NBA Questions: Asked & Answers

Daniel Haven, Aaron Oyer, Mark Stalnaker

We wanted to explore the relationships between salary, performance and popularity of NBA players. We used 2016-2017 season data for individual players and teams to answer various questions about these relationships. We are all NBA fans and are interested in how analytics are changing the game.

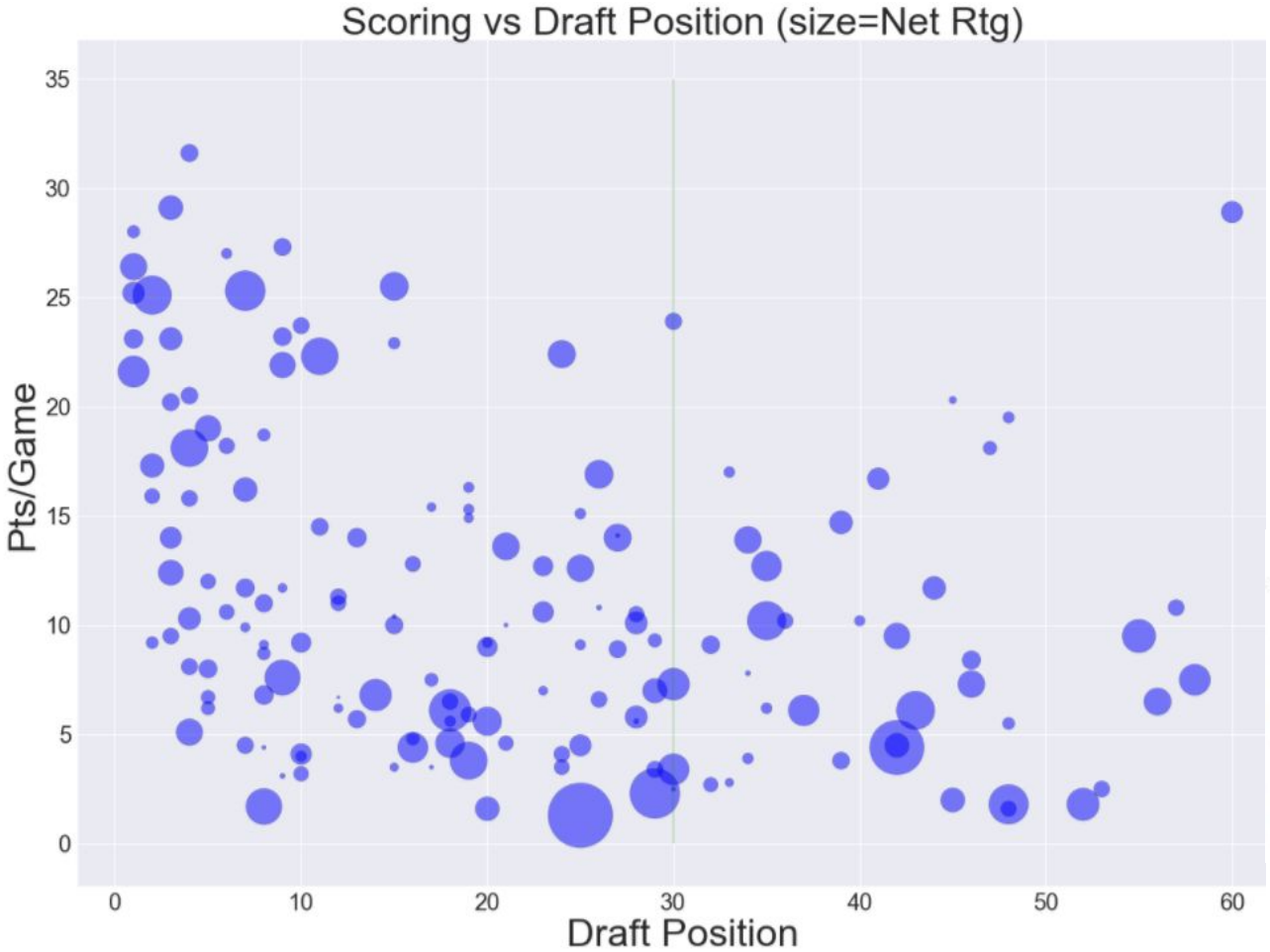
Data:

- Used multiple sets of NBA data found on Kaggle
- Twitter API

Issues:

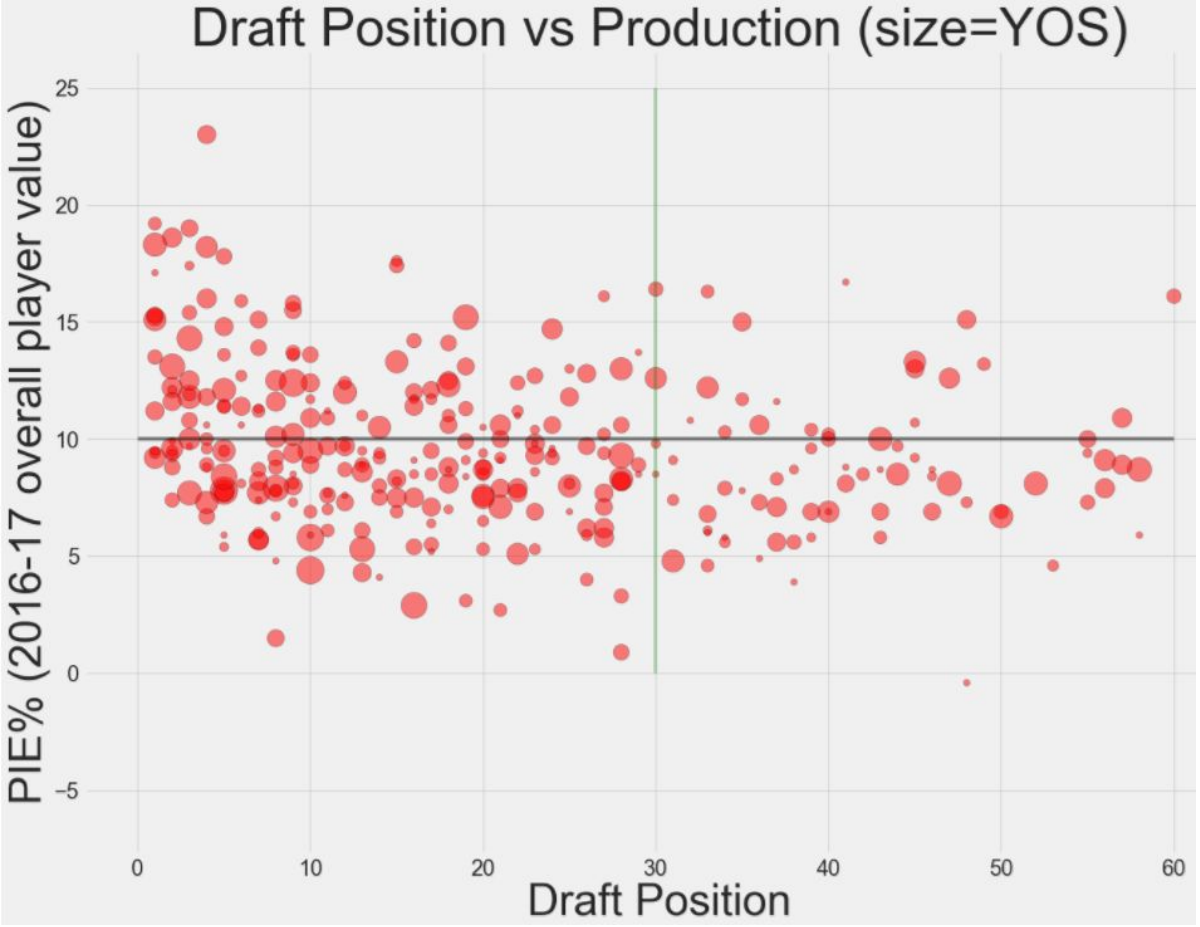
- Players listed multiple times (if traded in-season)
- Players listed with blank stats
- Drafted vs undrafted players
- Mapping issues

Where are the best scorers drafted?



	Draft Status
0	2013 Rnd 2 Pick 2
1	2012 Rnd 2 Pick 7
2	2013 Rnd 1 Pick 12
3	2007 Rnd 1 Pick 27
4	2008 Rnd 1 Pick 20

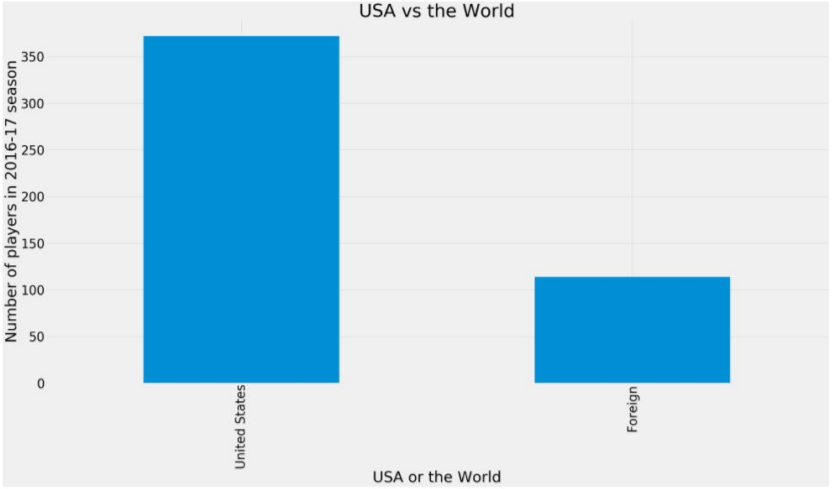
	year	round	round#	pick	pick#
0	2013	Rnd	30	Pick	2
1	2012	Rnd	30	Pick	7
2	2013	Rnd	0	Pick	12
3	2007	Rnd	0	Pick	27
4	2008	Rnd	0	Pick	20



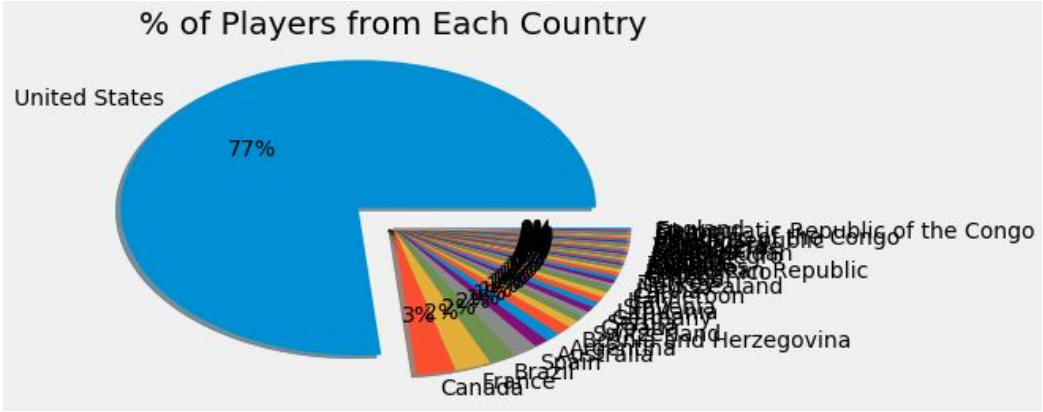

```
plt.style.use('fivethirtyeight')

PIE = new_df['PIE']
draft_pos = new_df['drafted']
size = new_df['YOS']


plt.xlim(0,60)
plt.ylim(0,25)
plt.figure(figsize=(20,15))
plt.rcParams.update({'font.size': 40})

plt.hlines(10, 0, 60, alpha=.5)
plt.vlines(30, 0, 25, color='g', alpha=.25)
plt.tick_params(axis = 'x', labelsize = 24)
plt.tick_params(axis = 'y', labelsize = 24)

plt.xlabel("Draft Position")
plt.ylabel("PIE% (2016-17 overall player value)")
plt.title("Draft Position vs Production (size=YOS)")

plt.scatter(draft_pos, PIE, color='r', marker='o',
            s=size*50, edgecolors='black', alpha=.5)
plt.savefig("PIE vs Draft Pos.png")
plt.show()
```

From NBA.com: **"In its simplest terms, PIE shows what % of game events did that player or team achieve. The stats being analyzed are your traditional basketball statistics. A player that achieves more than 10% is likely to be better than the average player. A high PIE % is highly correlated to winning. In fact, a team's PIE rating and a team's winning percentage correlate at an R square of .908 which indicates a "strong" correlation."**

Where do NBA players come from?



- Hoped to show what countries NBA players are from in a map
- Unfortunately ran into multiple issues doing so
- Another tool (such as Tableau) would have been better to show this
- Even with influx of foreign players in last 10 years, USA still dominates the league with 77% of players as of 2016-17 season

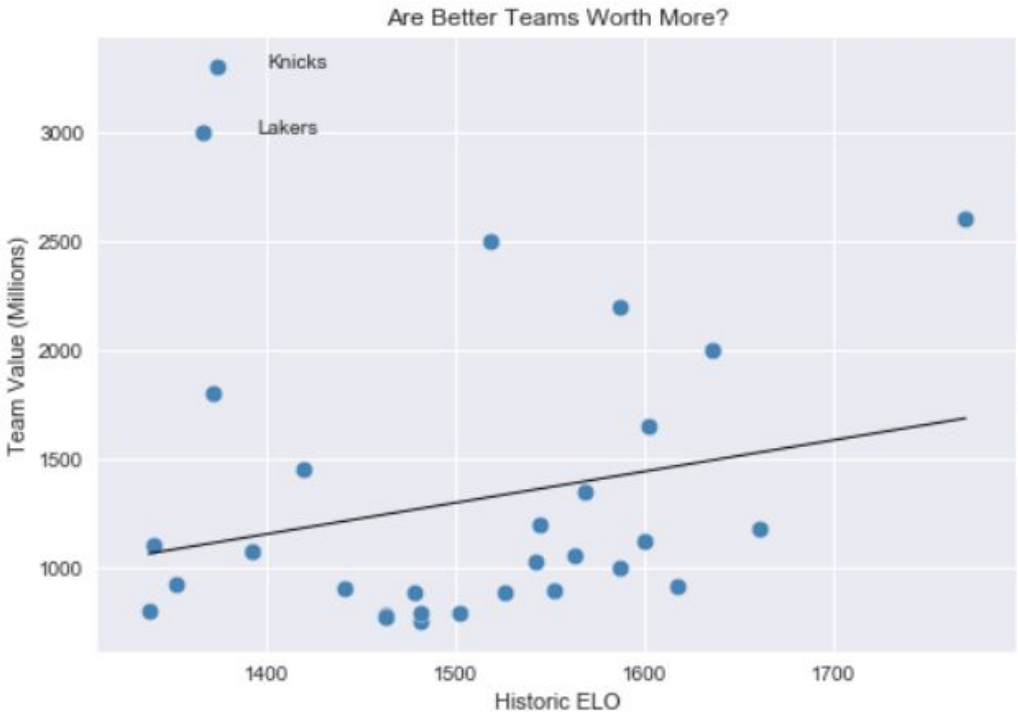


United States	372
Canada	13
France	12
Brazil	9
Spain	8
Australia	5
Argentina	5
Bosnia and Herzegovina	5
Switzerland	4
Croatia	3

Basic Conf Metrics

Conference	Conference Value (Millions)	Total Conference Elo	Average Conference Elo
East	20425	22254	1483.600000
West	20235	22891	1526.066667

Are Better Teams Worth More?




```
print('Slope: %.3f' % model.coef_[0])
print('Intercept: %.3f' % model.intercept_)
```

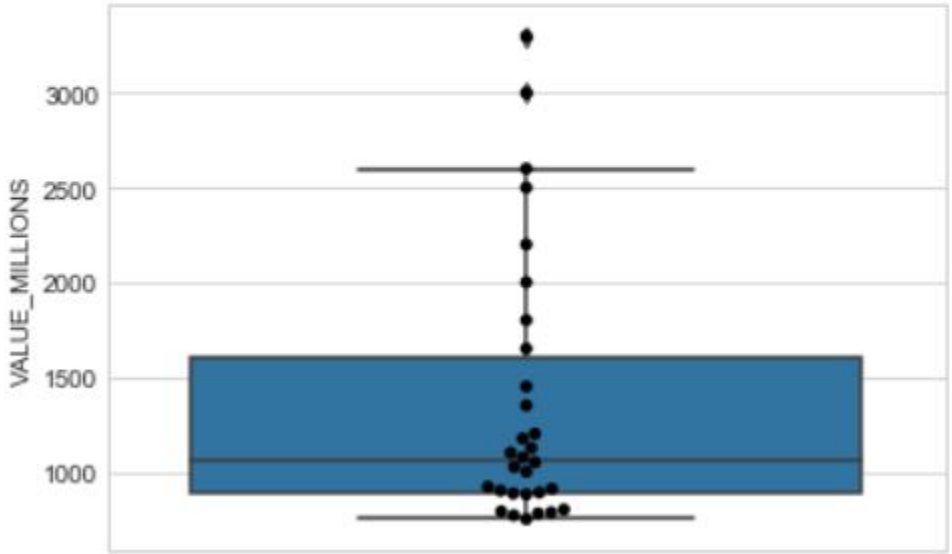
```
Slope: 1.442
Intercept: -866.985
```

```
print(model.score(X_train, y_train))
```

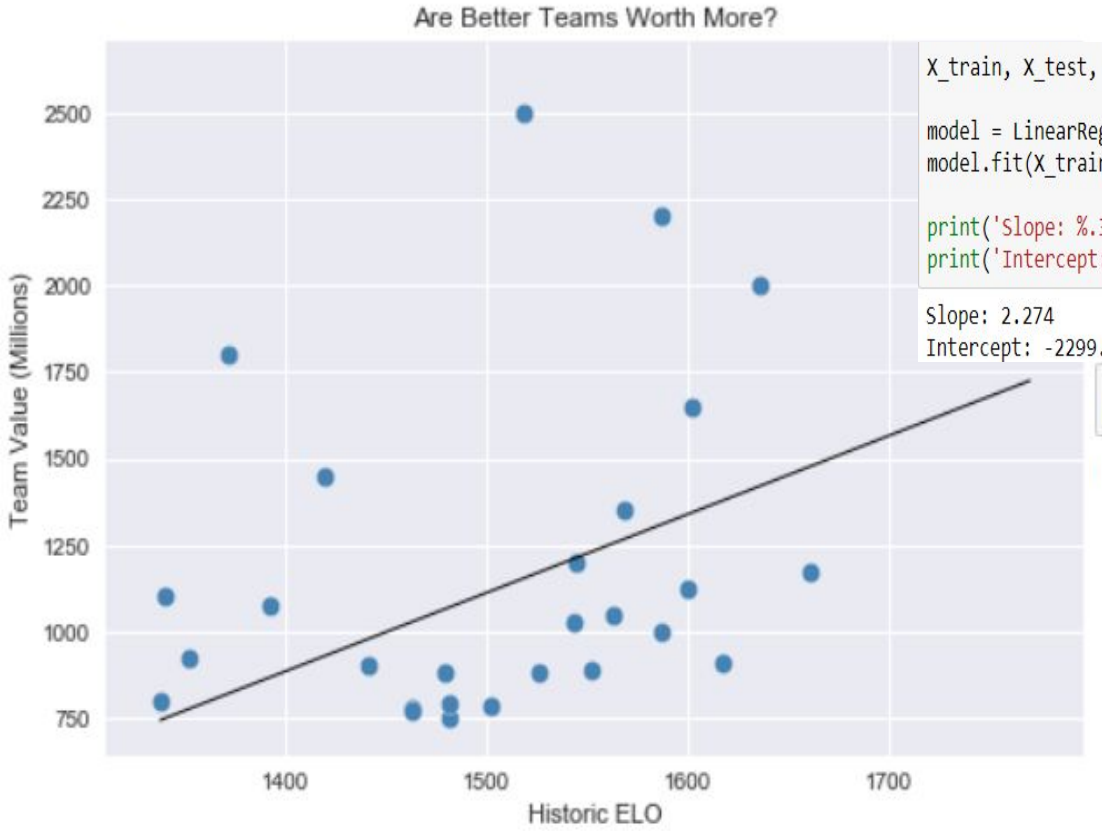
```
0.0702142508063
```

Identifying and Removing Outliers

```
sns.set_style("whitegrid")
ax = sns.boxplot(y=nba_df['VALUE_MILLIONS'])
ax = sns.swarmplot(y=nba_df['VALUE_MILLIONS'], color='k')
plt.show()
```



Rerun Regression without Outliers




```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=2)

model = LinearRegression()
model.fit(X_train, y_train)

print('Slope: %.3f' % model.coef_[0])
print('Intercept: %.3f' % model.intercept_)
```

```
Slope: 2.274
Intercept: -2299.056
```

```
print(model.score(X_train, y_train))
```

```
0.280747109188
```

Multivariable Regression

```
attributes = ['ELO', 'POP_MILLIONS', 'REVENUE_MILLIONS', 'TOTAL']
X = nba_df[attributes]
Y = nba_df['VALUE_MILLIONS']
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=2)
model2 = LinearRegression()
model2.fit(X_train, y_train)
```

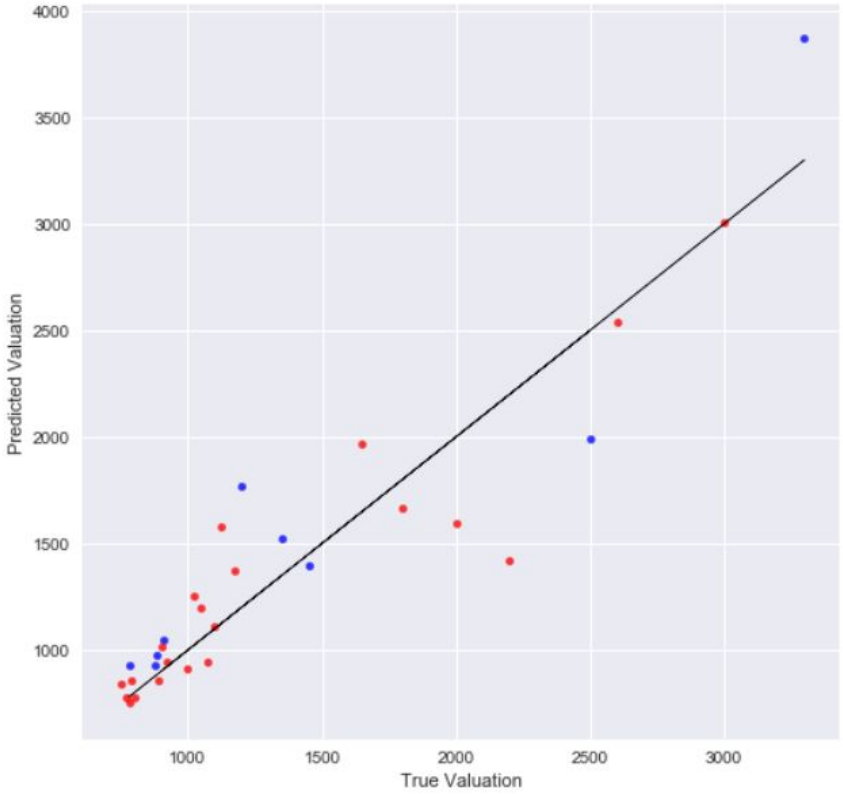
```
print('Slope: %.3f' % model2.coef_[0])
print('Intercept: %.3f' % model2.intercept_)
```

```
Slope: 0.307
Intercept: -1707.546
```

```
print(model2.score(X_train, y_train))
```

```
0.849472814933
```

Multivariable Regression




```
def TrueValue(X_test, y_test, X_train, y_train, model):
    fig = plt.figure(figsize=(8,8))
    plt.scatter(y_test, model.predict(X_test), alpha = 0.8, s=20, color='blue')
    plt.scatter(y_train, model.predict(X_train), alpha = 0.8, s=20, color='red')
    plt.plot(y_test, y_test, color='black', lw=0.5)
    plt.xlabel('True Valuation')
    plt.ylabel('Predicted Valuation')
    plt.style.use('seaborn')
```

```
estimate = model2.predict([[1350, 3, 250, 80000]])
estimate
```

```
array([ 1566.06137913])
```

NBA “Bigs”: Role of C’s and PF’s changing to beyond the arc

Over the last few seasons, the theory is that there has been a sea change in the role of bigs in the NBA- a switch from guys playing close to the basket with an emphasis on rebounds and blocks to the value being moved to players who can shoot 3s.

Pulled data from last seven seasons, using only players whose position was listed as PF or C (i.e. “Bigs”), as they historically take few 3-point shots and play near the basket

In [126]:
```
data_2011 = os.path.join("2011_nba_bigs_data.csv")
data_2012 = os.path.join("2012_nba_bigs_data.csv")
data_2013 = os.path.join("2013_nba_bigs_data.csv")
data_2014 = os.path.join("2014_nba_bigs_data.csv")
data_2015 = os.path.join("2015_nba_bigs_data.csv")
data_2016 = os.path.join("2016_nba_bigs_data.csv")
data_2017 = os.path.join("2017_nba_bigs_data_formatted.csv")
orig_2011 = pd.read_csv(data_2011)
orig_2012 = pd.read_csv(data_2012)
orig_2013 = pd.read_csv(data_2013)
orig_2014 = pd.read_csv(data_2014)
orig_2015 = pd.read_csv(data_2015)
orig_2016 = pd.read_csv(data_2016)
orig_2017 = pd.read_csv(data_2017)
orig_2017.head()
```

Out[126]:

	Player	POSITION	Age	Tm	G	GS	MP	FG	FGA	FG%	...	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PS/G	2017 Salary
0	Anthony Davis	C	23	NO	75	NaN	36.1	10.3	20.3	0.505	...	2.3	9.5	11.8	2.1	1.3	2.2	2.4	2.2	28.0	22116750.0
1	DeMarcus Cousins	C	26	NO/SAC	72	NaN	34.2	9.0	19.9	0.452	...	2.1	8.9	11.0	4.6	1.4	1.3	3.7	3.9	27.0	16957900.0
2	Karl-Anthony Towns	C	21	MIN	82	NaN	37.0	9.8	18.0	0.542	...	3.6	8.7	12.3	2.7	0.7	1.3	2.6	2.9	25.1	5960160.0
3	Blake Griffin	PF	27	LAC	61	NaN	34.0	7.9	15.9	0.493	...	1.8	6.3	8.1	4.9	0.9	0.4	2.3	2.6	21.6	20140838.0
4	Brook Lopez	C	28	BKN	75	NaN	29.6	7.4	15.6	0.474	...	1.6	3.8	5.4	2.3	0.5	1.7	2.5	2.6	20.5	21165675.0

5 rows × 30 columns

To get a true representation of trend, used only players who logged at least the league average in total minutes for a season

In [136]:
```
#Use only players who played greater than the season's Mean for Total Minutes Played (gets ~Top 75% of each season)
Min_Mean_2011 = condensed_2011[(condensed_2011['2011 Total Min'] >= [(condensed_2011['2011 Total Min'].mean())])]
Min_Mean_2012 = condensed_2012[(condensed_2012['2012 Total Min'] >= [(condensed_2012['2012 Total Min'].mean())])]
Min_Mean_2013 = condensed_2013[(condensed_2013['2013 Total Min'] >= [(condensed_2013['2013 Total Min'].mean())])]
Min_Mean_2014 = condensed_2014[(condensed_2014['2014 Total Min'] >= [(condensed_2014['2014 Total Min'].mean())])]
Min_Mean_2015 = condensed_2015[(condensed_2015['2015 Total Min'] >= [(condensed_2015['2015 Total Min'].mean())])]
Min_Mean_2016 = condensed_2016[(condensed_2016['2016 Total Min'] >= [(condensed_2016['2016 Total Min'].mean())])]
Min_Mean_2017 = condensed_2017[(condensed_2017['2017 Total Min'] >= [(condensed_2017['2017 Total Min'].mean())])]
Min_Mean_2011.head()
```

Out[136]:

	Player	Pos	2011 Age	2011 3P	2011 3PA	2011 3P%	2011 2PA	2011 Reb	2011 Blk	2011 Pts/Gm	2011 Salary	2011 Salary (MM)	2011 Total Min
2	LaMarcus Aldridge	PF	25	0.0	0.3	17.40	17.2	8.8	1.2	21.8	12372000	12.372	3207.6
4	Ryan Anderson	PF	22	2.1	5.3	39.30	2.9	5.5	0.6	10.6	2244600	2.245	1427.2
5	Joel Anthony	C	28	0.0	0.0	0.00	1.3	3.5	1.2	2.0	3600000	3.600	1462.5
6	Darrell Arthur	PF	22	0.0	0.1	0.00	7.7	4.3	0.8	9.1	2027118	2.027	1608.0
8	Andrea Bargnani	C	25	1.2	3.4	34.50	14.4	5.2	0.7	21.4	9250000	9.250	2356.2

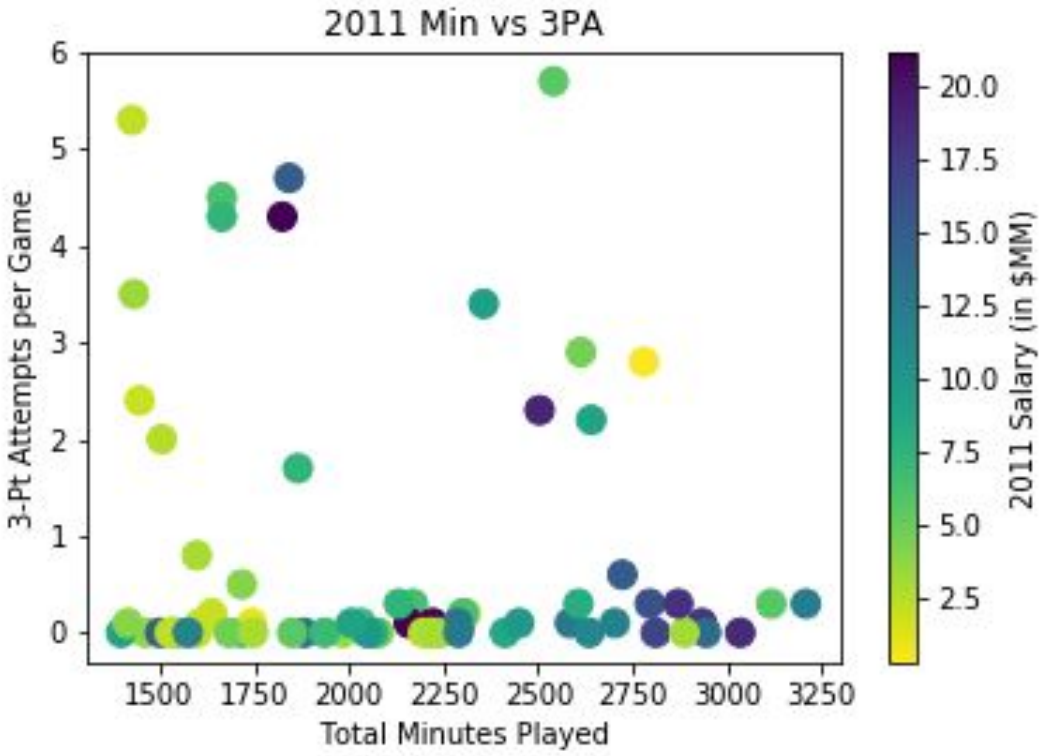
Used a “heat map” to show the correlation between the number of 3-point attempts by year vs the total season minutes played, using salary to show the players who are making the most.

In [139]:
```
plt.scatter(Min_Mean_2011['2011 Total Min'], Min_Mean_2011['2011 3PA'], c=Min_Mean_2011['2011 Salary (MM)'],
            cmap='viridis_r', s=100)
plt.title('2011 Min vs 3PA')
plt.xlabel('Total Minutes Played')
plt.ylabel('3-Pt Attempts per Game')
plt.colorbar(label='2011 Salary (in $MM)')
plt.savefig('2011 Data')
plt.show()
```

Also used value_counts to list number of qualifying players for the given year.

2011

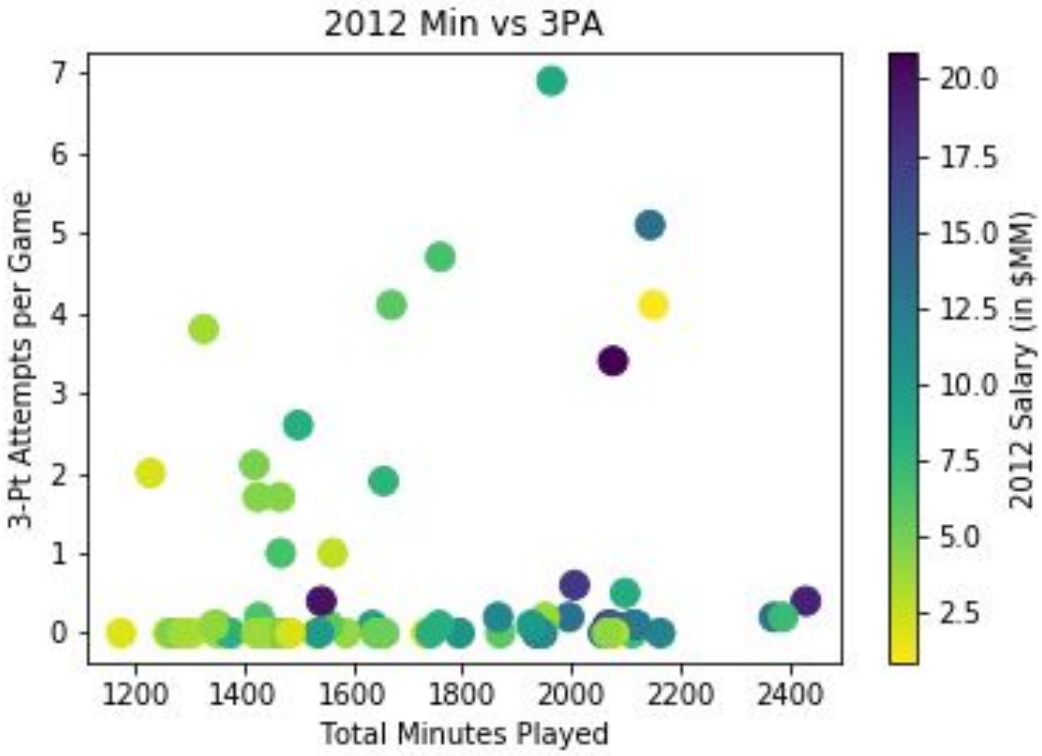
2011	
PF	40
C	32



2012

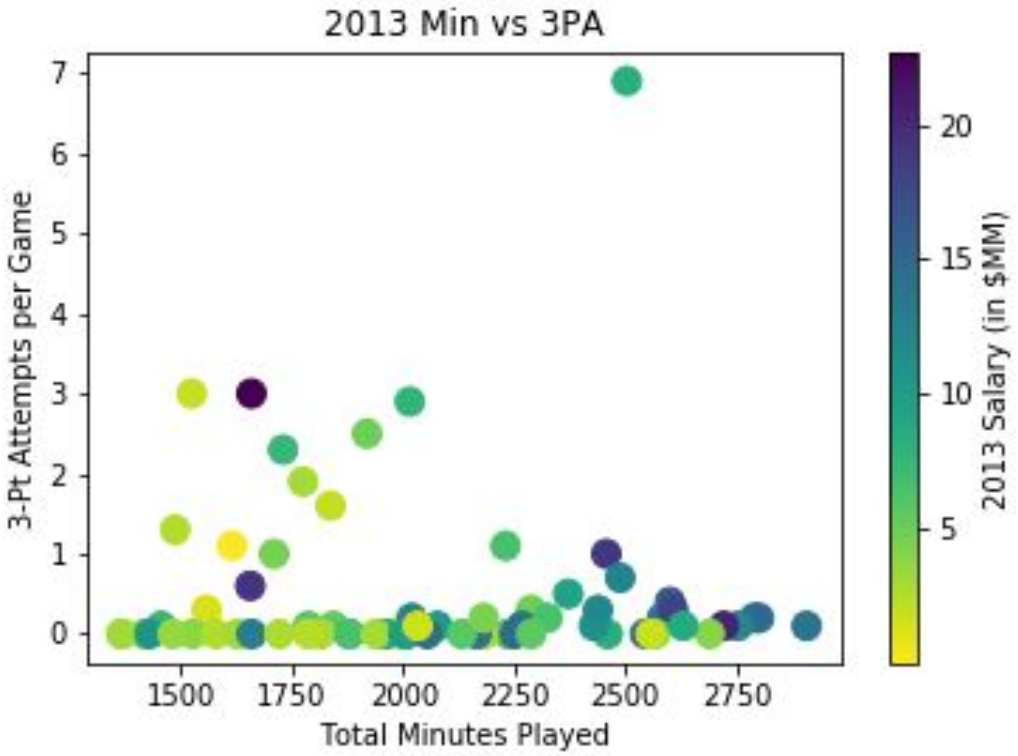
2012

PF	42
C	26



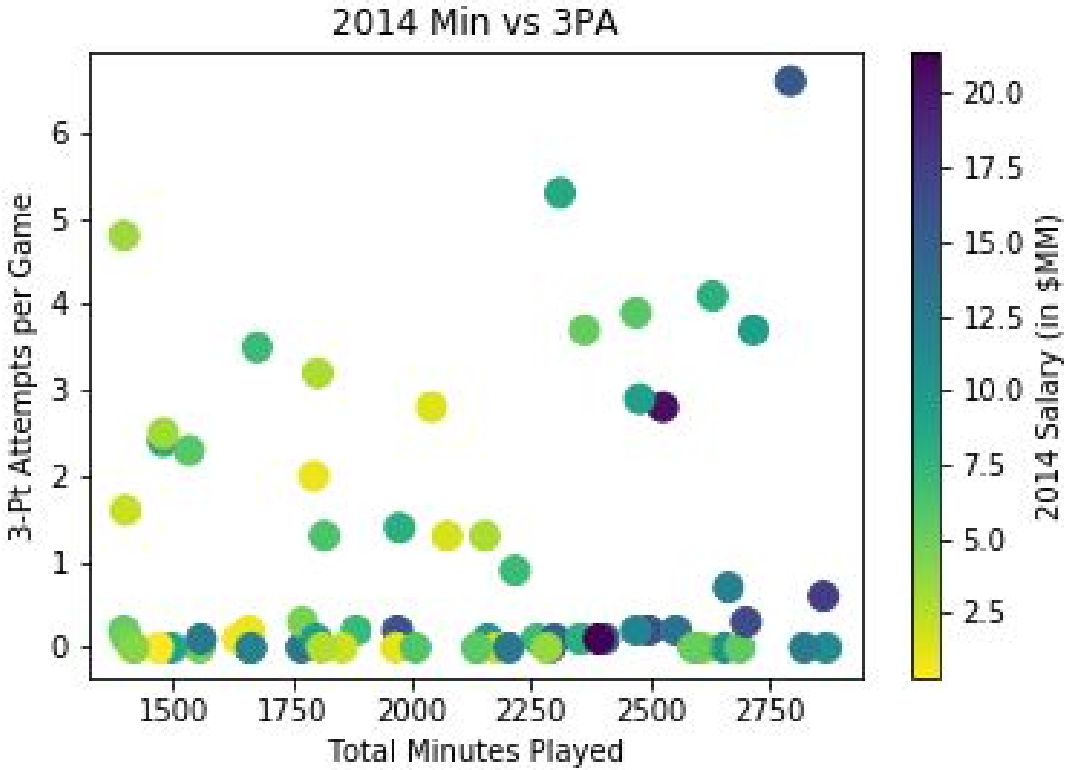
2013

2013	
PF	38
C	34



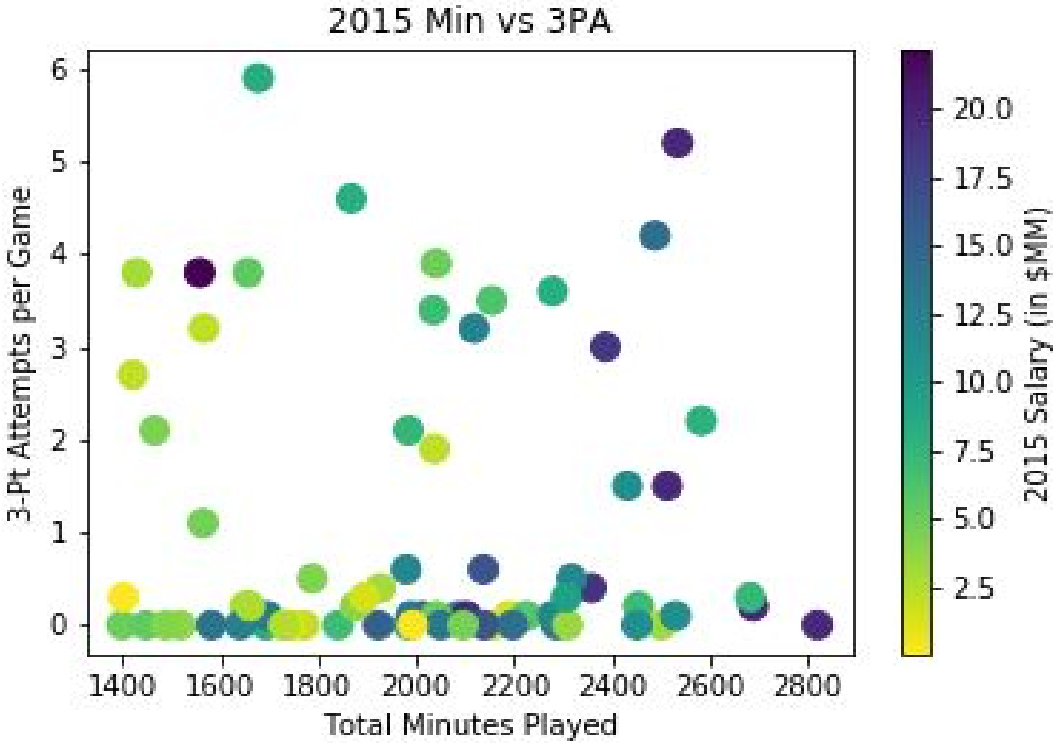
2014

2014	
PF	41
C	28



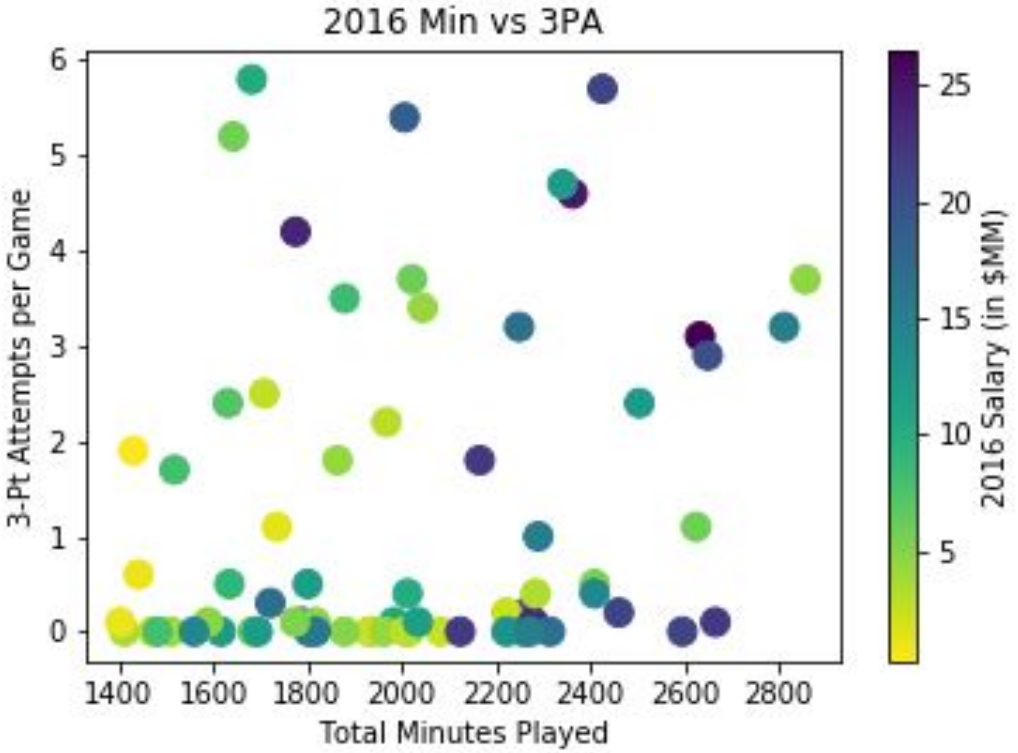
2015

2015	
PF	38
C	34



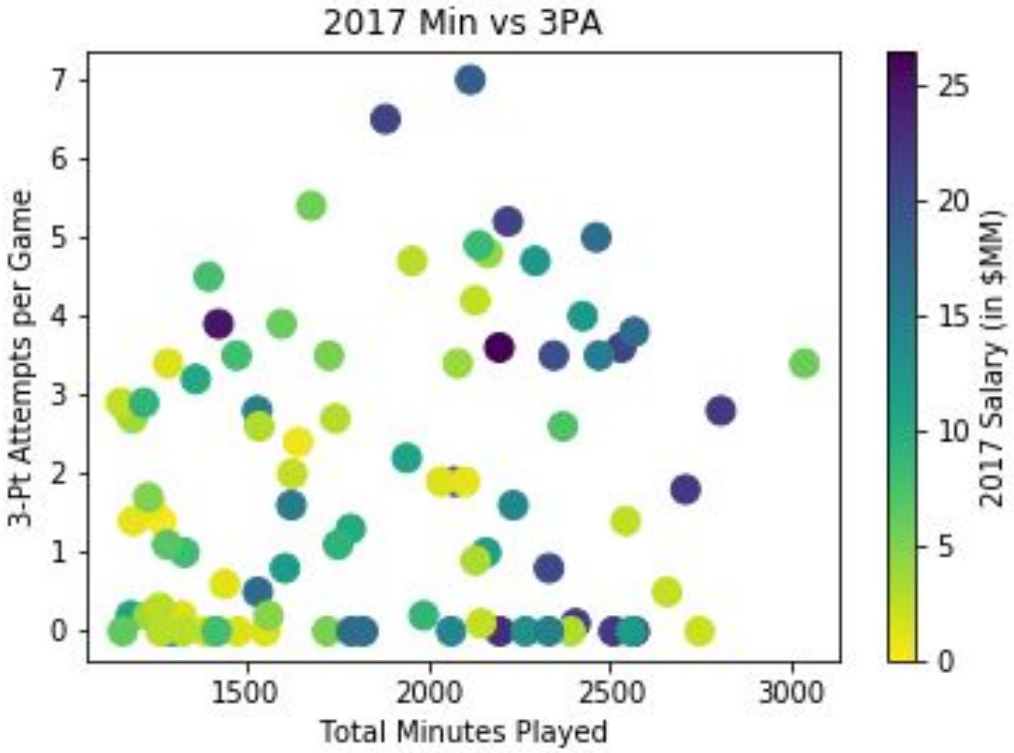
2016

2016	
C	36
PF	33



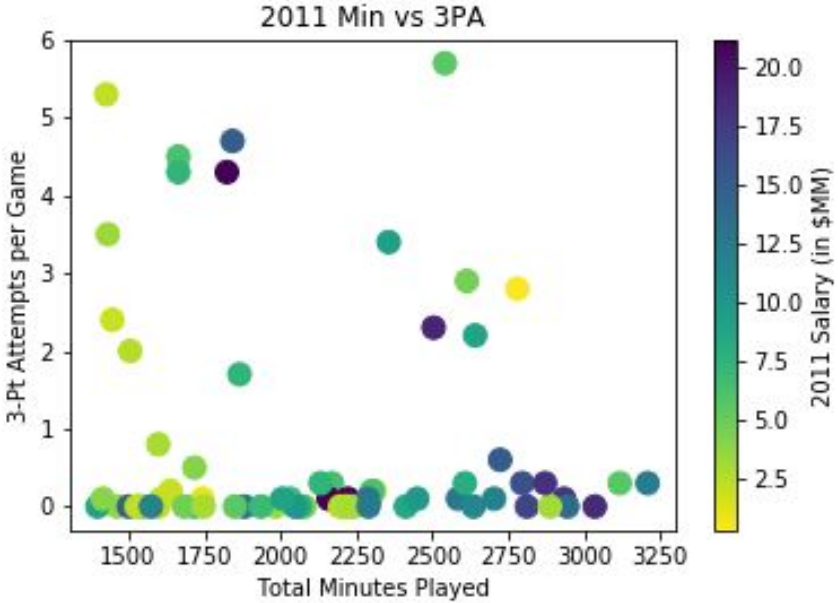
2017

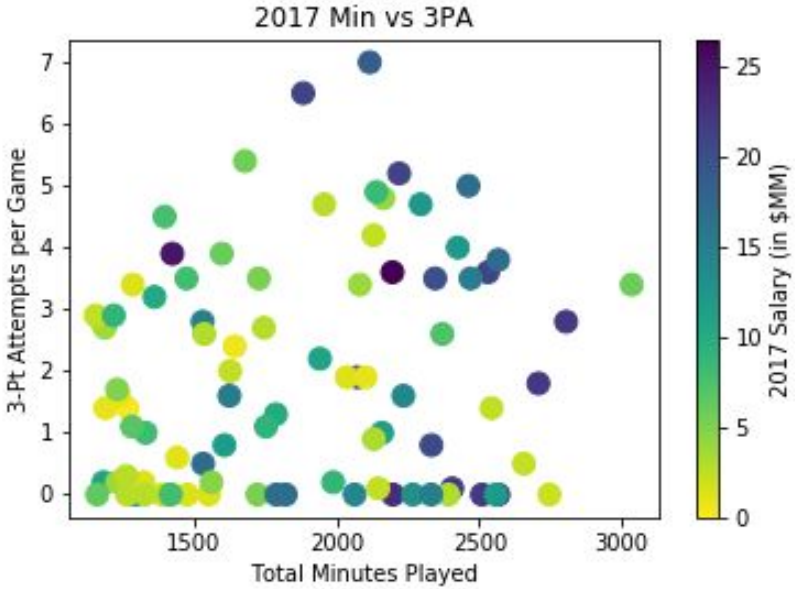
2017	
PF	46
C	43



2011-2017

Data shows a huge change in the role of NBA bigs





NBA Twitter: Who tweets most, Twitter followers vs Min/Gm or $$$

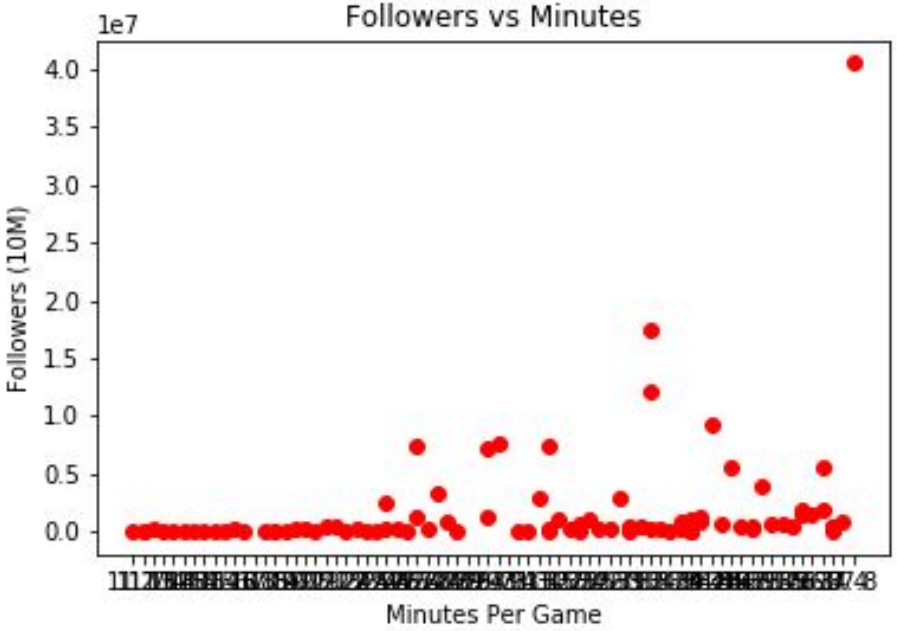
Any correlation in player minutes or player salaries and the number of their Tweets

Used Twitter API and a Kaggle csv with around 100 players including their Twitter handle

In [8]:
```
for index, row in nba_tweeters.iterrows():

    try:
        target_user = row["TWITTER_HANDLE"]

        user_account = api.get_user(target_user)
        user_real_name = user_account["name"]
        print(user_real_name)


        tweets = user_account["statuses_count"]
        followers = user_account['followers_count']
        favorites = user_account['favourites_count']
        friends = user_account['friends_count']


        nba_tweeters.set_value(index, "Real Name", user_real_name)
        nba_tweeters.set_value(index, "Tweets", tweets)
        nba_tweeters.set_value(index, "Followers", followers)
        nba_tweeters.set_value(index, "Favorites Count", favorites)
        nba_tweeters.set_value(index, "Following", friends)

    except tweepy.TweepError:
        print ("target user skipped: " + target_user)
        continue

nba_tweeters.to_csv("project_nba_twitter.csv", index = False)
```

Plotted graph using minutes plus Twitter followers

```
In [104]: followers = twitter_sort['Followers']
          minutes = twitter_sort['2017 Min/Gm']
          tweets = twitter_sort['Tweets']
          plt.scatter(minutes, followers, color='red')
          plt.title('Followers vs Minutes')
          plt.ylabel('Followers (10M)')
          plt.xlabel('Minutes Per Game')
          plt.savefig('MPG_v_Foll.png')
          plt.show()
```

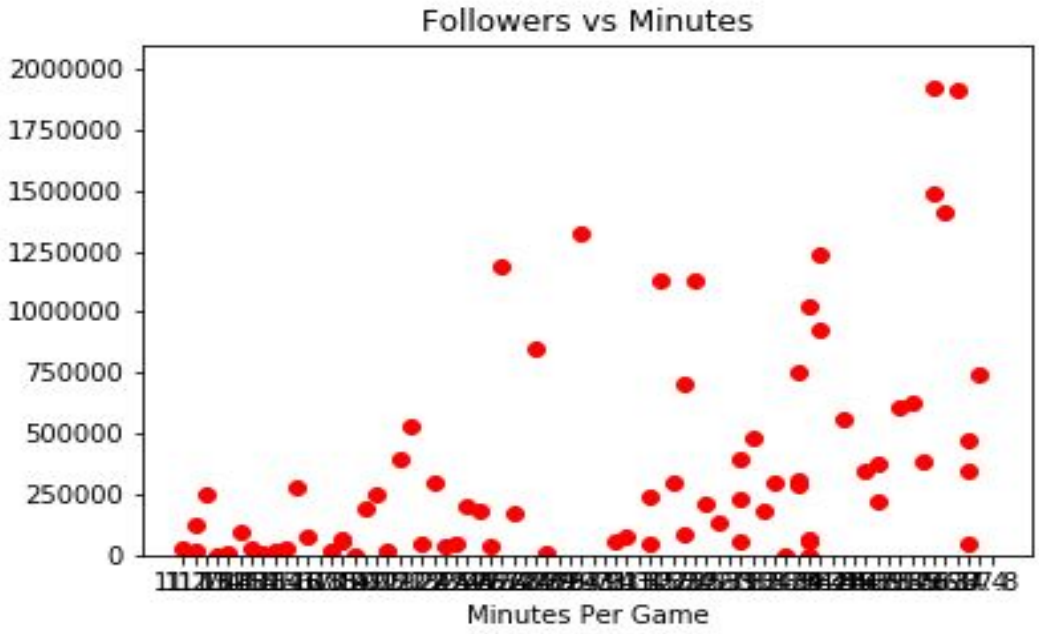


Thanks, LeBron!

One outlier skewed the whole graph and made it unreadable

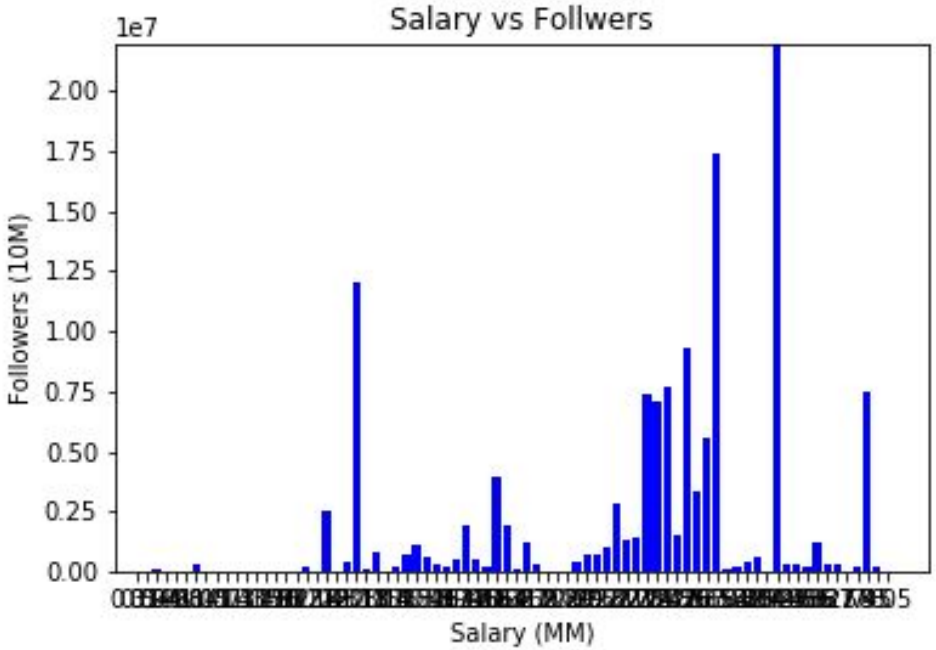
Removing LeBron James (>40M followers, almost double 2nd place), we get a much more readable plot

```
In [105]: #Removed LeBron
plt.scatter(minutes, followers, color='red')
plt.title('Followers vs Minutes')
plt.ylabel('Followers (10M)')
plt.ylim(0,2100000)
plt.xlabel('Minutes Per Game')
plt.savefig('MPG_v_Foll_no_King.png')
plt.show()
```



As expected, more minutes played starts to equal more followers on Twitter

Again removing LeBron, did the same for salary vs followers



Top salaries weren't necessarily the most followed... around the 67th percentile was the winner in follows.

Thanks for listening!

Now, time for questions...