Predicting NBA Salaries using Multiple Regression

Stephen Zhong

June 15, 2021

Contents

Introduction

In the NBA, when constructing a team, the only thing better than getting talent is getting talent at a good price. With the salary cap changing from year to years, front offices are tasked with building a competitive roster while managing the team payroll. Having spent the past semester learning the fundamentals of R and data analysis, in this project, I plan on using multiple regression to build a model to help predict the estimated value of a player in terms of salary based on his statistics.

Preliminaries

Style Edits

```
knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
knitr::opts_chunk$set(echo = TRUE)
```

Necessary Packages

```
library("knitr")
library("ggplot2")
library("kableExtra")
library("readr")
library("car")
library("readxl")
library("grid")
library("gridExtra")
library(data.table)
library(tidyverse)
```

Importing and Cleaning Data

The "nbaStats" dataset used data taken from the Basketball Reference website. Contained in the data was their statistics (points, rebounds, etc.), and I added information on their salary and contract. For the project, I decided to look at percent of salary cap, as the NBA salary cap is constantly evolving. The 2020-21 NBA season has the salary cap set at $109.1 million, with next year's being 112.4 million dollars.

```
nbaStats <- read_excel("nbaStats.xlsx")
str(nbaStats)
```

```
## tibble [540 x 32] (S3: tbl_df/tbl/data.frame)
##  $ Player      : chr [1:540] "Precious Achiuwa" "Jaylen Adams" "Steven Adams" "Bam Adebayo" ...
##  $ Pos         : chr [1:540] "PF" "PG" "C" "C" ...
##  $ Age         : num [1:540] 21 24 27 23 35 22 22 25 22 30 ...
##  $ Tm          : chr [1:540] "MIA" "MIL" "NOP" "MIA" ...
##  $ G           : num [1:540] 61 7 58 64 26 15 46 50 63 23 ...
##  $ GS          : num [1:540] 4 0 58 64 23 0 13 38 45 14 ...
##  $ MP          : num [1:540] 12.1 2.6 27.7 33.5 25.9 3.1 21.9 25.2 29.6 18.9 ...
##  $ FGM         : num [1:540] 2 0.1 3.3 7.1 5.4 0.2 4.2 3.5 4.7 1.7 ...
##  $ FGA         : num [1:540] 3.7 1.1 5.3 12.5 11.4 0.8 10 8.3 7.7 4.3 ...
##  $ FG          : num [1:540] 0.544 0.125 0.614 0.57 0.473 0.25 0.419 0.418 0.618 0.384 ...
##  $ 3PM         : num [1:540] 0 0 0 0 1.2 0.1 1.7 2.1 0.1 0.3 ...
##  $ 3PA         : num [1:540] 0 0.3 0.1 0.1 3.1 0.6 4.8 5.5 0.3 1.6 ...
##  $ 3P          : num [1:540] 0 0 0 0.25 0.388 0.222 0.347 0.391 0.316 0.216 ...
```

```
##  $ 2PM         : num [1:540] 2 0.1 3.3 7.1 4.2 0.1 2.5 1.3 4.6 1.3 ...
##  $ 2PA         : num [1:540] 3.7 0.9 5.3 12.4 8.3 0.2 5.2 2.8 7.3 2.7 ...
##  $ 2P          : num [1:540] 0.546 0.167 0.62 0.573 0.505 0.333 0.485 0.471 0.631 0.484 ...
##  $ eFG         : num [1:540] 0.544 0.125 0.614 0.571 0.525 0.333 0.502 0.547 0.624 0.424 ...
##  $ FTM         : num [1:540] 0.9 0 1 4.4 1.6 0.1 1 1.6 3.2 0.8 ...
##  $ FTA         : num [1:540] 1.8 0 2.3 5.5 1.8 0.1 1.4 1.8 4.6 1 ...
##  $ FT          : num [1:540] 0.509 NA 0.444 0.799 0.872 0.5 0.727 0.868 0.703 0.818 ...
##  $ ORB         : num [1:540] 1.2 0 3.7 2.2 0.7 0.1 0.3 0.4 3.1 1 ...
##  $ DRB         : num [1:540] 2.2 0.4 5.2 6.7 3.8 0.5 2.8 2.8 6.9 3.8 ...
##  $ TRB         : num [1:540] 3.4 0.4 8.9 9 4.5 0.7 3.1 3.2 10 4.8 ...
##  $ AST         : num [1:540] 0.5 0.3 1.9 5.4 1.9 0.4 2.2 2.2 1.7 1.3 ...
##  $ STL         : num [1:540] 0.3 0 0.9 1.2 0.4 0 1 0.9 0.5 0.8 ...
##  $ BLK         : num [1:540] 0.5 0 0.7 1 1.1 0.1 0.5 0.2 1.4 0.4 ...
##  $ TOV         : num [1:540] 0.7 0 1.3 2.6 1 0.2 1.5 1 1.6 1.2 ...
##  $ PF          : num [1:540] 1.5 0.1 1.9 2.3 1.8 0.1 1.9 1.4 1.5 1.3 ...
##  $ PTS         : chr [1:540] "5" "0.3" "7.6" "18.7" ...
##  $ Salary      : num [1:540] 2582160 NA 27528090 5115492 19078340 ...
##  $ PercentOfCap: num [1:540] 0.0237 NA 0.2523 0.0469 0.1749 ...
##  $ ContractType: chr [1:540] "1st Round Pick" NA "1st Round Pick" "1st Round Pick" ...
```

For cleaning the data, I wanted to limit the statistics only to players who have played more than 18 games (1/4 of the 20-21 season) so their sample size is not too small, and also eliminated certain statistics I knew were redundant or useless. For example, with 2 point percentage as a statistic, there is no need for 2 points made and 2 points attempted. I also set any numeric stats that are blank to 0, and any type of contract that doesn't fit into one of the given categories as "Other".

```
nbaStatsCleaned <- subset(nbaStats,
                          G >= 15,
                          select = -c(FGM, `2PM`, `3PM`, FTM, `3PA`, `2PA`, ORB, DRB, Tm, Salary, G, FG

nbaStatsCleaned$PTS <- as.numeric(nbaStatsCleaned$PTS)
nbaStatsCleaned$PTS <- round(nbaStatsCleaned$PTS, digits = 1)

nbaStatsCleaned[c(4:18)][is.na(nbaStatsCleaned[c(4:18)])] <- 0
nbaStatsCleaned[c(19)][is.na(nbaStatsCleaned[c(19)])] <- "Other"
```

Exploratory Data Analysis

For our dataset, we will be looking at all NBA players in the 2020-21 NBA season who played in more than 15 games. The different potential predictors of salary are listed below:

Pos: position played
Age: years of age (as of June 2021)
GS: number of games started
MP: minutes played per game
FG: field goal percentage
3P: three point field goal percentage
2P: two point field goal percentage
eFG: effective field goal percentage (accounts for difference in points of 3P% and 2P%)
FT: free throw percentage
TRB: total rebounds per game (offensive and defensive)
AST: assists per game
STL: steals per game
BLK: blocks per game
TOV: turnovers per game
PTS: points per game

Our response variable is also listed below:

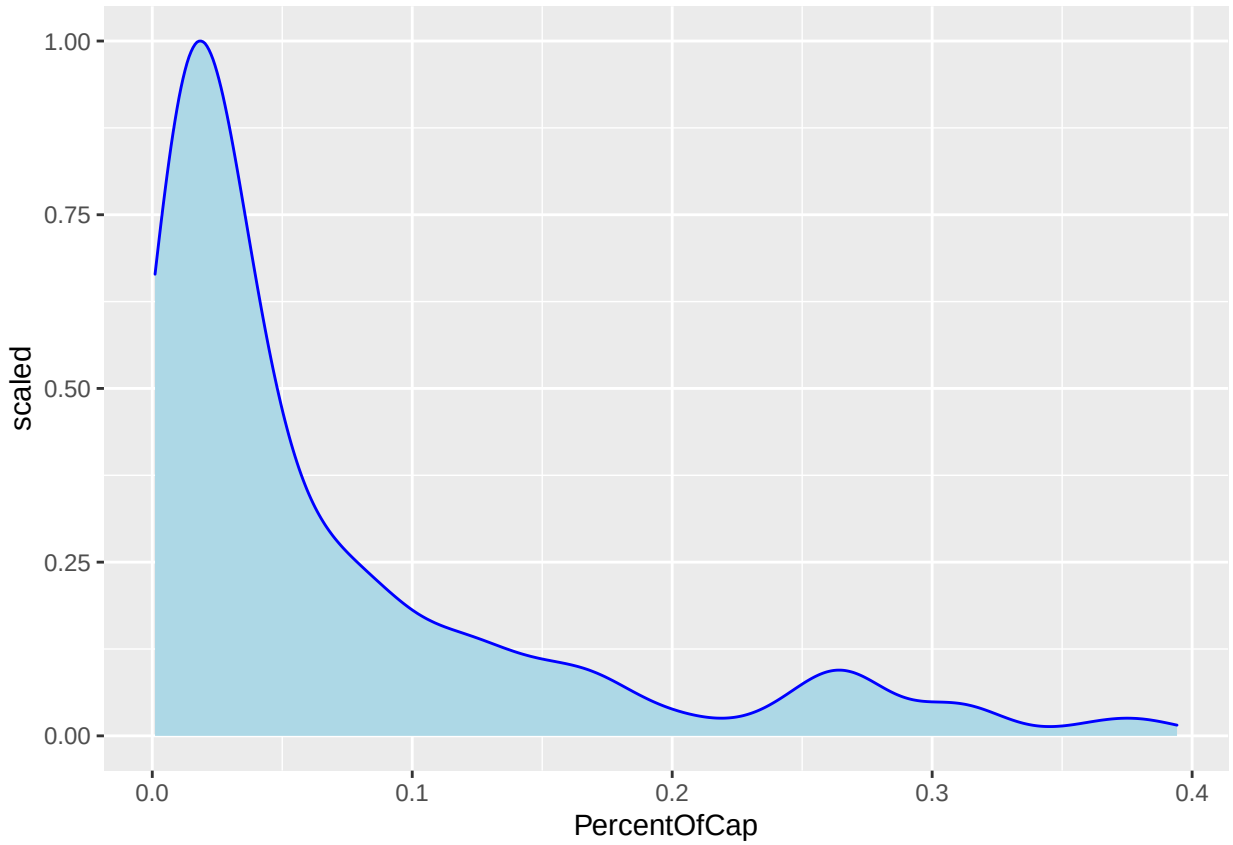
PercentOfCap: percent of salary cap of player's 2020-21 season salary

Univariate Analysis

Response Variable

First, let's took a look at our response variable, which is the percent of cap space.

```
ggplot(data = nbaStatsCleaned, aes(PercentOfCap)) + geom_density(aes(y = ..scaled..), color = "blue", fi
```




```
summary(nbaStatsCleaned$PercentOfCap)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00107 0.01539 0.03324 0.07142 0.09179 0.39419
```

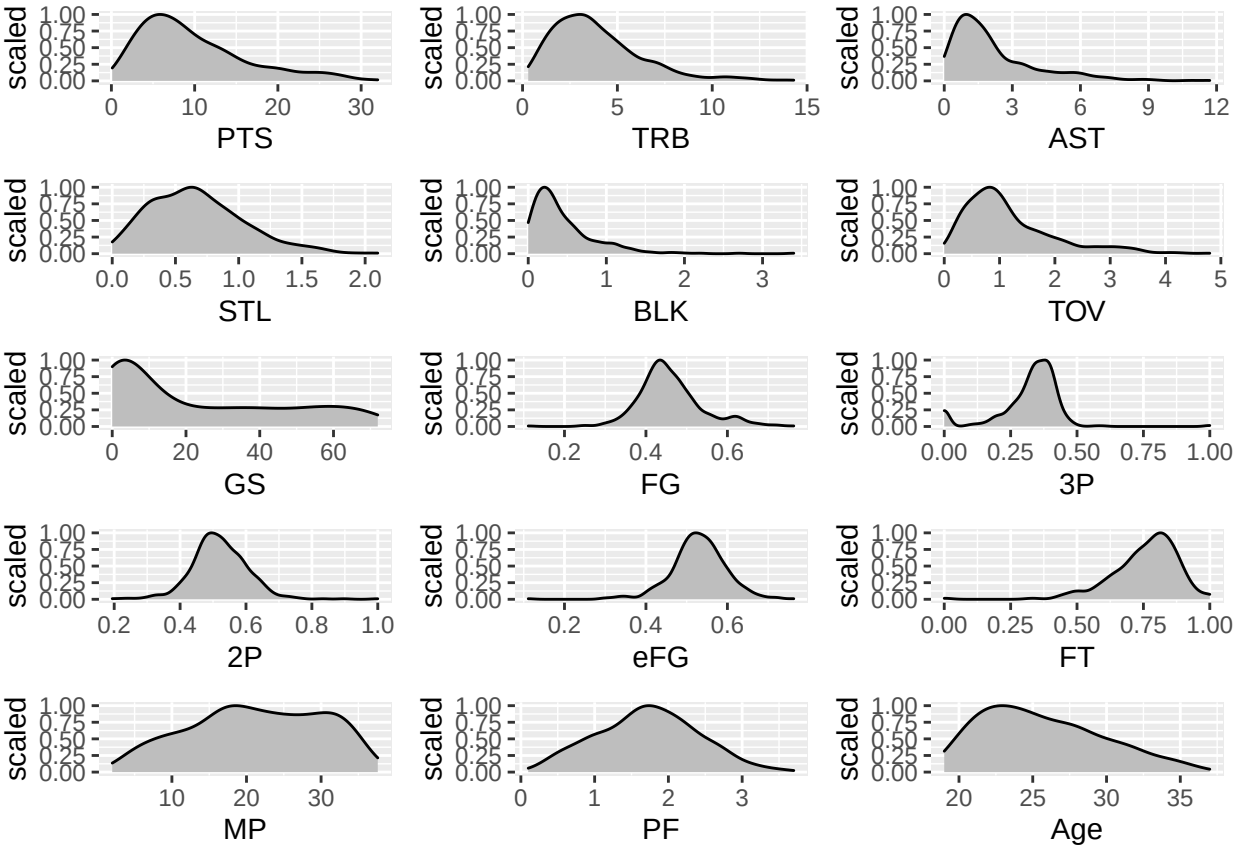
As we can see based on the density plot, the contracts are heavily skewed to the right, with most players having smaller sized contracts, and only a select few having contracts worth a heavy portion of the cap.

Predictor Variables

Next, let's take a look at all the potential predictor variables.

```
pts <- ggplot(data = nbaStatsCleaned, aes(PTS)) + geom_density(aes(y = ..scaled..), color = "black", fi
trb <- ggplot(data = nbaStatsCleaned, aes(TRB)) + geom_density(aes(y = ..scaled..), color = "black", fi
ast <- ggplot(data = nbaStatsCleaned, aes(AST)) + geom_density(aes(y = ..scaled..), color = "black", fi
stl <- ggplot(data = nbaStatsCleaned, aes(STL)) + geom_density(aes(y = ..scaled..), color = "black", fi
blk <- ggplot(data = nbaStatsCleaned, aes(BLK)) + geom_density(aes(y = ..scaled..), color = "black", fi
tov <- ggplot(data = nbaStatsCleaned, aes(TOV)) + geom_density(aes(y = ..scaled..), color = "black", fi
gs <- ggplot(data = nbaStatsCleaned, aes(GS)) + geom_density(aes(y = ..scaled..), color = "black", fill
fg <- ggplot(data = nbaStatsCleaned, aes(`FG`)) + geom_density(aes(y = ..scaled..), color = "black", fi
`3p` <- ggplot(data = nbaStatsCleaned, aes(`3P`)) + geom_density(aes(y = ..scaled..), color = "black",
`2p` <- ggplot(data = nbaStatsCleaned, aes(`2P`)) + geom_density(aes(y = ..scaled..), color = "black",
efg <- ggplot(data = nbaStatsCleaned, aes(`eFG`)) + geom_density(aes(y = ..scaled..), color = "black",
ft <- ggplot(data = nbaStatsCleaned, aes(`FT`)) + geom_density(aes(y = ..scaled..), color = "black", fi
mp <- ggplot(data = nbaStatsCleaned, aes(MP)) + geom_density(aes(y = ..scaled..), color = "black", fill
pf <- ggplot(data = nbaStatsCleaned, aes(PF)) + geom_density(aes(y = ..scaled..), color = "black", fill
age <- ggplot(data = nbaStatsCleaned, aes(Age)) + geom_density(aes(y = ..scaled..), color = "black", fi
```

```
grid.arrange(pts, trb, ast, stl, blk, tov, gs, fg, `3p`, `2p`, efg, ft, mp, pf, age, ncol = 3)
```



There's a lot to note with all the different potential predictor variables in our multiple regression model. First and foremost, all distributions are unimodal. As we can see, with a lot of the counting statistics or traditional box score metrics, such as points, rebounds, assists, etc., those are all skewed heavily to the right. This makes sense, as most players are going to be pretty low in counting statistics, while there will be few elite players in each statistical category who have higher numbers.

In comparison, when looking at percentage based statistics, such as field goal percentage and free throw percentage, these graphs follow a lot more of a normal distribution, with most players hovering around the league average, while some are elite, and others are really bad. You'll notice with 3P% that there is a little spike at 0%, but this is because anyone who did not shoot a three pointer on the season was given 0% in three point percentage.

Bivariate Analysis

```
cor(nbaStatsCleaned[c(3:17)], nbaStatsCleaned$PercentOfCap)
```

```
##           [,1]
## Age 0.42270356
## GS  0.56992926
## MP  0.62774563
## FG  0.12197383
## 3P  0.12006708
## 2P  0.04038261
## eFG 0.13597144
## FT  0.22835307
```

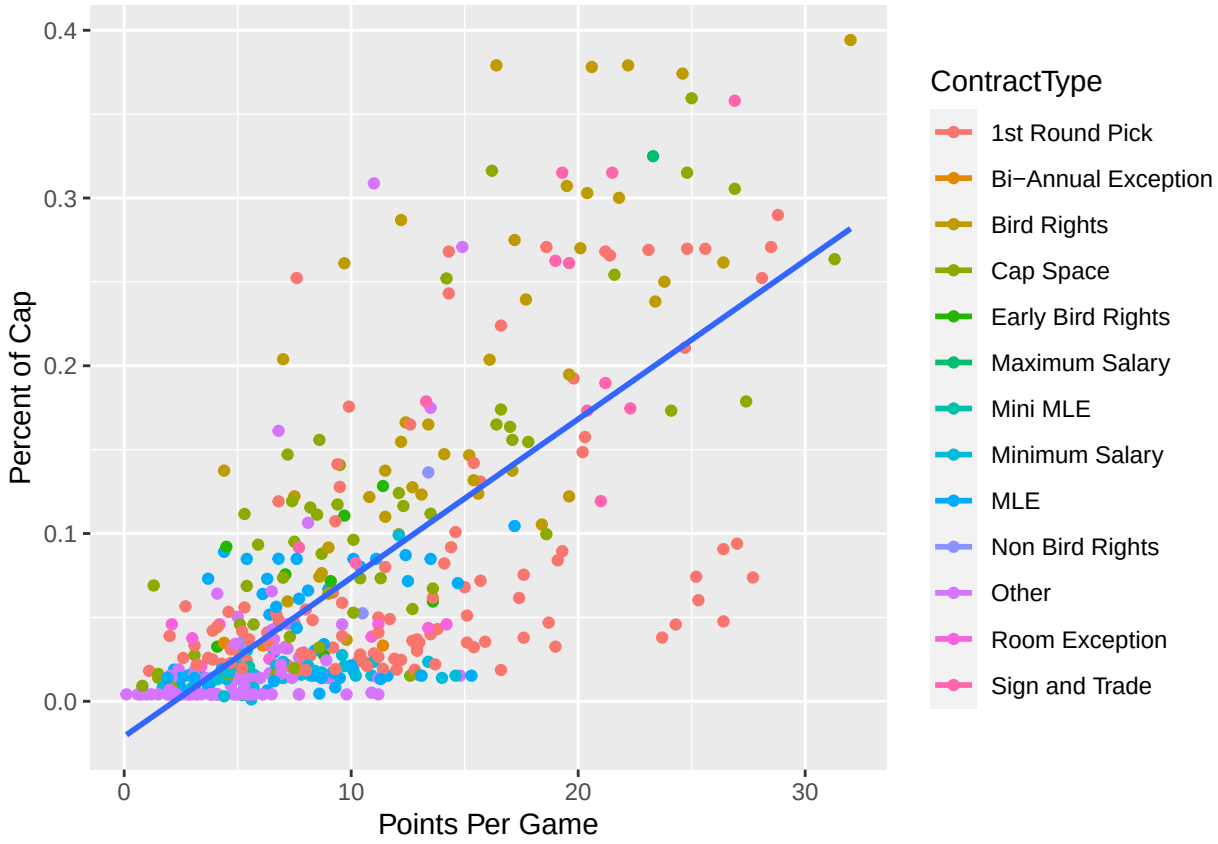
```
## TRB 0.49240479
## AST 0.64402603
## STL 0.52817862
## BLK 0.23477819
## TOV 0.65315309
## PF  0.37807924
## PTS 0.71351162
```

As we can see, the order of statistics that have the strongest correlation with PercentOfCap are as follows:

PTS > FGA > TOV > AST > FTA > MP > STL > TRB > Age > PF > BLK > FT > 3P > 2P

For the sake of this part, let's take a look at how points per game correlates with percent of cap.

```
ggplot(data = nbaStatsCleaned, aes(x = PTS, y = PercentOfCap, colour = ContractType)) + geom_point() + l
```



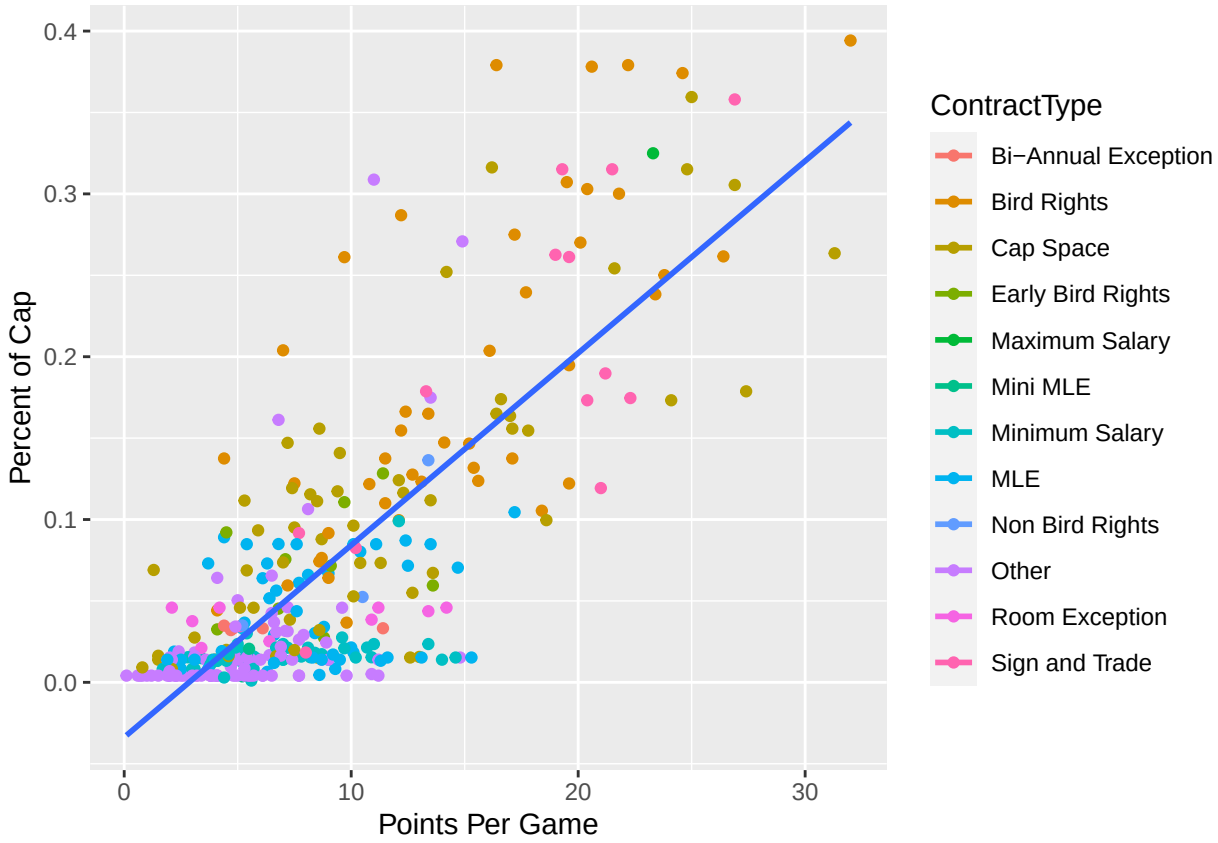

```
cor(nbaStatsCleaned$PTS, nbaStatsCleaned$PercentOfCap)
```

```
## [1] 0.7135116
```

Interestingly enough, we see that below the regression line, there are a number of players who all have the same contract type, which is 1st round pick. This can be explained as first round picks are signed to rookie deals, and won't have a salary that reflects their actual performance until they are in the league for up to 4 years. In order to account for this, I'll also have a separate dataset called "nonRookieContracts" that looks at all players in the league that are no longer on rookie contracts.

```
nonRookieContracts <- subset(nbaStatsCleaned,
                             ContractType != "1st Round Pick")
```

```
ggplot(data = nonRookieContracts, aes(x = PTS, y = PercentOfCap, color = ContractType)) + geom_point() +
```




```
cor(nonRookieContracts$PTS, nonRookieContracts$PercentOfCap)
```

```
## [1] 0.7971384
```

As we can see, once we remove players still on their rookie contracts, the correlation increases from around .71 to .79. For the rest of this project, we'll focus more on this dataset without rookie contracts instead.

Modeling

First, let's build an initial model for both the nonRookieContracts dataset using all the predictor variables. We will assume our model will be reasonably linear.

```
nonRookieModel <- lm(PercentOfCap ~ Age + MP + GS + eFG + FG + `3P` + `2P` + FT + TRB + AST + STL + BLK

summary(nonRookieModel)
```

```
##
## Call:
## lm(formula = PercentOfCap ~ Age + MP + GS + eFG + FG + `3P` +
##     `2P` + FT + TRB + AST + STL + BLK + TOV + PF + PTS, data = nonRookieContracts)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.135338 -0.023394 -0.002268  0.020449  0.202528
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0937227  0.0261538  -3.584 0.000391 ***
## Age          0.0054162  0.0006761   8.011 2.10e-14 ***
## MP          -0.0044002  0.0009172  -4.797 2.46e-06 ***
## GS           0.0006404  0.0001811   3.535 0.000467 ***
## eFG          0.0870379  0.0853091   1.020 0.308368
## FG          -0.1859598  0.0801088  -2.321 0.020892 *
## `3P`        -0.0290376  0.0254010  -1.143 0.253819
## `2P`         0.0048166  0.0351349   0.137 0.891046
## FT           0.0144573  0.0214895   0.673 0.501580
## TRB          0.0067487  0.0018886   3.573 0.000406 ***
## AST          0.0147655  0.0031376   4.706 3.76e-06 ***
## STL          0.0109196  0.0113929   0.958 0.338548
## BLK          0.0201471  0.0096180   2.095 0.036977 *
## TOV          0.0057061  0.0089991   0.634 0.526483
## PF          -0.0210617  0.0061386  -3.431 0.000680 ***
## PTS          0.0099559  0.0010632   9.364  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04261 on 322 degrees of freedom
## Multiple R-squared:  0.7816, Adjusted R-squared:  0.7714
## F-statistic: 76.83 on 15 and 322 DF,  p-value: < 2.2e-16
```

As we can see, our multiple R-squared came out to .7816, or 78.16% percent of the variation in percent of cap can be explained by the predictor variables. However, let's also take a look at potential multicollinearity between predictor variables.

```
round(cor(nonRookieContracts[c(3:17)]), digits = 2)
```

```
##      Age   GS   MP    FG    3P    2P  eFG    FT   TRB   AST   STL   BLK  TOV
## Age 1.00 0.28 0.42  0.09  0.14  0.06 0.22  0.25  0.26  0.35  0.31  0.14 0.30
## GS  0.28 1.00 0.81  0.18  0.11  0.10 0.24  0.18  0.60  0.56  0.60  0.36 0.62
## MP  0.42 0.81 1.00  0.18  0.22  0.07 0.29  0.31  0.64  0.70  0.77  0.37 0.75
## FG  0.09 0.18 0.18  1.00 -0.15  0.65 0.83 -0.08  0.49  0.03  0.10  0.52 0.15
## 3P  0.14 0.11 0.22 -0.15  1.00 -0.20 0.21  0.42 -0.15  0.21  0.14 -0.27 0.12
## 2P  0.06 0.10 0.07  0.65 -0.20  1.00 0.55 -0.18  0.28 -0.06 -0.01  0.34 0.01
```

```
## eFG 0.22 0.24 0.29  0.83  0.21  0.55 1.00  0.16  0.35  0.05  0.14  0.36 0.11
## FT  0.25 0.18 0.31 -0.08  0.42 -0.18 0.16  1.00 -0.03  0.27  0.26 -0.19 0.18
## TRB 0.26 0.60 0.64  0.49 -0.15  0.28 0.35 -0.03  1.00  0.38  0.45  0.67 0.56
## AST 0.35 0.56 0.70  0.03  0.21 -0.06 0.05  0.27  0.38  1.00  0.72  0.07 0.86
## STL 0.31 0.60 0.77  0.10  0.14 -0.01 0.14  0.26  0.45  0.72  1.00  0.26 0.67
## BLK 0.14 0.36 0.37  0.52 -0.27  0.34 0.36 -0.19  0.67  0.07  0.26  1.00 0.25
## TOV 0.30 0.62 0.75  0.15  0.12  0.01 0.11  0.18  0.56  0.86  0.67  0.25 1.00
## PF  0.30 0.61 0.70  0.37  0.00  0.19 0.34  0.08  0.70  0.38  0.54  0.58 0.55
## PTS 0.31 0.70 0.87  0.19  0.24  0.07 0.26  0.32  0.58  0.71  0.64  0.29 0.83
##       PF  PTS
## Age 0.30 0.31
## GS  0.61 0.70
## MP  0.70 0.87
## FG  0.37 0.19
## 3P  0.00 0.24
## 2P  0.19 0.07
## eFG 0.34 0.26
## FT  0.08 0.32
## TRB 0.70 0.58
## AST 0.38 0.71
## STL 0.54 0.64
## BLK 0.58 0.29
## TOV 0.55 0.83
## PF  1.00 0.53
## PTS 0.53 1.00
```

Once we run the correlation table, we can see that between certain predictor variables, there are some rather high correlations. This means we are likely going to be dealing with multicollinearity. In order to better understand this, let's took a look at the vif values, or variation inflation factors of this mulitlinear regression model.

```
vif(nonRookieModel)
```

```
##       Age        MP        GS       eFG        FG      `3P`      `2P`        FT
##  1.359316 11.642774  3.039244  7.011597  7.832631  1.945429  1.869182  1.553072
##       TRB       AST       STL       BLK       TOV        PF       PTS
##  3.520635  6.126715  3.223357  2.472569  8.163627  3.334407  7.607475
```

As we can see, there are a few variables with high vifs, most notably MP, eFG%, FG%, AST, TOV, and PTS. However, a lot of these make sense. For FG% and eFG%, they measure the same thing (shooting efficiency), but with slightly different calculations. As a result, we'll likely remove all shooting efficiency metrics except eFG%, as it takes into account 3P% and 2P%, and FT%.

We also see that MP has a pretty high vif value. This also makes sense, as more minutes played means that there will be an increase in your other counting statistics, such as rebounds, assists, and points. As a result, we don't really need MP as a variable.

Finally, assists and turnovers have high vifs. Oftentimes, in basketball, analysts look at assist to turnover ratio, as that indicates not only if a player is good at getting assists, but also whether they can get them efficiently. In order to account for this, I plan to add an assist to turnover ratio statistic, and use that in the new model.

Making a New Model with No Multicollinearity

Using the modifications I suggested above, let's make a new model, and hope to find reasonable vif values.

```
nonRookieContracts$ASTtoTOV <- (nonRookieContracts$AST / nonRookieContracts$TOV)

newModel <- lm(PercentOfCap ~ Age + GS + eFG + FT + TRB + AST + ASTtoTOV + STL + BLK + PF + PTS, data =

summary(newModel)
```

```
##
## Call:
## lm(formula = PercentOfCap ~ Age + GS + eFG + FT + TRB + AST +
##     ASTtoTOV + STL + BLK + PF + PTS, data = nonRookieContracts)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.141291 -0.024971 -0.002442  0.019893  0.203625
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0907505  0.0244411  -3.713 0.000241 ***
## Age          0.0050241  0.0006695   7.504 5.99e-13 ***
## GS           0.0003363  0.0001680   2.002 0.046095 *
## eFG         -0.0855821  0.0384423  -2.226 0.026683 *
## FT           0.0179566  0.0211255   0.850 0.395951
## TRB          0.0052833  0.0018289   2.889 0.004127 **
## AST          0.0180950  0.0025851   7.000 1.48e-11 ***
## ASTtoTOV    -0.0083212  0.0035509  -2.343 0.019711 *
## STL         -0.0077110  0.0107573  -0.717 0.474002
## BLK          0.0185440  0.0095429   1.943 0.052852 .
## PF          -0.0326245  0.0056033  -5.822 1.39e-08 ***
## PTS          0.0070377  0.0007520   9.359  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04374 on 325 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7674, Adjusted R-squared:  0.7595
## F-statistic: 97.46 on 11 and 325 DF,  p-value: < 2.2e-16
```

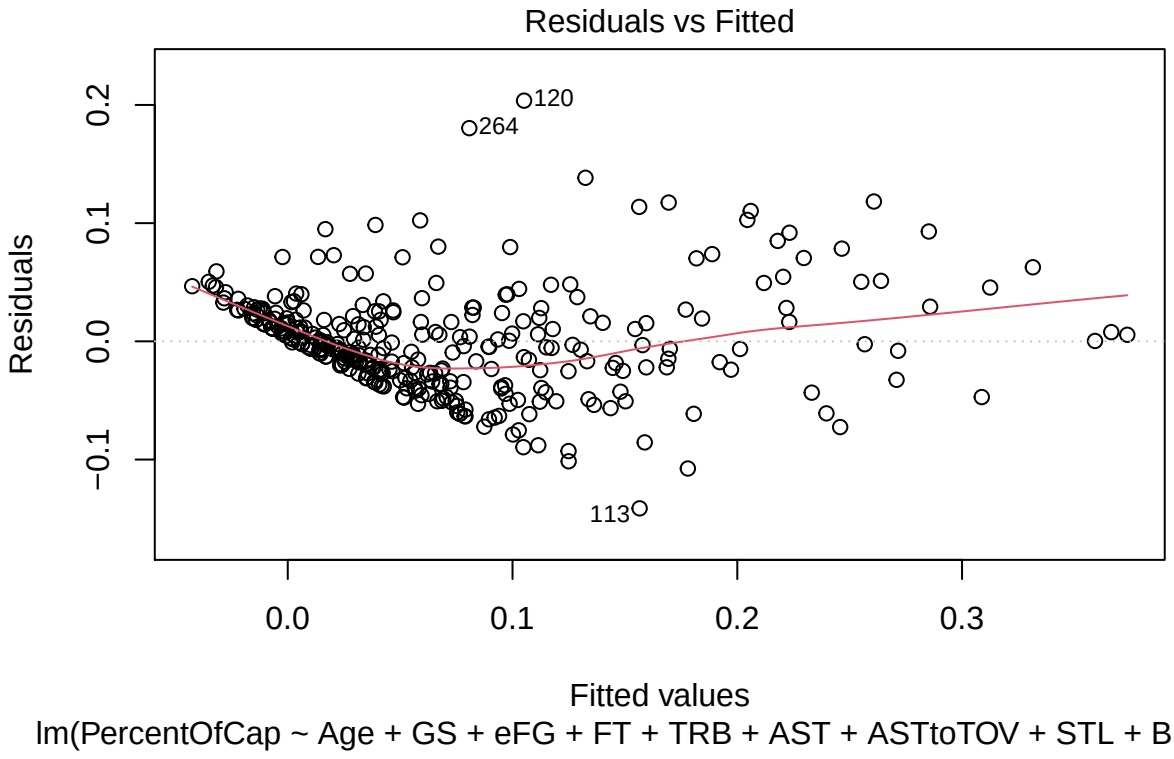
As we can see, our R-squared went down slightly from .7816 to .7674, but that change is small. Let's check the variation inflation factors again.

```
vif(newModel)
```

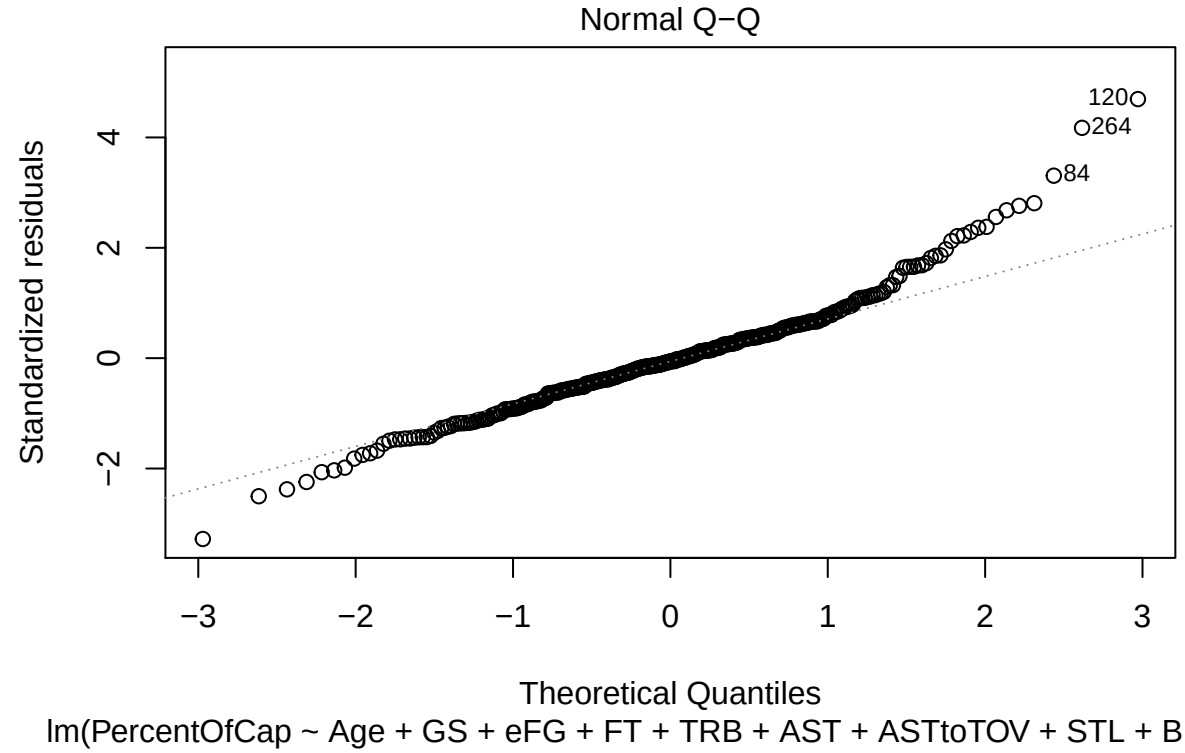
```
##      Age       GS      eFG       FT      TRB      AST ASTtoTOV      STL
## 1.257273 2.474120 1.322552 1.424377 3.119139 3.933663 1.734775 2.702954
##      BLK       PF      PTS
## 2.305665 2.619163 3.595557
```

As we can see, this time, we have no vif value over 5, which is a good sign, meaning that multicollinearity is no longer an issue. This model seems to do a pretty good job of predicting percent of salary cap, as 76.74% of variation in percent of cap is predicted by our predictor variables. Let's lastly take a look at the residual plots to make sure those conditions are satisfied.

```
plot(newModel, which = 1)
```




```
plot(newModel, which = 2)
```



As we can see in the residuals plot, the red line does not follow straight across perfectly, as it deviates from the middle near the beginning and end, but overall is in the middle. In the normal qq-plot, we can see that

the same deviation near the beginning and end occur. However, it follows nicely along the middle, so we can confidently say that the residual conditions have been satisfied.

This now means that our model we built earlier is valid. As we can see in the summary, all variables except steals, blocks, and free throw percentage are statistically significant. Surprisingly, effective field goal percentage, assist to turnover ratio, and steals all have negative coefficients, suggesting that having lower stats in these categories leads to a higher percentage of cap. However, while this is weird and intuitively seems wrong, most other coefficients are positive.

Making Predictions Using Our Model

Building a model is pointless if it can't be used. Below, I made a function named "salaryPrediction" that will take counting statistics in and produce the percent of salary cap.

```
salaryPrediction <- function(model, player, Age, GS, eFG, FT, TRB, AST, ASTtoTOV, STL, BLK, PF, PTS){
  predictionDataFrame = data.frame(Age, GS, eFG, FT, TRB, AST, ASTtoTOV, STL, BLK, PF, PTS)
  predictedSalary <- predict(model, predictionDataFrame)
  print(paste(player, ":", format(predictedSalary)))
}
```

Testing the Model

Now, let's test it on a few notable players.

First up, let's look at Luka Doncic. Because he was on his rookie contract, his statistics were not accounted for in the model we built. Luka just finished his third year in the NBA, is viewed as arguably the brightest young star in the league, and because of back to back All-NBA first teams, is eligible for a rookie-scale supermax, which is a 5 year extension that pays him 30% of the salary cap.

```
salaryPrediction(newModel, "Luka Doncic", 21, 66, 0.550, 0.730, 8.0, 8.6, 2, 1.0, 0.5, 2.3, 27.7)
```

```
## [1] "Luka Doncic : 0.3056989"
```

As seen above, our model almost perfectly predicts Luka Doncic being worth 30% of the salary cap, although most people would argue he is worth even more based on his potential.

Let's take another look at another interesting case: Mikal Bridges. Mikal also just finished his third year in the NBA, making him eligible for an extension, and has played a huge role in helping the Phoenix Suns in their current playoff run that is still going on as of the dat of me writing this. Mikal is a 3 and D player, and while his statistics may not pop off the chart, his value is undeniable, and experts see him signing a contract up to 20 million and maybe even more based on his growth and potential for continual growth.

```
salaryPrediction(newModel, "Mikal Bridges", 24, 72, 0.643, 0.84, 4.3, 2.1, 2.62, 1.1, 0.9, 1.6, 13.5)
```

```
## [1] "Mikal Bridges : 0.1040286"
```

Our model predicts that Bridges is worth around 10.4% of the salary cap, or somewhere around 12 million annually. Any analyst would see a contract like that for a player like Bridges as an absolute steal. Our model likely falls short because it cannot take into account Bridges's impact on the defensive end, along with his potential for growth.

Finally, let's take a look at Kyle Lowry, a 35 year old veteran point guard who has played at an all-star level for 6-7 years. Known for his hard-nosed defense, reliable scorer, and veteran leadership, he is valued highly around the league. However, due to his rising age and injuries, teams may be more reluctant to sign him to a longer and larger contract as he enters free agency this offseason (The Basketball Reference website has his age at 34).

```
salaryPrediction(newModel, "Kyle Lowry", 34, 46, 0.546, 0.875, 5.4, 7.3, 2.7, 1.0, 0.3, 3.1, 17.2)
```

```
## [1] "Kyle Lowry : 0.2204424"
```

Our model predicts Lowry is worth around 22% of the salary cap, or assuming the 112.4 salary cap for the 2021-22 season, 24.7 million dollars annually. While that might be considered expensive by some for an aging point guard, his level of play and leadership may lead a contending team to offer him that type of money, if not even more.

Shortcomings

Our model, despite being somehwat accurate, and fitted to account for around 77% of the variability in percent of cap, is not perfect. Numbers can only help in so many ways, and there are a nunber of shortcomings that our model has.

1. Playoff Production

While regular season production is important, playoff production often can lead to a player having their stock boosted or being frowned upon. For example, Ben Simmons in the regular season was productive, but due to his playoff shortcomings, is likely not viewed as the same level of player as he was before.

2. No Potential/Historical Context

When viewing younger players, teams often offer contracts not only based on the level of play they currently are at, but the potential that they could be something much greater. A good example was Mikal Bridges, as seen above. Additionally, previous seasons was not taken into account, such as whether or not their success was sustained, whether or not the had injuries in the past, and whether or not they have followed a steady growth and are looking to continue to grow.

3. Purely Statistical

Finally, and perhaps most obviously, the model is purely statistical. Looking at numbers alone is no way to determine a player's worth, as many players often provide much more off the court and things that can't be easily tracked, such as their defense, leadership, or IQ.

Final Discussion

In this project, using multiple regression, we were able to construct a working model that helped predict player's salary based on their statistical performance. However, it is important to note that while our model seems to work fairly well, it is far from perfect, and has many potential downfalls.

Regardless, this project was fun to build, and definitely is interesting to take a look at. It'll be interesting to see how players are paid in this upcoming offseason and see how well my model fares.