

In [ ]: war_lineup = TeamLineups(raps, season=last_season).lineups().sort_values('GP', ascendi
        war_lineup

In [ ]: TeamGameLogs(warriors, season=last_season).info()
```

```
In [ ]: from nba_py.game import BoxscoreMisc
        BoxscoreMisc("0021700842").sql_players_misc()

In [ ]: PlayerGameLogs(player, season=last_season).info()['AST'].mean()

In [ ]: PlayerGeneralSplits("101112", season=last_season).starting_position()

In [ ]:

In [ ]:
```