In [134]:
```
import pandas as pd
```

In [135]:
```
s2017_df = pd.read_csv('2017_season_data.csv')
```

In [136]:
```
players_df = pd.read_csv('player_data.csv')
```

In [137]:
```
s2017_df.head(3)
```

Out[137]:

	Year	Player	Pos	Age	Tm	G	GS	MP	PER	TS%	...	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
0	2017.0	Alex Abrines	SG	23.0	OKC	68.0	6.0	1055.0	10.1	0.560	...	0.898	18.0	68.0	86.0	40.0	37.0	8.0	33.0	114.0	406.0
1	2017.0	Quincy Acy	PF	26.0	TOT	38.0	1.0	558.0	11.8	0.565	...	0.750	20.0	95.0	115.0	18.0	14.0	15.0	21.0	67.0	222.0
2	2017.0	Quincy Acy	PF	26.0	DAL	6.0	0.0	48.0	-1.4	0.355	...	0.667	2.0	6.0	8.0	0.0	0.0	0.0	2.0	9.0	13.0

3 rows × 52 columns

In [138]:
```
players_df.head(3)
```

Out[138]:

	name	year_start	year_end	position	height	weight	birth_date	college
0	Alaa Abdelnaby	1991	1995	F-C	6-10	240.0	June 24, 1968	Duke University
1	Zaid Abdul-Aziz	1969	1978	C-F	6-9	235.0	April 7, 1946	Iowa State University
2	Kareem Abdul-Jabbar	1970	1989	C	7-2	225.0	April 16, 1947	University of California, Los Angeles

Data Wrangling Activities

1. Merge `s2017_df` and `players_df` with a left join

In [139]:
```
df = s2017_df.merge(players_df,how='left',left_on='Player',right_on='name')
df.head()
```

Out[139]:

	Year	Player	Pos	Age	Tm	G	GS	MP	PER	TS%	...	PF	PTS	name	year_start	year_end	position	height	weight	birth
0	2017.0	Alex Abrines	SG	23.0	OKC	68.0	6.0	1055.0	10.1	0.560	...	114.0	406.0	Alex Abrines	2017.0	2018.0	G-F	6-6	190.0	Aug
1	2017.0	Quincy Acy	PF	26.0	TOT	38.0	1.0	558.0	11.8	0.565	...	67.0	222.0	Quincy Acy	2013.0	2018.0	F	6-7	240.0	Octo
2	2017.0	Quincy Acy	PF	26.0	DAL	6.0	0.0	48.0	-1.4	0.355	...	9.0	13.0	Quincy Acy	2013.0	2018.0	F	6-7	240.0	Octo
3	2017.0	Quincy Acy	PF	26.0	BRK	32.0	1.0	510.0	13.1	0.587	...	58.0	209.0	Quincy Acy	2013.0	2018.0	F	6-7	240.0	Octo
4	2017.0	Steven Adams	C	23.0	OKC	80.0	80.0	2389.0	16.5	0.589	...	195.0	905.0	Steven Adams	2014.0	2018.0	C	7-0	255.0	Ju

5 rows × 60 columns

In [140]:
```
# Use it before modifying the `df` to have a copy
# just in case a modification doesn't go as expected
df_copy = df.copy()
```

2. Are there misses (mismatches) in the resulting dataframe?

In [141]:
```
df['name'].isna().sum()
```

Out[141]:
```
4
```

3. How many rows couldn't be matched?

In [142]:
```
df['name'].isna().sum()
```

Out[142]:
```
4
```

4. Extract the names of the players that couldn't be matched

In [143]:
```
df.loc[df['name'].isna()]['Player']
```

Out[143]:
```
349             Luc Mbah
350        James Michael
352    Sheldon McClellan
593          Metta World
Name: Player, dtype: object
```

In [144]:
```
player_misses = list(df.loc[df['name'].isna()]['Player'])
player_misses
```

Out[144]:
```
['Luc Mbah', 'James Michael', 'Sheldon McClellan', 'Metta World']
```

5. Modify `players_df` with the correct names to re-try a successful merge

In [145]:
```
# Use it before modifying the `df` to have a copy
# just in case a modification doesn't go as expected
df_copy = df.copy()
```

In [146]:
```
players_df.loc[players_df['name'].str.lower().str.contains('mbah')]
```

Out[146]:

	name	year_start	year_end	position	height	weight	birth_date	college
2595	Luc Mbah a Moute	2009	2018	F	6-8	230.0	September 9, 1986	University of California, Los Angeles

In [147]:
```
players_df.loc[players_df['name'].str.lower().str.contains('sheldon')]
```

Out[147]:

	name	year_start	year_end	position	height	weight	birth_date	college
2475	Sheldon Mac	2017	2017	G	6-5	200.0	December 21, 1992	University of Miami

In [148]:

```
players_df.loc[players_df['name'].str.lower().str.contains('metta')]
```

Out[148]:

	name	year_start	year_end	position	height	weight	birth_date	college
4487	Metta World Peace	2000	2017	F	6-6	260.0	November 13, 1979	St. John's University

In [149]:

```
players_df.loc[players_df['name'].str.contains('James Michael')]
```

Out[149]:

	name	year_start	year_end	position	height	weight	birth_date	college
2597	James Michael McAdoo	2015	2018	F	6-9	230.0	January 4, 1993	University of North Carolina

In [150]:

```
# name_mapping = {'Luc Mbah':'Luc Mbah a Moute',
#                 'Sheldon McClellan':'Sheldon Mac',
#                 'Metta World':'Metta World Peace',
#                 'James Michael':'James Michael McAdoo'}

name_mapping = {'Luc Mbah a Moute':'Luc Mbah',
               'Sheldon Mac':'Sheldon McClellan',
               'Metta World Peace':'Metta World',
               'James Michael McAdoo':'James Michael'}
```

In [151]:

```
for name in name_mapping.keys():
    players_df.loc[players_df['name']==name,'name']=name_mapping[name]
```

6. Perform the merge between `s2017_df` and `players_df` again, this time, without misses

In [152]:

```
df = s2017_df.merge(players_df,how='left',left_on='Player',right_on='name')
```

7. Remove unnecessary columns

In [153]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 605 entries, 0 to 604
Data columns (total 60 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Year        605 non-null    float64
 1   Player      605 non-null    object 
 2   Pos         605 non-null    object 
 3   Age         605 non-null    float64
 4   Tm          605 non-null    object 
 5   G           605 non-null    float64
 6   GS          605 non-null    float64
 7   MP          605 non-null    float64
 8   PER         605 non-null    float64
 9   TS%         603 non-null    float64
 10  3PAr        603 non-null    float64
 11  FTr         603 non-null    float64
 12  ORB%        605 non-null    float64
 13  DRB%        605 non-null    float64
 14  TRB%        605 non-null    float64
 15  AST%        605 non-null    float64
 16  STL%        605 non-null    float64
 17  BLK%        605 non-null    float64
 18  TOV%        603 non-null    float64
 19  USG%        605 non-null    float64
 20  blanl       0 non-null      float64
 21  OWS         605 non-null    float64
 22  DWS         605 non-null    float64
 23  WS          605 non-null    float64
 24  WS/48       605 non-null    float64
 25  blank2      0 non-null      float64
 26  OBPM        605 non-null    float64
 27  DBPM        605 non-null    float64
 28  BPM         605 non-null    float64
 29  VORP        605 non-null    float64
 30  FG          605 non-null    float64
 31  FGA         605 non-null    float64
 32  FG%         603 non-null    float64
 33  3P          605 non-null    float64
 34  3PA         605 non-null    float64
 35  3P%         558 non-null    float64
 36  2P          605 non-null    float64
 37  2PA         605 non-null    float64
 38  2P%         600 non-null    float64
 39  eFG%        603 non-null    float64
 40  FT          605 non-null    float64
 41  FTA         605 non-null    float64
 42  FT%         581 non-null    float64
 43  ORB         605 non-null    float64
 44  DRB         605 non-null    float64
 45  TRB         605 non-null    float64
 46  AST         605 non-null    float64
 47  STL         605 non-null    float64
 48  BLK         605 non-null    float64
 49  TOV         605 non-null    float64
 50  PF          605 non-null    float64
 51  PTS         605 non-null    float64
 52  name        605 non-null    object
 53  year_start  605 non-null    int64
 54  year_end    605 non-null    int64
 55  position    605 non-null    object
 56  height      605 non-null    object
 57  weight      605 non-null    float64
 58  birth_date  605 non-null    object
 59  college     491 non-null    object
dtypes: float64(50), int64(2), object(8)
memory usage: 283.7+ KB
```

In [154]:

```
columns_to_drop = [
    "Year",
    "PER",
    "TS%",
    "3PAr",
    "FTr",
    "USG%",
    "blanl",
    "OWS",
    "DWS",
    "WS",
    "WS/48",
    "blank2",
    "OBPM",
    "DBPM",
    "BPM",
    "VORP",
    "FG%",
    "3P%",
    "eFG%",
    "FT%",
    "name",
]
```

In [155]:

```
df.drop(columns = columns_to_drop,inplace=True)
```

In [156]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 605 entries, 0 to 604
Data columns (total 39 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Player      605 non-null    object 
 1   Pos         605 non-null    object 
 2   Age         605 non-null    float64
 3   Tm          605 non-null    object 
 4   G           605 non-null    float64
 5   GS          605 non-null    float64
 6   MP          605 non-null    float64
 7   ORB%        605 non-null    float64
 8   DRB%        605 non-null    float64
 9   TRB%        605 non-null    float64
 10  AST%        605 non-null    float64
 11  STL%        605 non-null    float64
 12  BLK%        605 non-null    float64
 13  TOV%        603 non-null    float64
 14  FG          605 non-null    float64
 15  FGA         605 non-null    float64
 16  3P          605 non-null    float64
 17  3PA         605 non-null    float64
 18  2P          605 non-null    float64
 19  2PA         605 non-null    float64
 20  2P%         600 non-null    float64
 21  FT          605 non-null    float64
 22  FTA         605 non-null    float64
 23  ORB         605 non-null    float64
 24  DRB         605 non-null    float64
 25  TRB         605 non-null    float64
 26  AST         605 non-null    float64
 27  STL         605 non-null    float64
 28  BLK         605 non-null    float64
 29  TOV         605 non-null    float64
 30  PF          605 non-null    float64
 31  PTS         605 non-null    float64
 32  year_start  605 non-null    int64  
 33  year_end    605 non-null    int64
 34  position    605 non-null    object
 35  height      605 non-null    object
 36  weight      605 non-null    float64
 37  birth_date  605 non-null    object
 38  college     491 non-null    object
dtypes: float64(30), int64(2), object(7)
memory usage: 184.5+ KB
```

8. Rename teams to their full name

In [157]:

```
team_mapping = {
    "OKC": "Oklahoma City Thunder",
    "DAL": "Dallas Mavericks",
    "BRK": "Brooklyn Nets",
    "SAC": "Sacramento Kings",
    "NOP": "New Orleans Pelicans",
    "MIN": "Minnesota Timberwolves",
    "SAS": "San Antonio Spurs",
    "IND": "Indiana Pacers",
    "MEM": "Memphis Grizzlies",
    "POR": "Portland Trail Blazers",
    "CLE": "Cleveland Cavaliers",
    "LAC": "Los Angeles Clippers",
    "PHI": "Philadelphia 76ers",
    "HOU": "Houston Rockets",
    "MIL": "Milwaukee Bucks",
    "NYK": "New York Knicks",
    "DEN": "Denver Nuggets",
    "ORL": "Orlando Magic",
    "MIA": "Miami Heat",
    "PHO": "Phoenix Suns",
    "GSW": "Golden State Warriors",
    "CHO": "Charlotte Hornets",
    "DET": "Detroit Pistons",
    "ATL": "Atlanta Hawks",
    "WAS": "Washington Wizards",
    "LAL": "Los Angeles Lakers",
    "UTA": "Utah Jazz",
    "BOS": "Boston Celtics",
    "CHI": "Chicago Bulls",
    "TOR": "Toronto Raptors"
}
```

In [158]:
```
df['Team']=df['Tm'].replace(team_mapping)
df.head()
```

Out[158]:

	Player	Pos	Age	Tm	G	GS	MP	ORB%	DRB%	TRB%	...	PF	PTS	year_start	year_end	position	height	weight	birth_date
0	Alex Abrines	SG	23.0	OKC	68.0	6.0	1055.0	1.9	7.1	4.5	...	114.0	406.0	2017	2018	G-F	6-6	190.0	August 1, 1993
1	Quincy Acy	PF	26.0	TOT	38.0	1.0	558.0	3.9	18.0	11.0	...	67.0	222.0	2013	2018	F	6-7	240.0	October 6, 1990
2	Quincy Acy	PF	26.0	DAL	6.0	0.0	48.0	4.6	15.2	9.7	...	9.0	13.0	2013	2018	F	6-7	240.0	October 6, 1990
3	Quincy Acy	PF	26.0	BRK	32.0	1.0	510.0	3.8	18.2	11.1	...	58.0	209.0	2013	2018	F	6-7	240.0	October 6, 1990
4	Steven Adams	C	23.0	OKC	80.0	80.0	2389.0	13.0	15.5	14.2	...	195.0	905.0	2014	2018	C	7-0	255.0	July 20, 1993

5 rows × 40 columns

9. Convert birthday to a datetime object

In [159]:
```
df['birth_date'] = pd.to_datetime(df['birth_date'])
df.head()
```

Out[159]:

	Player	Pos	Age	Tm	G	GS	MP	ORB%	DRB%	TRB%	...	PF	PTS	year_start	year_end	position	height	weight	birth_date
0	Alex Abrines	SG	23.0	OKC	68.0	6.0	1055.0	1.9	7.1	4.5	...	114.0	406.0	2017	2018	G-F	6-6	190.0	1993-08-01
1	Quincy Acy	PF	26.0	TOT	38.0	1.0	558.0	3.9	18.0	11.0	...	67.0	222.0	2013	2018	F	6-7	240.0	1990-10-06
2	Quincy Acy	PF	26.0	DAL	6.0	0.0	48.0	4.6	15.2	9.7	...	9.0	13.0	2013	2018	F	6-7	240.0	1990-10-06
3	Quincy Acy	PF	26.0	BRK	32.0	1.0	510.0	3.8	18.2	11.1	...	58.0	209.0	2013	2018	F	6-7	240.0	1990-10-06
4	Steven Adams	C	23.0	OKC	80.0	80.0	2389.0	13.0	15.5	14.2	...	195.0	905.0	2014	2018	C	7-0	255.0	1993-07-20

5 rows × 40 columns

10. Delete all players from the `TOT` team

In [160]:
```
# if something goes wrong, just execute this line:
df_copy = df.copy()
```

In [162]:
```
df = df.loc[df['Tm']!='TOT']
```

In [163]:
```
df.shape
```

Out[163]:
```
(551, 40)
```

Analysis

11. What's the team with the most players in the league?

```
In [181]: df = df_copy.copy()
```

In [182]:
```
df['Team'].value_counts().sort_values(ascending=False)
```

Out[182]:
```
Team
TOT                       54
New Orleans Pelicans      27
Dallas Mavericks          24
Philadelphia 76ers        22
Atlanta Hawks             22
Cleveland Cavaliers       22
Brooklyn Nets             21
Milwaukee Bucks           20
Los Angeles Lakers        19
Charlotte Hornets         19
Orlando Magic             19
Denver Nuggets            19
Oklahoma City Thunder     19
Sacramento Kings          19
Washington Wizards        18
Houston Rockets           18
Chicago Bulls             18
Phoenix Suns              18
San Antonio Spurs         17
Golden State Warriors     17
Toronto Raptors           17
Indiana Pacers            17
Memphis Grizzlies         17
Minnesota Timberwolves    16
New York Knicks           16
Miami Heat                15
Detroit Pistons           15
Utah Jazz                 15
Boston Celtics            15
Portland Trail Blazers    15
Los Angeles Clippers      15
Name: count, dtype: int64
```

12. What's the team with the lowest `FG` ?

```
In [183]: df.groupby('Team')['FG'].sum().sort_values(ascending=True)
```

```
Out[183]: Team
          Dallas Mavericks          2968.0
          Memphis Grizzlies         2984.0
          Utah Jazz                 3033.0
          Charlotte Hornets         3093.0
          Brooklyn Nets             3102.0
          Sacramento Kings          3105.0
          Orlando Magic             3139.0
          Boston Celtics            3168.0
          Chicago Bulls             3169.0
          Milwaukee Bucks           3190.0
          Miami Heat                3202.0
          Toronto Raptors           3211.0
          New Orleans Pelicans      3218.0
          Minnesota Timberwolves    3235.0
          Oklahoma City Thunder     3237.0
          Los Angeles Clippers      3242.0
          Portland Trail Blazers    3243.0
          New York Knicks           3244.0
          Detroit Pistons           3269.0
          Phoenix Suns              3270.0
          Houston Rockets           3305.0
          Cleveland Cavaliers       3311.0
          Philadelphia 76ers        3322.0
          Denver Nuggets            3377.0
          Indiana Pacers            3379.0
          Washington Wizards        3388.0
          Los Angeles Lakers        3414.0
          San Antonio Spurs         3470.0
          Golden State Warriors     3532.0
          Atlanta Hawks             3595.0
          TOT                       8434.0
          Name: FG, dtype: float64
```

13. What's the team with the best FG% ?

In [187]:

```
FG_df = df.groupby('Team')[['FG','FGA']].sum()
FG_df
```

Out[187]:

	FG	FGA
Team		
Atlanta Hawks	3595.0	7961.0
Boston Celtics	3168.0	6978.0
Brooklyn Nets	3102.0	6987.0
Charlotte Hornets	3093.0	7000.0
Chicago Bulls	3169.0	7142.0
Cleveland Cavaliers	3311.0	7053.0
Dallas Mavericks	2968.0	6750.0
Denver Nuggets	3377.0	7194.0
Detroit Pistons	3269.0	7282.0
Golden State Warriors	3532.0	7140.0
Houston Rockets	3305.0	7152.0
Indiana Pacers	3379.0	7270.0
Los Angeles Clippers	3242.0	6819.0
Los Angeles Lakers	3414.0	7525.0
Memphis Grizzlies	2984.0	6854.0
Miami Heat	3202.0	7037.0
Milwaukee Bucks	3190.0	6737.0
Minnesota Timberwolves	3235.0	6922.0
New Orleans Pelicans	3218.0	7154.0
New York Knicks	3244.0	7255.0
Oklahoma City Thunder	3237.0	7169.0

	FG	FGA
Team		
Orlando Magic	3139.0	7133.0
Philadelphia 76ers	3322.0	7545.0
Phoenix Suns	3270.0	7260.0
Portland Trail Blazers	3243.0	7059.0
Sacramento Kings	3105.0	6735.0
San Antonio Spurs	3470.0	7284.0
TOT	8434.0	18976.0
Toronto Raptors	3211.0	6918.0

```
In [188]: FG_df['FG%'] = FG_df['FG']/FG_df['FGA']
```

```
In [189]: FG_df.loc[FG_df['FG%']==FG_df['FG%'].max()]
```

Out[189]:

	FG	FGA	FG%
Team			
Golden State Warriors	3532.0	7140.0	0.494678

14. What's the difference between the best and worst 3P shooters (by position)?

In [194]:

```
Pos_df = df.groupby('Pos')[['3P','3PA']].sum()
Pos_df.head()
```

Out[194]:

	3P	3PA
Pos		
C	1617.0	4583.0
PF	3903.0	11325.0
PF-C	33.0	98.0
PG	6116.0	17034.0
SF	6495.0	18508.0

In [199]:

```
Pos_df['3P%'] = Pos_df['3P']/Pos_df['3PA']
Pos_df
```

Out[199]:

	3P	3PA	3P%
Pos			
C	1617.0	4583.0	0.352826
PF	3903.0	11325.0	0.344636
PF-C	33.0	98.0	0.336735
PG	6116.0	17034.0	0.359047
SF	6495.0	18508.0	0.350929
SG	8357.0	22627.0	0.369338

In [201]:

```
Pos_df['3P%'].max()
```

Out[201]:

```
0.36933751712555796
```

```
In [202]: Pos_df['3P%'].min()

Out[202]: 0.336734693877551
```

15. Find the best scorers in each team

In [206]:
```
df.iloc[0].T
```

Out[206]:
```
Player                 Alex Abrines
Pos                              SG
Age                            23.0
Tm                              OKC
G                              68.0
GS                              6.0
MP                           1055.0
ORB%                            1.9
DRB%                            7.1
TRB%                            4.5
AST%                            5.5
STL%                            1.7
BLK%                            0.6
TOV%                            8.3
FG                            134.0
FGA                           341.0
3P                             94.0
3PA                           247.0
2P                             40.0
2PA                            94.0
2P%                           0.426
FT                             44.0
FTA                            49.0
ORB                            18.0
DRB                            68.0
TRB                            86.0
AST                            40.0
STL                            37.0
BLK                             8.0
TOV                            33.0
PF                            114.0
PTS                           406.0
year_start                     2017
year_end                       2018
position                        G-F
height                          6-6
weight                        190.0
birth_date      1993-08-01 00:00:00
```

```
college                         NaN
Team           Oklahoma City Thunder
Name: 0, dtype: object
```

In [211]:
```
df['Best_score_per_team'] = df.groupby('Team')['PTS'].transform('max')
df.head()
```

Out[211]:

	Player	Pos	Age	Tm	G	GS	MP	ORB%	DRB%	TRB%	...	PTS	year_start	year_end	position	height	weight	birth_date	col
0	Alex Abrines	SG	23.0	OKC	68.0	6.0	1055.0	1.9	7.1	4.5	...	406.0	2017	2018	G-F	6-6	190.0	1993-08-01	
1	Quincy Acy	PF	26.0	TOT	38.0	1.0	558.0	3.9	18.0	11.0	...	222.0	2013	2018	F	6-7	240.0	1990-10-06	B Unive
2	Quincy Acy	PF	26.0	DAL	6.0	0.0	48.0	4.6	15.2	9.7	...	13.0	2013	2018	F	6-7	240.0	1990-10-06	B Unive
3	Quincy Acy	PF	26.0	BRK	32.0	1.0	510.0	3.8	18.2	11.1	...	209.0	2013	2018	F	6-7	240.0	1990-10-06	B Unive
4	Steven Adams	C	23.0	OKC	80.0	80.0	2389.0	13.0	15.5	14.2	...	905.0	2014	2018	C	7-0	255.0	1993-07-20	Unive Pittsb

5 rows × 41 columns

```
In [223]: best_scorers_per_team = df.loc[df['PTS']==df['Best_score_per_team'],['Player','Team','Pos','PTS']].sort_valu
          best_scorers_per_team = best_scorers_per_team.loc[best_scorers_per_team['Team']!='TOT']
          best_scorers_per_team
```

```
Out[223]:
                   Player                    Team  Pos     PTS
     567    Russell Westbrook   Oklahoma City Thunder   PG  2558.0
     214         James Harden         Houston Rockets   PG  2356.0
     525       Isaiah Thomas          Boston Celtics   PG  2199.0
     122       Anthony Davis    New Orleans Pelicans    C  2099.0
     538   Karl-Anthony Towns  Minnesota Timberwolves    C  2061.0
     331      Damian Lillard  Portland Trail Blazers   PG  2024.0
     130      DeMar DeRozan         Toronto Raptors   SG  2020.0
     120       Stephen Curry   Golden State Warriors   PG  1999.0
     274        LeBron James     Cleveland Cavaliers   SF  1954.0
     324       Kawhi Leonard       San Antonio Spurs   SF  1888.0
      19 Giannis Antetokounmpo     Milwaukee Bucks   SF  1832.0
     558        Kemba Walker       Charlotte Hornets   PG  1830.0
      79        Jimmy Butler           Chicago Bulls   SF  1816.0
     559           John Wall      Washington Wizards   PG  1805.0
     185         Paul George          Indiana Pacers   SF  1775.0
      62        Devin Booker            Phoenix Suns   SG  1726.0
      20     Carmelo Anthony         New York Knicks   SF  1659.0
     229      Gordon Hayward               Utah Jazz   SF  1601.0
     336         Brook Lopez           Brooklyn Nets    C  1539.0
     111    DeMarcus Cousins        Sacramento Kings    C  1528.0
      31     Harrison Barnes        Dallas Mavericks   PF  1518.0
```

	Player	Team	Pos	PTS
136	Goran Dragic	Miami Heat	PG	1483.0
180	Marc Gasol	Memphis Grizzlies	C	1446.0
488	Dennis Schroder	Atlanta Hawks	PG	1414.0
222	Tobias Harris	Detroit Pistons	PF	1321.0
209	Blake Griffin	Los Angeles Clippers	PF	1316.0
289	Nikola Jokic	Denver Nuggets	C	1221.0
101	Jordan Clarkson	Los Angeles Lakers	SG	1205.0

```
In [224]: best_scorers_per_team.shape
```

```
Out[224]: (30, 4)
```

16. Which team has the 'youngest squad', by average player age?

In [228]:

```
s1 = df['Team'].value_counts()
s1
```

Out[228]:

```
Team
TOT                       54
New Orleans Pelicans      27
Dallas Mavericks          24
Philadelphia 76ers        22
Atlanta Hawks             22
Cleveland Cavaliers       22
Brooklyn Nets             21
Milwaukee Bucks           20
Los Angeles Lakers        19
Charlotte Hornets         19
Orlando Magic             19
Denver Nuggets            19
Oklahoma City Thunder     19
Sacramento Kings          19
Phoenix Suns              18
Chicago Bulls             18
Washington Wizards        18
Houston Rockets           18
San Antonio Spurs         17
Golden State Warriors     17
Toronto Raptors           17
Indiana Pacers            17
Memphis Grizzlies         17
Minnesota Timberwolves    16
New York Knicks           16
Miami Heat                15
Detroit Pistons           15
Utah Jazz                 15
Boston Celtics            15
Portland Trail Blazers    15
Los Angeles Clippers      15
Name: count, dtype: int64
```

```
In [229]: s2 = df.groupby('Team')['Age'].sum()
          s2
```

```
Out[229]: Team
          Atlanta Hawks              624.0
          Boston Celtics             379.0
          Brooklyn Nets              542.0
          Charlotte Hornets          490.0
          Chicago Bulls              466.0
          Cleveland Cavaliers        667.0
          Dallas Mavericks           642.0
          Denver Nuggets             484.0
          Detroit Pistons            382.0
          Golden State Warriors      474.0
          Houston Rockets            468.0
          Indiana Pacers             454.0
          Los Angeles Clippers       443.0
          Los Angeles Lakers         505.0
          Memphis Grizzlies          463.0
          Miami Heat                 399.0
          Milwaukee Bucks            518.0
          Minnesota Timberwolves     411.0
          New Orleans Pelicans       701.0
          New York Knicks            426.0
          Oklahoma City Thunder      493.0
          Orlando Magic              484.0
          Philadelphia 76ers         548.0
          Phoenix Suns               462.0
          Portland Trail Blazers     365.0
          Sacramento Kings           508.0
          San Antonio Spurs          493.0
          TOT                       1434.0
          Toronto Raptors            427.0
          Utah Jazz                  393.0
          Washington Wizards         462.0
          Name: Age, dtype: float64
```

In [231]:

```
Age_df = pd.concat([s1,s2],axis=1)
Age_df
```

Out[231]:

	count	Age
Team		
TOT	54	1434.0
New Orleans Pelicans	27	701.0
Dallas Mavericks	24	642.0
Philadelphia 76ers	22	548.0
Atlanta Hawks	22	624.0
Cleveland Cavaliers	22	667.0
Brooklyn Nets	21	542.0
Milwaukee Bucks	20	518.0
Los Angeles Lakers	19	505.0
Charlotte Hornets	19	490.0
Orlando Magic	19	484.0
Denver Nuggets	19	484.0
Oklahoma City Thunder	19	493.0
Sacramento Kings	19	508.0
Phoenix Suns	18	462.0
Chicago Bulls	18	466.0
Washington Wizards	18	462.0
Houston Rockets	18	468.0
San Antonio Spurs	17	493.0
Golden State Warriors	17	474.0
Toronto Raptors	17	427.0

	count	Age
Team		
Indiana Pacers	17	454.0
Memphis Grizzlies	17	463.0
Minnesota Timberwolves	16	411.0
New York Knicks	16	426.0
Miami Heat	15	399.0
Detroit Pistons	15	382.0
Utah Jazz	15	393.0
Boston Celtics	15	379.0

In [234]:

```
Age_df['Avg_Age'] = Age_df['Age']/Age_df['count']
Age_df.sort_values('Avg_Age',ascending=True)
```

Out[234]:

	count	Age	Avg_Age
Team			
Portland Trail Blazers	15	365.0	24.333333
Philadelphia 76ers	22	548.0	24.909091
Toronto Raptors	17	427.0	25.117647
Boston Celtics	15	379.0	25.266667
Detroit Pistons	15	382.0	25.466667
Orlando Magic	19	484.0	25.473684
Denver Nuggets	19	484.0	25.473684
Phoenix Suns	18	462.0	25.666667
Washington Wizards	18	462.0	25.666667
Minnesota Timberwolves	16	411.0	25.687500
Charlotte Hornets	19	490.0	25.789474
Brooklyn Nets	21	542.0	25.809524
Chicago Bulls	18	466.0	25.888889
Milwaukee Bucks	20	518.0	25.900000
Oklahoma City Thunder	19	493.0	25.947368
New Orleans Pelicans	27	701.0	25.962963
Houston Rockets	18	468.0	26.000000
Utah Jazz	15	393.0	26.200000
TOT	54	1434.0	26.555556
Los Angeles Lakers	19	505.0	26.578947
Miami Heat	15	399.0	26.600000

	count	Age	Avg_Age
Team			
New York Knicks	16	426.0	26.625000
Indiana Pacers	17	454.0	26.705882
Sacramento Kings	19	508.0	26.736842
Dallas Mavericks	24	642.0	26.750000
Memphis Grizzlies	17	463.0	27.235294
Golden State Warriors	17	474.0	27.882353
Atlanta Hawks	22	624.0	28.363636
San Antonio Spurs	17	493.0	29.000000

```
In [ ]:
```