NBA Salary Predictor

Brandon Rufino

06/12/2019

SECTION 1: Overview

Hello all! As a student in Toronto and a long time NBA fan, basketball in the North has never been sweeter. That being said, I got some serious concerns. That is, teams continually tie themselves to players via large contracts when the performance of a player may not indicate they deserve said salary. Thus, taking inspiration from Koki Ando on Kaggle I will create a NBA Salary Predictor using simple regression models.

To be specific the goal is to look at the season performance in 2016-2017 to predict the salary they should make in a season.

1.1 Problem statement

As stated in the overview our goal is to look at the season performance 2016-2017 to predict the salary they should make in a season. We will perform the following steps in this project: (1) load and clean our data, (2) data preprocessing, (3) data exploration, and (4) creating a few regression models.

1.2 SETUP: Load required packages

Before we begin our data crunch let us load all required packages and the data-set.

1.3 SETUP: Prepare data

Here we prepare 2 datasets. The first dataset salary_table which was scraped from Koki Ando https://github.com/koki25ando/NBA-Players-2017-18-dataset. This dataset contains salaries of the 2017-2018 season. The second dataset, stats, is the NBA players season stats since the 1950 season.

salary_table looks like this:

```
head(salary_table)
```

```
##   X         Player  Tm season17_18
## 1 1  Stephen Curry GSW    34682550
## 2 2   LeBron James CLE    33285709
## 3 3   Paul Millsap DEN    31269231
## 4 4 Gordon Hayward BOS    29727900
## 5 5  Blake Griffin DET    29512900
## 6 6     Kyle Lowry TOR    28703704
```

stats looks like this:

```
head(stats)
```

```
##   X Year           Player Pos Age  Tm  G GS MP PER   TS. X3PAr   FTr ORB. DRB.
## 1 0 1950 Curly Armstrong G-F  31 FTW 63 NA NA  NA 0.368    NA 0.467   NA   NA
## 2 1 1950    Cliff Barker  SG  29 INO 49 NA NA  NA 0.435    NA 0.387   NA   NA
## 3 2 1950   Leo Barnhorst  SF  25 CHS 67 NA NA  NA 0.394    NA 0.259   NA   NA
## 4 3 1950      Ed Bartels   F  24 TOT 15 NA NA  NA 0.312    NA 0.395   NA   NA
## 5 4 1950      Ed Bartels   F  24 DNN 13 NA NA  NA 0.308    NA 0.378   NA   NA
## 6 5 1950      Ed Bartels   F  24 NYK  2 NA NA  NA 0.376    NA 0.750   NA   NA
##   TRB. AST. STL. BLK. TOV. USG. blanl  OWS  DWS   WS WS.48 blank2 OBPM DBPM BPM
## 1   NA   NA   NA   NA   NA   NA    NA -0.1  3.6  3.5    NA     NA   NA   NA  NA
## 2   NA   NA   NA   NA   NA   NA    NA  1.6  0.6  2.2    NA     NA   NA   NA  NA
## 3   NA   NA   NA   NA   NA   NA    NA  0.9  2.8  3.6    NA     NA   NA   NA  NA
## 4   NA   NA   NA   NA   NA   NA    NA -0.5 -0.1 -0.6    NA     NA   NA   NA  NA
## 5   NA   NA   NA   NA   NA   NA    NA -0.5 -0.1 -0.6    NA     NA   NA   NA  NA
## 6   NA   NA   NA   NA   NA   NA    NA  0.0  0.0  0.0    NA     NA   NA   NA  NA
##   VORP  FG FGA   FG. X3P X3PA X3P. X2P X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB
## 1   NA 144 516 0.279  NA   NA   NA 144  516 0.279 0.279 170 241 0.705  NA  NA
## 2   NA 102 274 0.372  NA   NA   NA 102  274 0.372 0.372  75 106 0.708  NA  NA
## 3   NA 174 499 0.349  NA   NA   NA 174  499 0.349 0.349  90 129 0.698  NA  NA
## 4   NA  22  86 0.256  NA   NA   NA  22   86 0.256 0.256  19  34 0.559  NA  NA
## 5   NA  21  82 0.256  NA   NA   NA  21   82 0.256 0.256  17  31 0.548  NA  NA
## 6   NA   1   4 0.250  NA   NA   NA   1    4 0.250 0.250   2   3 0.667  NA  NA
##   TRB AST STL BLK TOV  PF PTS
## 1  NA 176  NA  NA  NA 217 458
## 2  NA 109  NA  NA  NA  99 279
## 3  NA 140  NA  NA  NA 192 438
## 4  NA  20  NA  NA  NA  29  63
## 5  NA  20  NA  NA  NA  27  59
## 6  NA   0  NA  NA  NA   2   4
```

SECTION 2: Methods

In this section we will we will preprocess our data to extrapolate features that we may use in our model training. We will then explore our dataset. Lastly, we will describe the different models we will train and test.

2.1: Data preprocessing

Because the stats variable contains stats since the 1950 season let us filter to keep only the 2016-2017 season. Remember our goal is to look at the season performance (2016-2017) to predict the salary they should make in the 2017-2018 season. Some features we would like to include in our regression model is averages since we would like to gauge players who put up great numbers but are injured for majority of the season.

```r
# must mutate some columns to get MPG, PPG, APG, RPG, TOPG, BPG, SPG
stats_1617 <-
  stats %>% filter(Year >= 2017) %>%
  select(Year:G, MP, PER, FG:PTS) %>%
  distinct(Player, .keep_all = TRUE) %>%
  mutate(MPG = MP/G, PPG = PTS/G, APG = AST/G,
         RPG = TRB/G, TOPG = TOV/G, BPG = BLK/G,
        SPG = STL/G)
```

Lets take a look at our data:

```r
head(stats_1617)
```

```
##   Year          Player Pos Age  Tm  G   MP  PER  FG FGA   FG. X3P X3PA  X3P. X2P
## 1 2017   Alex Abrines  SG  23 OKC 68 1055 10.1 134 341 0.393  94  247 0.381  40
## 2 2017     Quincy Acy  PF  26 TOT 38  558 11.8  70 170 0.412  37   90 0.411  33
## 3 2017   Steven Adams   C  23 OKC 80 2389 16.5 374 655 0.571   0    1 0.000 374
## 4 2017 Arron Afflalo  SG  31 SAC 61 1580  9.0 185 420 0.440  62  151 0.411 123
## 5 2017 Alexis Ajinca   C  28 NOP 39  584 12.9  89 178 0.500   0    4 0.000  89
## 6 2017   Cole Aldrich   C  28 MIN 62  531 12.7  45  86 0.523   0    0    NA  45
##   X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB TRB AST STL BLK TOV  PF PTS       MPG
## 1   94 0.426 0.531  44  49 0.898  18  68  86  40  37   8  33 114 406 15.514706
## 2   80 0.413 0.521  45  60 0.750  20  95 115  18  14  15  21  67 222 14.684211
## 3  654 0.572 0.571 157 257 0.611 282 333 615  86  88  78 146 195 905 29.862500
## 4  269 0.457 0.514  83  93 0.892   9 116 125  78  21   7  42 104 515 25.901639
## 5  174 0.511 0.500  29  40 0.725  46 131 177  12  20  22  31  77 207 14.974359
## 6   86 0.523 0.523  15  22 0.682  51 107 158  25  25  23  17  85 105  8.564516
##         PPG       APG      RPG      TOPG       BPG       SPG
## 1  5.970588 0.5882353 1.264706 0.4852941 0.1176471 0.5441176
## 2  5.842105 0.4736842 3.026316 0.5526316 0.3947368 0.3684211
## 3 11.312500 1.0750000 7.687500 1.8250000 0.9750000 1.1000000
## 4  8.442623 1.2786885 2.049180 0.6885246 0.1147541 0.3442623
## 5  5.307692 0.3076923 4.538462 0.7948718 0.5641026 0.5128205
## 6  1.693548 0.4032258 2.548387 0.2741935 0.3709677 0.4032258
```

Great now we have our stats of the previous season (2016-2017) and our salary of the 2017-2018 season. Let us merge these datasets in order to make integration into our models easy:

```r
#merge by player and name column
stats1617_salary1718 <- merge(stats_1617, salary_table, by.x = "Player", by.y = "Player")
names(stats1617_salary1718)[40] <- "salary_1718"
#remove tm.y column
stats1617_salary1718 <- stats1617_salary1718[-39]
```

Lets take a look:

```r
head(stats1617_salary1718)
```

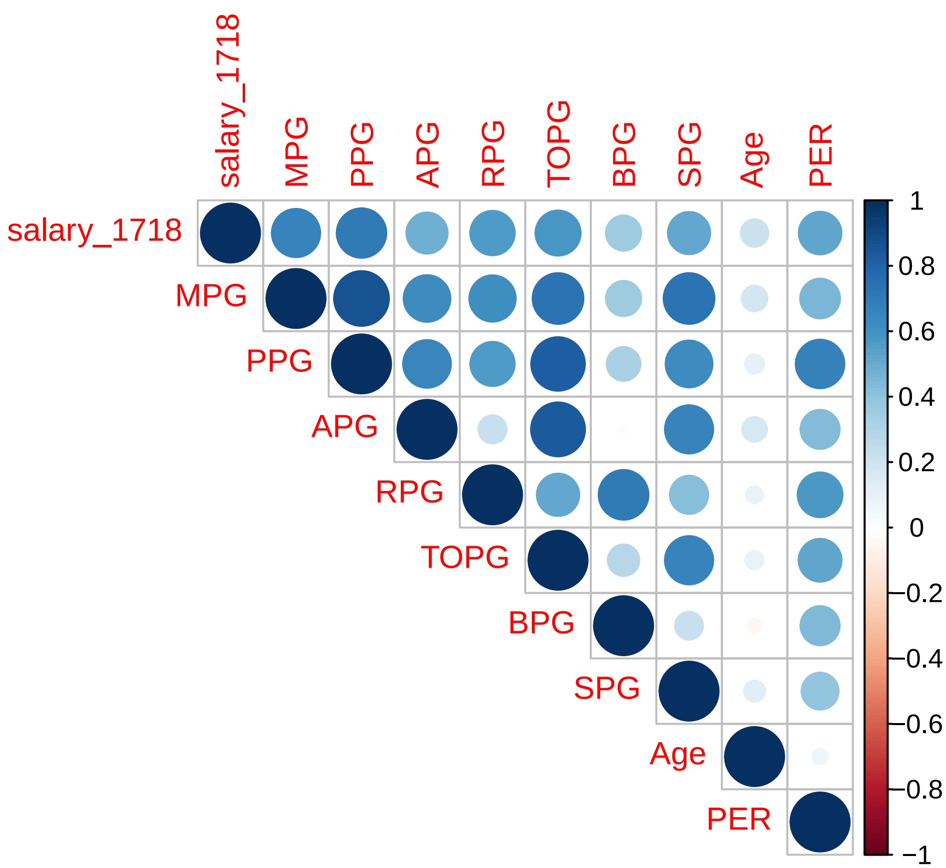
```
##            Player Year Pos Age Tm.x  G   MP  PER  FG FGA   FG. X3P X3PA  X3P.
## 1     A.J. Hammons 2017   C  24  DAL 22  163  8.4  17  42 0.405   5   10 0.500
## 2     Aaron Brooks 2017  PG  32  IND 65  894  9.5 121 300 0.403  48  128 0.375
## 3     Aaron Gordon 2017  SF  21  ORL 80 2298 14.4 393 865 0.454  77  267 0.288
## 4 Al-Farouq Aminu 2017  SF  26  POR 61 1773 11.3 183 466 0.393  70  212 0.330
## 5       Al Horford 2017   C  30  BOS 68 2193 17.7 379 801 0.473  86  242 0.355
## 6     Al Jefferson 2017   C  32  IND 66  931 18.9 235 471 0.499   0    1 0.000
##   X2P X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB TRB AST STL BLK TOV  PF  PTS
## 1  12   32 0.375 0.464   9  20 0.450   8  28  36   4   1  13  10  21   48
## 2  73  172 0.424 0.483  32  40 0.800  18  51  69 125  25   9  66  93  322
## 3 316  598 0.528 0.499 156 217 0.719 116 289 405 150  64  40  89 172 1019
## 4 113  254 0.445 0.468  96 136 0.706  77 374 451  99  60  44  94 102  532
## 5 293  559 0.524 0.527 108 135 0.800  95 369 464 337  52  87 116 138  952
## 6 235  470 0.500 0.499  65  85 0.765  75 203 278  57  19  16  33 125  535
##         MPG       PPG       APG      RPG      TOPG       BPG        SPG   X
```

```
## 1  7.409091  2.181818 0.1818182 1.636364 0.4545455 0.5909091 0.04545455 411
## 2 13.753846  4.953846 1.9230769 1.061538 1.0153846 0.1384615 0.38461538 319
## 3 28.725000 12.737500 1.8750000 5.062500 1.1125000 0.5000000 0.80000000 190
## 4 29.065574  8.721311 1.6229508 7.393443 1.5409836 0.7213115 0.98360656 154
## 5 32.250000 14.000000 4.9558824 6.823529 1.7058824 1.2794118 0.76470588  11
## 6 14.106061  8.106061 0.8636364 4.212121 0.5000000 0.2424242 0.28787879 128
##   salary_1718
## 1     1312611
## 2     2116955
## 3     5504420
## 4     7319035
## 5    27734405
## 6     9769821
```

2.2: Data exploration

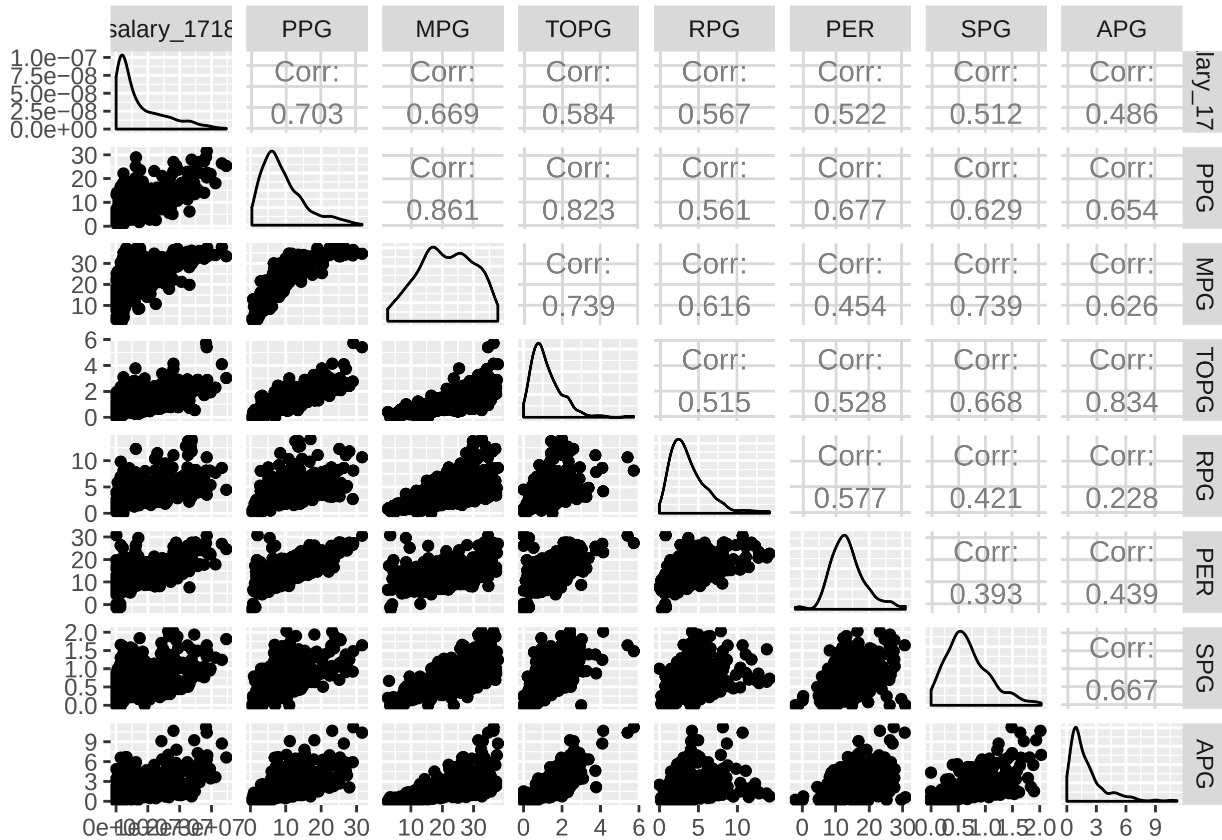
Lets take a look at the correlation between salary and the players per game stats:

```
#create correlation plot of players salary with their MPG, PPG, APG, RPG, TOPG, BPG, SPG, Age, and PER
corrplot(cor(stats1617_salary1718 %>%
               select(salary_1718, MPG:SPG,
                      Age, PER, contains("%")),
             use = "complete.obs"),
         method = "circle",type = "upper")
```



Lets look at another correlation technique to better visualize this data:

```r
stats_salary_cor <-
  stats1617_salary1718 %>%
  select(salary_1718, PPG, MPG, TOPG, RPG, PER, SPG, APG)
ggpairs(stats_salary_cor)
```



Lets focus on the top row:

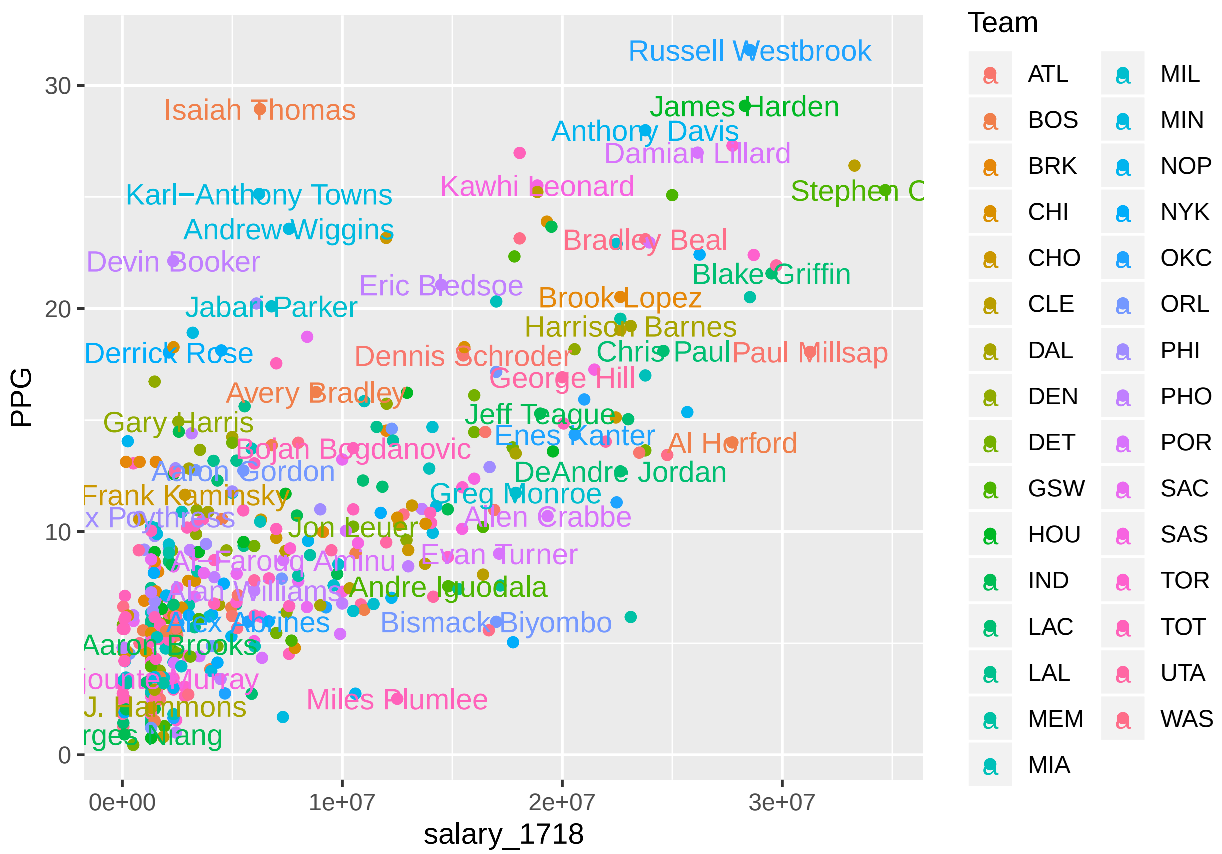
```r
cor(stats_salary_cor)[,"salary_1718"]
```

```
## salary_1718         PPG         MPG        TOPG         RPG         PER
##   1.0000000   0.7031051   0.6693910   0.5842982   0.5665350   0.5215509
##         SPG         APG
##   0.5118549   0.4856552
```

Note there is a strong correlation between minutes per game and salary. This makes sense because players playing the most time on the court usually are deserving of more pay. However, a surprising correlation is the turnovers per game with salary. No wonder Westbrook is getting paid (joking obviously).

Let us look at some plots. The first one we will look at is the salary against points per game:

```r
#name the team column Team
names(stats1617_salary1718)[5] <- "Team"
# #plot salary vs ppg with team as different groups
stats1617_salary1718 %>%
  ggplot(aes(x = salary_1718, y = PPG, color=Team, label=Player)) +
  geom_point() +geom_text(check_overlap = TRUE)
```



Looking back at this data it is crazy to see Isaiah Thomas put up incredible points given his low salary. If it was not for an injury in the post season and a Kyrie Irving trade I wonder how much money he would have made.

2.3: Model exploration: regression

Model 1: PPG

First of all, we would like to visualize if PPG represents a strong enough correlation to salary. Let us use a simple regression function:

```
#plot regression line
stats1617_salary1718 %>%
  ggplot(aes(x = salary_1718, y = PPG)) +
  geom_point() +
  geom_smooth(method = "lm")
```

The line does not look like a great fit. Let us dive deeper into our analysis.

Model 2: using per game values for regression

We will use MPG, PPG, APG, RPG, TOPG, BPG, SPG for our regression analysis with the help of the 'lm' function:

```
#create regression variable
stats_salary_regression <-
  stats1617_salary1718 %>% select(salary_1718, MPG:SPG)
#run regression on dataset
lm(salary_1718~., data=stats_salary_regression)
```

```
##
## Call:
## lm(formula = salary_1718 ~ ., data = stats_salary_regression)
##
## Coefficients:
## (Intercept)          MPG          PPG          APG          RPG         TOPG
##    -2792909        30565       686815      1059087       916087     -2709447
##         BPG          SPG
##      470136       631255
```

Wow! It seems from this model analysis that as APG increases by a unit they will be predicted to make an additional 1,059,087 USD per year. Interesting that TOPG takes a huge hit of -2,709,447 USD per year.

Model 3 Experiment: turnover and playing time salary analysis

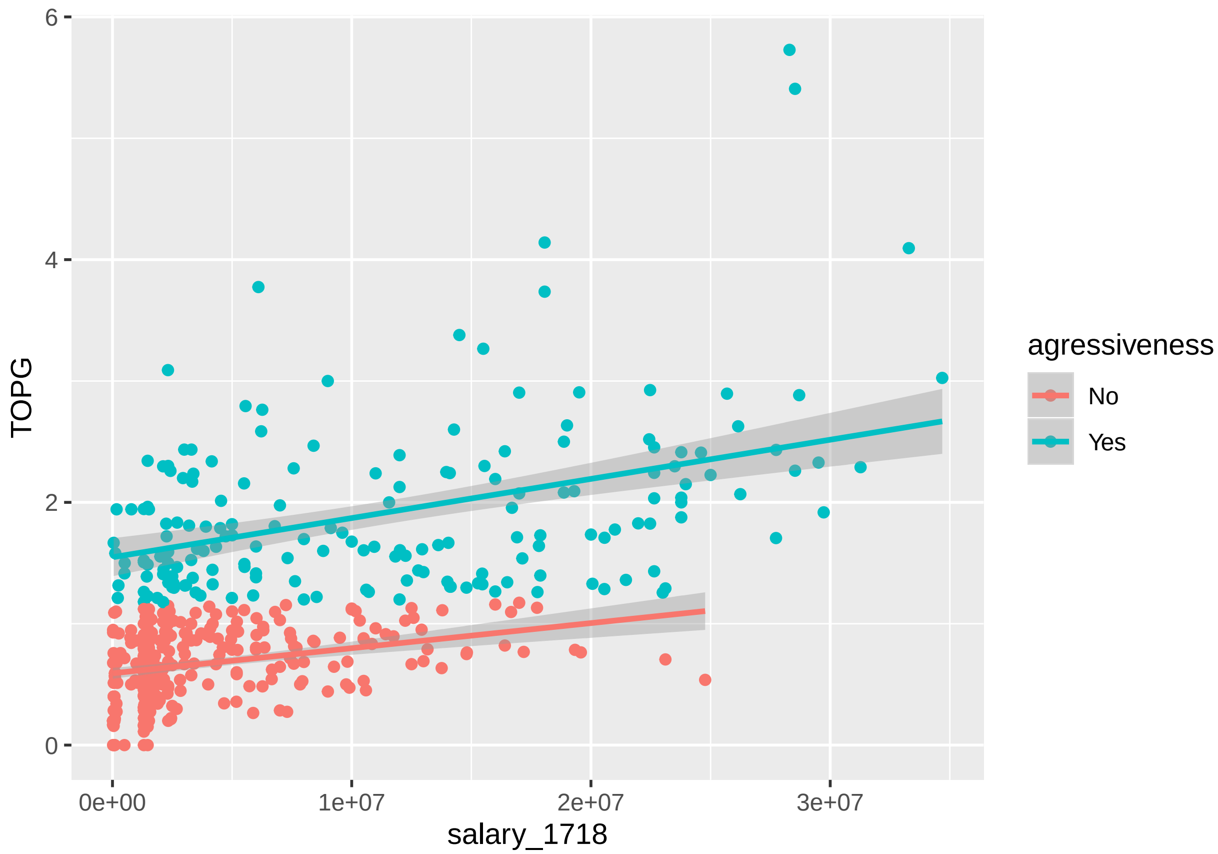
Let us revisit the turnover subject. Recall we saw a high turnover rate lead to a positive correlation with salary. Let us see if players average higher salaries if they have a high turnover rates:

```
#find the average MPG and TOPG
avg.minutes <- mean(stats_salary_regression$MPG)
avg.turnover <- mean(stats_salary_regression$TOPG)

#create a trusted column if players get above average minutes
stats1617_salary1718$trusted <- as.factor(ifelse(stats1617_salary1718$MPG >= avg.minutes, "Yes", "No"))

#create a aggressiveness column if players turn over the ball more then the average
stats1617_salary1718$agressiveness <- as.factor(ifelse(stats1617_salary1718$TOPG >= avg.turnover, "Yes"

#lets look at our new dataset
head(stats1617_salary1718)
```

```
##            Player Year Pos Age Team  G   MP  PER  FG FGA   FG. X3P X3PA  X3P.
## 1    A.J. Hammons 2017   C  24  DAL 22  163  8.4  17  42 0.405   5   10 0.500
## 2    Aaron Brooks 2017  PG  32  IND 65  894  9.5 121 300 0.403  48  128 0.375
## 3    Aaron Gordon 2017  SF  21  ORL 80 2298 14.4 393 865 0.454  77  267 0.288
## 4 Al-Farouq Aminu 2017  SF  26  POR 61 1773 11.3 183 466 0.393  70  212 0.330
## 5      Al Horford 2017   C  30  BOS 68 2193 17.7 379 801 0.473  86  242 0.355
## 6    Al Jefferson 2017   C  32  IND 66  931 18.9 235 471 0.499   0    1 0.000
##   X2P X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB TRB AST STL BLK TOV  PF  PTS
## 1  12   32 0.375 0.464   9  20 0.450   8  28  36   4   1  13  10  21   48
## 2  73  172 0.424 0.483  32  40 0.800  18  51  69 125  25   9  66  93  322
## 3 316  598 0.528 0.499 156 217 0.719 116 289 405 150  64  40  89 172 1019
## 4 113  254 0.445 0.468  96 136 0.706  77 374 451  99  60  44  94 102  532
## 5 293  559 0.524 0.527 108 135 0.800  95 369 464 337  52  87 116 138  952
## 6 235  470 0.500 0.499  65  85 0.765  75 203 278  57  19  16  33 125  535
##         MPG       PPG       APG      RPG      TOPG       BPG        SPG   X
## 1  7.409091  2.181818 0.1818182 1.636364 0.4545455 0.5909091 0.04545455 411
## 2 13.753846  4.953846 1.9230769 1.061538 1.0153846 0.1384615 0.38461538 319
## 3 28.725000 12.737500 1.8750000 5.062500 1.1125000 0.5000000 0.80000000 190
## 4 29.065574  8.721311 1.6229508 7.393443 1.5409836 0.7213115 0.98360656 154
## 5 32.250000 14.000000 4.9558824 6.823529 1.7058824 1.2794118 0.76470588  11
## 6 14.106061  8.106061 0.8636364 4.212121 0.5000000 0.2424242 0.28787879 128
##   salary_1718 trusted agressiveness
## 1     1312611      No            No
## 2     2116955      No            No
## 3     5504420     Yes            No
## 4     7319035     Yes           Yes
## 5    27734405     Yes           Yes
## 6     9769821      No            No
```

Let us plot two separate regression lines: (1) for players who aren't considered agressive (they do not turn over the ball frequently) and (2) for players who are agressive (they have a high turnover rate):

```
stats1617_salary1718 %>%
  ggplot(aes(x = salary_1718, y = TOPG, colour = agressiveness)) +
  geom_point() +
  geom_smooth(method="lm")
```



Looks like players who play agressive (noted by a high turnover rate) tend to have higher salaries.

Lastly, let us make a regression line looking at if the coach trusts a player (noted by above league average playing time), and if the player is agressive (noted by an above league average turnover rate):

```
lm(formula = salary_1718 ~ trusted * agressiveness, data=stats1617_salary1718)
```

```
##
## Call:
## lm(formula = salary_1718 ~ trusted * agressiveness, data = stats1617_salary1718)
##
## Coefficients:
##                (Intercept)                       trustedYes
##                    2914582                          5125780
##           agressivenessYes   trustedYes:agressivenessYes
##                     969783                          3518647
```

Interesting. As we can see if a player is trusted (have a high playing time) they are predicted to make more salary than a player who plays agressively (and turns over the ball often). We will shortly see that a modle with two yes/no type parameters is a poor and limited way to predict salary. It is better to have continuous variables in this case then discrete yes or no features.

SECTION 3: Results

The 3 models discussed above will be ran against NBA player and fan favorite; Pascal Siakam. Pascal was just extended a max contract (29 million for the 2020-2021 season) with the Raptors based on his play in the

2018-2019 season (and his age). With Leonard gone the Raptors believe Pascal can be the franchise player. Let us see if he is living up to his contract extension in this 2019-2020 season. Thus far he is averaging: 36.9 MPG, 8.6 RPG, 3.8 APG, 0.9 SPG, 0.7 BPG, 2.9 TOPG and 25.0 PPG.

We will define a salary_prediction function for each model which takes in the required parameters.

Please note, the models are created based on the stats in the 2016-2017 season. Given more time it would have been useful to extrapolate the latest season stats. Also, factors such as age and player efficency should have been added to our model.

3.1 Model 1:

Recall, this model only considers points per game

```
#considers points per game in function
salary_prediction_model1 <- function(m, points){
  pre_new <- predict(m, data.frame(PPG = points))
  msg <- paste("PPG:", points, " ==> Expected Salary: $", format(round(pre_new), big.mark = ","), sep =
  print(msg)
}
#create model
model1<- lm(salary_1718~PPG, data=stats1617_salary1718)
#predict salary
predict1<-salary_prediction_model1(model1, 25.0)
```

```
## [1] "PPG:25 ==> Expected Salary: $21,139,420"
```

Interesting, given Pascal's PPG alone he is predicted to command a salary of approximately 21 million.

3.2 Model 2:

This model considers MPG, PPG, APG, RPG, TOPG, BPG, SPG

```
#considers points per game in function
salary_prediction_model2 <- function(m, minutes, points, assists, rebounds, turnovers, blocks, steals){
  pre_new <- predict(m, data.frame(PPG = points, MPG=minutes, APG=assists,
                                   RPG=rebounds, TOPG=turnovers, BPG=blocks,
                                   SPG=steals))
  msg <- paste("PPG:", points, "RPG:", rebounds, "MPG:", minutes, "APG:", assists, "TOPG:", turnovers,
  print(msg)
}
#create model
model2<- lm(salary_1718~., data=stats_salary_regression)
#predict salary
predict2<-salary_prediction_model2(model2, 36.9, 25.0, 3.8, 8.6, 2.9, 0.7, 0.9)
```

```
## [1] "PPG:25RPG:8.6MPG:36.9APG:3.8TOPG:2.9BPG:0.7SPG:0.9 ==> Expected Salary: $20,448,021"
```

Interesting Pascals estimated salary droped by the smallest margin (1 million in NBA contract is not a huge hit for a team) given his current play and per game stats. It is estimated in this model to be worth approximately 20 million.

3.3 Model 3:

In this model let us consider trusted and aggressive properties of players. Pascal is playing above league average minutes and is turning the ball over more than the league average in the 2016-2017 season. For the sake of this project let us consider that this is trusted and aggressive play:

```
#considers points per game in function
salary_prediction_model3 <- function(m, trusted, agressiveness){
  pre_new <- predict(m, data.frame(trusted = trusted, agressiveness = agressiveness))
  msg <- paste("Trusted:", trusted, "Agressive:", agressiveness, " ==> Expected Salary: $", format(round
  print(msg)
}
model3<-lm(formula = salary_1718 ~ trusted * agressiveness, data=stats1617_salary1718)
predict3<-salary_prediction_model3(model3, "Yes", "Yes")
```

```
## [1] "Trusted:YesAgressive:Yes ==> Expected Salary: $12,528,793"
```

Reasonably so, having just two true or false parameters limits how much we are able to predict the salary of a player. Pascal's salary is way off what is expected by just looking at two yes or no questions. It is estimated here to be approximately 12.5 million dollars.

SECTION 4: Conclusion

From this experiement we looked at different regression models and how they can predict a player salary. It can be seen that predicting salary off of two yes or no questions (i.e. are they above league average in turnovers/minutes) is not a good way of predicting. That said, simply looking at PPG misses on a lot of valuable information as well.

I believe the best metric for predicting salary is one not explored in this project. That is, offensive and defensive metrics should be considered as well as age. Also, another factor should be how they contribute to the culture and winning ways of a team. Not to mention, this talk complicates a lot more if we try analyzing their play in the playoffs.

This is such an exciting topic to look at, and given more time I would love to explore some of the above. Thank you for reading this and lets go Raptors! :)

SECTION 5: References

This project is credited to the project made on kaggle by Koki Ando at: https://www.kaggle.com/koki25ando/nba-salary-prediction-using-multiple-regression/report and the data science team at HarvardX for teaching the regression material.

SECTION 6: Appendix

This interactive plot would not export to .pdf and thus the code is here if you would like to see salary vs. points per game with the teams as the color differentiator:

Using Koki's code on Kaggle as reference, let us look at an interactive plot. We will look at salary against points per game:

```
# #name the team column Team
# names(stats1617_salary1718)[5] <- "Team"
# #plot salary vs ppg with team as different groups
# plot_ly(data = stats1617_salary1718, x = ~salary_1718, y = ~PPG, color = ~Team,
#         hoverinfo = "text",
#         text = ~paste("Player: ", Player,
#                       "<br>Salary: ", format(salary_1718, big.mark = ","),"$",
#                       "<br>PPG: ", round(PPG, digits = 3),
#                       "<br>Team: ", Team)) %>%
#   layout(
#     title = "Salary vs Point Per Game",
#     xaxis = list(title = "Salary (USD)"),
#     yaxis = list(title = "Point per Game (PPG)")
#   )
```