ECON 104 Project 1

Ishaan Shah, Tushar Malhotra, Austin Gigi, Avanish Jay Arun

4/9/2022

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df <- read_csv('nba_salary_final.csv')
```

```
## Warning: Missing column names filled in: 'X1' [1]

## Warning: Duplicated column names deduplicated: 'X1' => 'X1_1' [2]

##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   Player = col_character(),
##   Pos = col_character(),
##   Tm = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
df <- na.omit(df)
```

```
df_updated <- df[,c(1,6,8,10,11,13,16,18,19,21,22,24,25,26,28,31:35,37,38)] # Keeping only relevant col
```

This is a descriptor of the predictors used:
SalStartYr - The year the player joined the NBA
Pos - Position the player plays
Age - Age of the player that year
G - Games played that season

FG - The amount of baskets the player made that season per game
FG% - Field goals made/ Total field goals attempted
2P - The amount of 2 pointers the player made that season per game
2P% - 2 pointers made/ Total 2 pointers attempted
3P - The amount of 3 pointers the player made that season per game
3P% - 3 pointers made/ Total 3 pointers attempted
eFG% - A weighted average of 2 pointers made, 3 pointers made, and free throws made
FT - The amount of free throws the player made that season per game
FT% - free throws made/ Total free throws attempted
TRB - Total rebounds per game
AST - Total assists per game
STL - Total steals per game
BLK - Blocks per game
TOV- Turnovers per game
years_of_exp - How many years has the player been in the league (Rookies have 1 year of experience)

Question 1

```
dim(df_updated)
```

```
## [1] 8429   22
```

In this dataset we have 8,249 observations of NBA players' salary from 1990 to 2019. To predict their salaries we have 20 different predictors like FG%, FT%, Blocks, Assists, etc from which we want to find the best few.

```
library(gridExtra)
```

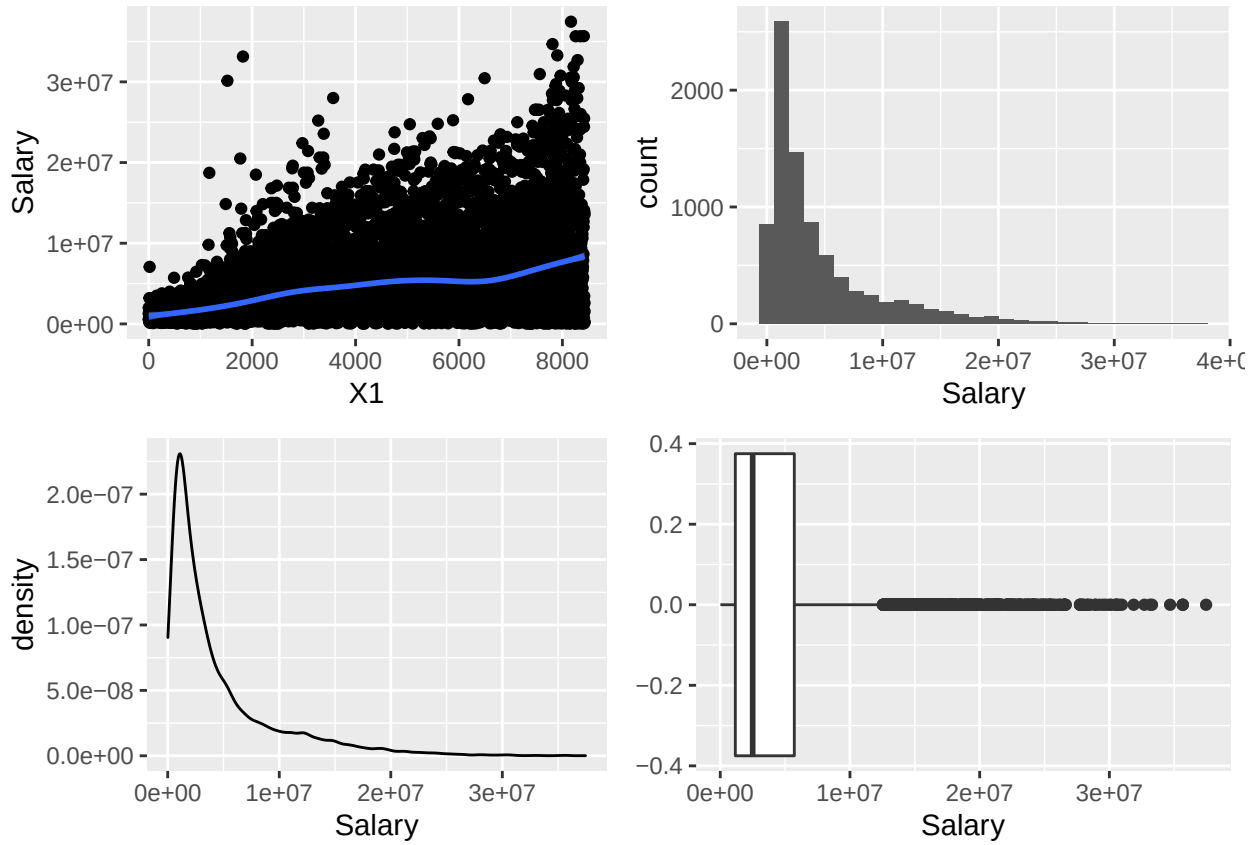
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
q1<-ggplot(data=df_updated,aes(x = X1, y = Salary))+geom_point()+geom_smooth()
q2 <- ggplot(data=df_updated,aes(x=Salary))+geom_histogram()
q3 <- ggplot(data=df_updated,aes(x = Salary))+geom_density()
q4 <- ggplot(data=df_updated,aes(x = Salary))+geom_boxplot()
grid.arrange(q1,q2,q3,q4,nrow = 2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The figures above contain a descriptive analysis of the response variable. Firstly, q1 illustrates a scatter plot of all corresponding data points against salary. The second graph q2 is a histogram of the distribution of salaries in the NBA, to which we can see that it is right-skewed. q3 and q4 are both graphs that illustrate the density of the scatter plots and we can see that they are both right-skewed as well.

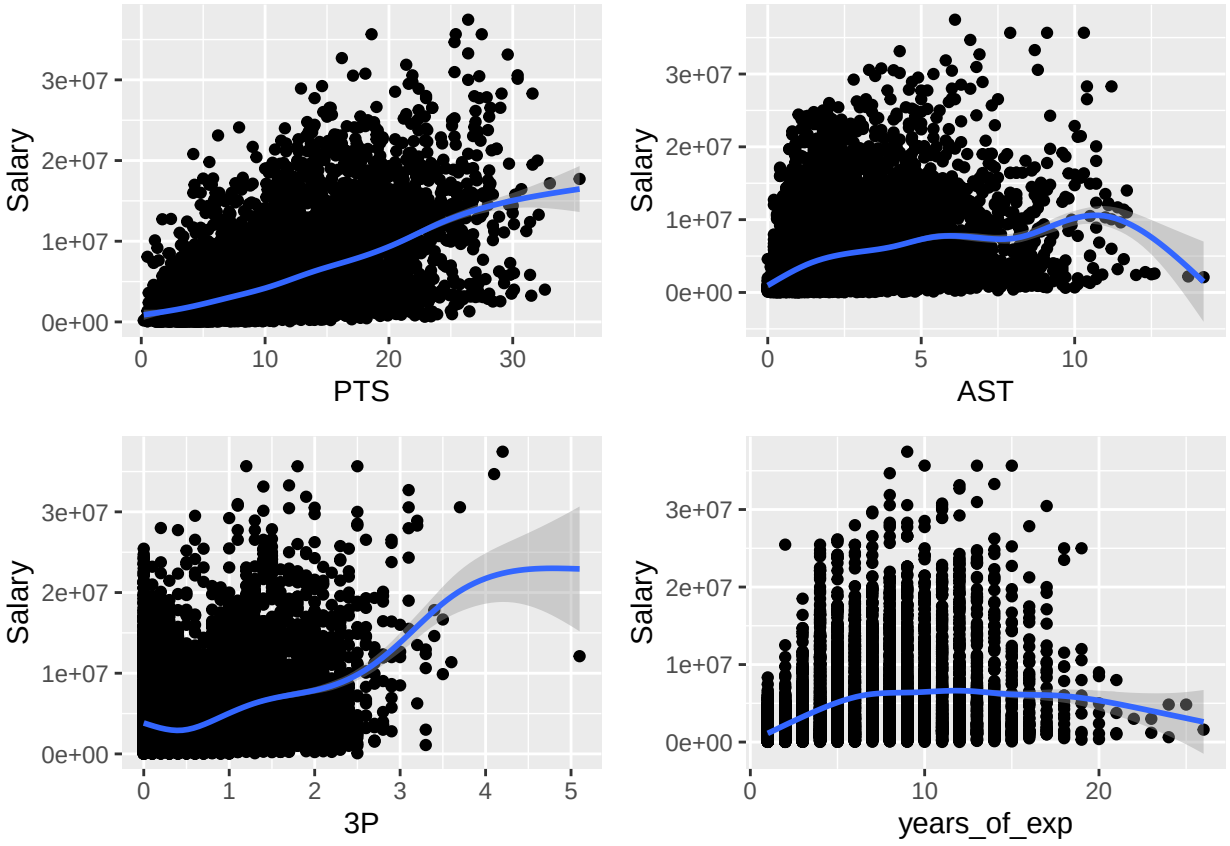
```
summary(df_updated$Salary)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    13158  1160760  2500000  4479120  5710000 37457154
```

Our statistical summary above shows that the mean NBA salary is centered at USD 4,357,257. The max NBA salary at USD 30,570,000 and the minimum being USD 13,158

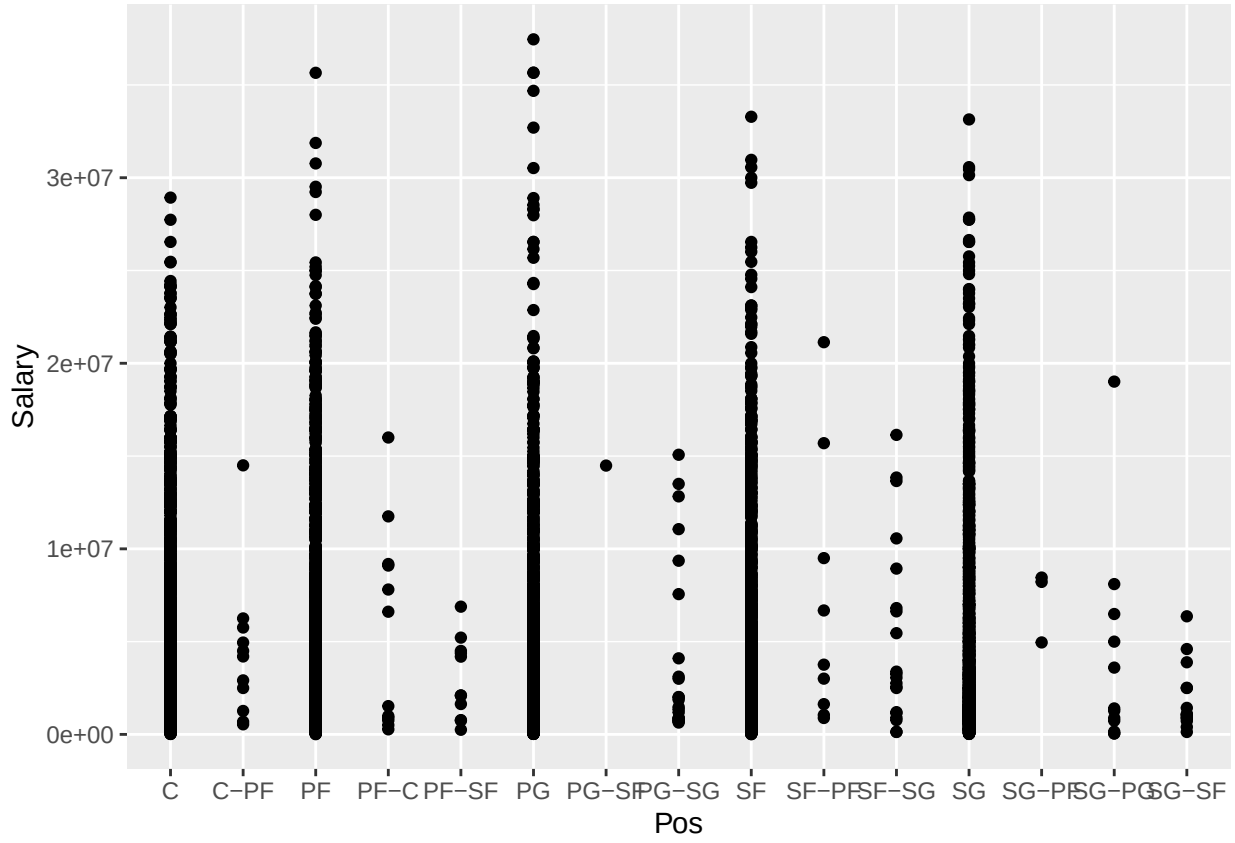
```
q5 <-ggplot(data=df_updated,aes(x = PTS, y = Salary))+geom_point()+geom_smooth()
q6 <- ggplot(data=df_updated,aes(x = years_of_exp, y = Salary))+geom_point()+geom_smooth()
q7 <- ggplot(data=df_updated,aes(x = AST, y = Salary))+geom_point()+geom_smooth()
q8 <- ggplot(data=df_updated,aes(x = `3P`, y = Salary))+geom_point()+geom_smooth()
grid.arrange(q5,q7,q8,q6,nrow = 2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



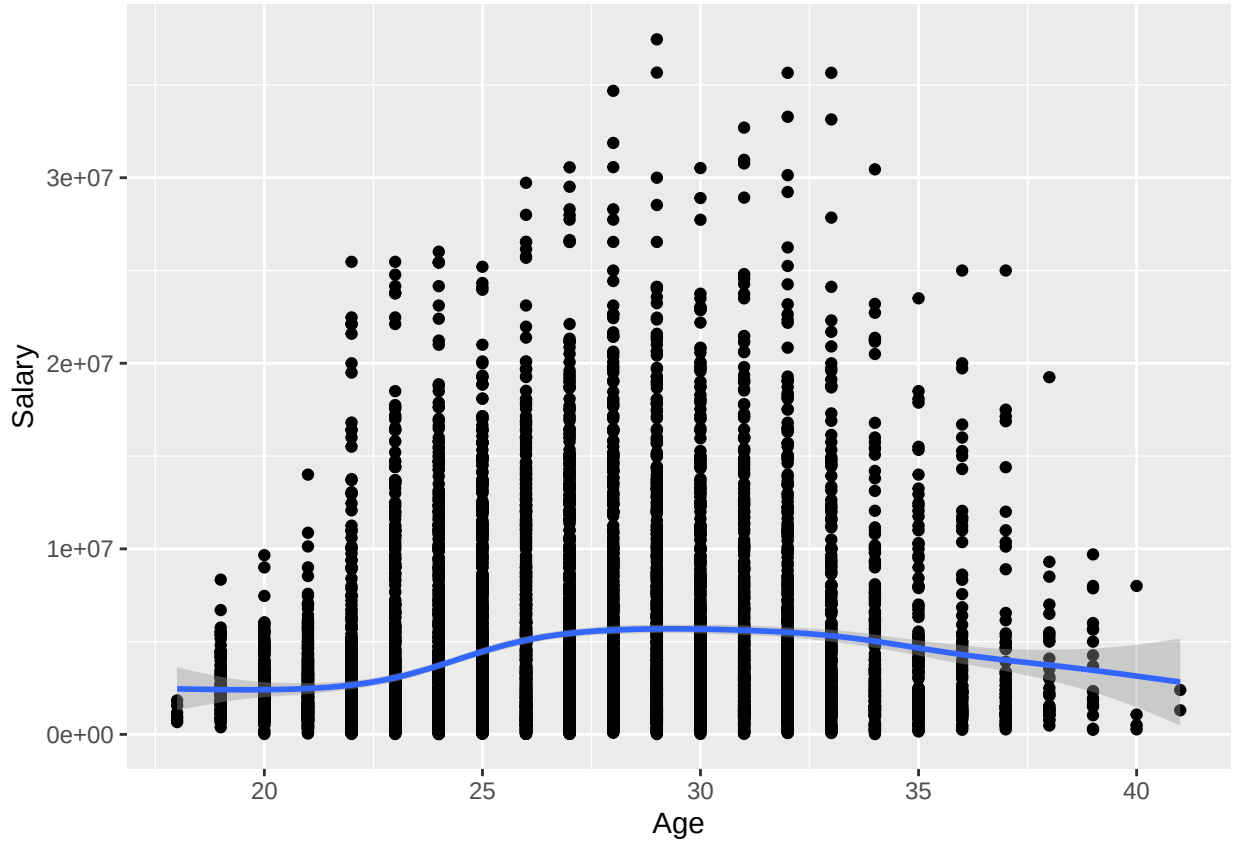
The graphs above are visualizations that illustrate the variables that we consider most important in the data set to evaluating an NBA player's salary. The first graph is a scatter plot of the points scored against the NBA players salary, and this graph has a positive correlation such that the more points that are scored the higher the salary tends to be. The second graph is an illustration of the NBA players' assists recorded against their salary, which does not have as strong a correlation as the points scored. The data points are clustered closer towards 0 as assists are relatively harder to come by as opposed to points, and this could indicate heteroskedasticity as the data points are sparsely distributed as the assist totals increase. The third graph shows the relation between NBA three-pointers made against salary which has interesting values to interpret as the majority of the dataset is clustered towards the values 0, 1 and 2 three pointers made whereas the salaries don't show much correlation with how many threes were made as the values are sparsely distributed. Finally, the last graph is a distribution of the years of experience in the NBA against salaries, and this very nearly represents a bell-shaped curve, indicating players in their prime years tend to earn the highest salary.

```
par(mfrow = c(2,1))
ggplot(data=df_updated,aes(x = Pos, y = Salary))+geom_point()
```




```
ggplot(data=df_updated,aes(x = Age, y = Salary))+geom_point()+geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



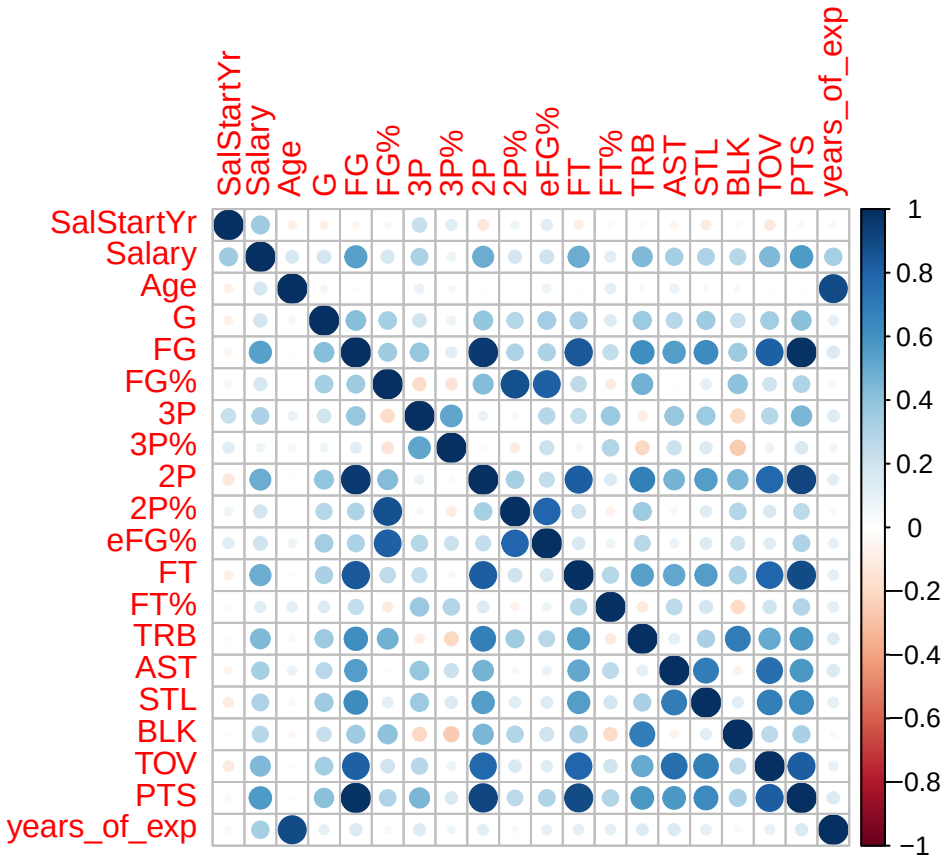
The first graph is a distribution of positions of NBA players against their respective salaries. This graph clearly shows somewhat of a uniform distribution that is weighted heavily towards players who only play one position in the NBA. It was an interesting observation to see that most NBA players who play multiple positions for their team do not earn as much as set players who only play one position.

The second graph is a distribution of the age of NBA players against their salaries. This is closely correlated to the years of experience variable that we saw in the last series of graphs, so we can expect similarities in both graphs as they both have a normal distribution. The peak in salary most prominently happens towards players who are around the mean which is 26.74528, which is typically the age when most NBA players are in their prime and have higher valuation in their salaries.

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
df_updated_cor <- cor(df_updated[,c(-1,-4)]) #Removing Position as it is a character
corrplot(df_updated_cor)
```




```
df_updated_cor_df <- data.frame(df_updated_cor)
df_updated_cor_df[,"Salary", drop = F]
```

```
##                  Salary
## SalStartYr   0.35723125
## Salary       1.00000000
## Age          0.17087282
## G            0.18741948
## FG           0.54938797
## FG%          0.17995396
## 3P           0.31199444
## 3P%          0.07521186
## 2P           0.49023845
## 2P%          0.18552199
## eFG%         0.20322910
## FT           0.49989632
## FT%          0.12597407
## TRB          0.44113517
## AST          0.33083727
## STL          0.30918321
## BLK          0.28077230
## TOV          0.44655015
## PTS          0.56303645
## years_of_exp 0.33226801
```

The variables highest correlated with Salary look like PTS, FG, 2 pointers made and Free throws made.

However, from the correlation plot we can see a lot of these variables are highly correlated with each other as they measure similar statistics. We will need to take this into account later to avoid multicollinearity.

Question 2

```
lin_reg_1 <- lm(Salary~., data = df_updated[,-1])
summary(lin_reg_1)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = df_updated[, -1])
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -13187575  -2046779   -236849   1602789  21391076
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -429219938   10524050 -40.785  < 2e-16 ***
## SalStartYr       215783       5176  41.687  < 2e-16 ***
## PosC-PF         -715344    1022467  -0.700 0.484181
## PosPF           -322445     138578  -2.327 0.019999 *
## PosPF-C          687511     980405   0.701 0.483165
## PosPF-SF       -1024540    1025740  -0.999 0.317905
## PosPG           -910517     207122  -4.396 1.12e-05 ***
## PosPG-SF        6074683    3380023   1.797 0.072335 .
## PosPG-SG        -975046     740476  -1.317 0.187947
## PosSF           -524960     163636  -3.208 0.001341 **
## PosSF-PF        1731711    1133037   1.528 0.126456
## PosSF-SG         859494     770017   1.116 0.264368
## PosSG           -629313     180595  -3.485 0.000495 ***
## PosSG-PF       -1415717    1956282  -0.724 0.469284
## PosSG-PG        -590548     990300  -0.596 0.550969
## PosSG-SF       -1003240     915743  -1.096 0.273308
## Age              -87846      22077  -3.979 6.98e-05 ***
## G                -16317       2187  -7.461 9.47e-14 ***
## FG              2297489    1169402   1.965 0.049485 *
## `FG%`          19316422    2688355   7.185 7.28e-13 ***
## `3P`             445770    1049084   0.425 0.670911
## `3P%`            166901     302425   0.552 0.581050
## `2P`           -1517133     835645  -1.816 0.069479 .
## `2P%`            398718    1344033   0.297 0.766735
## `eFG%`        -21854117    2339207  -9.343  < 2e-16 ***
## FT               762327     499400   1.526 0.126926
## `FT%`           -848835     391598  -2.168 0.030216 *
## TRB              318623      30486  10.451  < 2e-16 ***
## AST              490626      42709  11.488  < 2e-16 ***
## STL             -833958     128399  -6.495 8.77e-11 ***
## BLK              684321     110931   6.169 7.19e-10 ***
## TOV             -526988     116039  -4.541 5.66e-06 ***
## PTS             -159745     496648  -0.322 0.747728
```

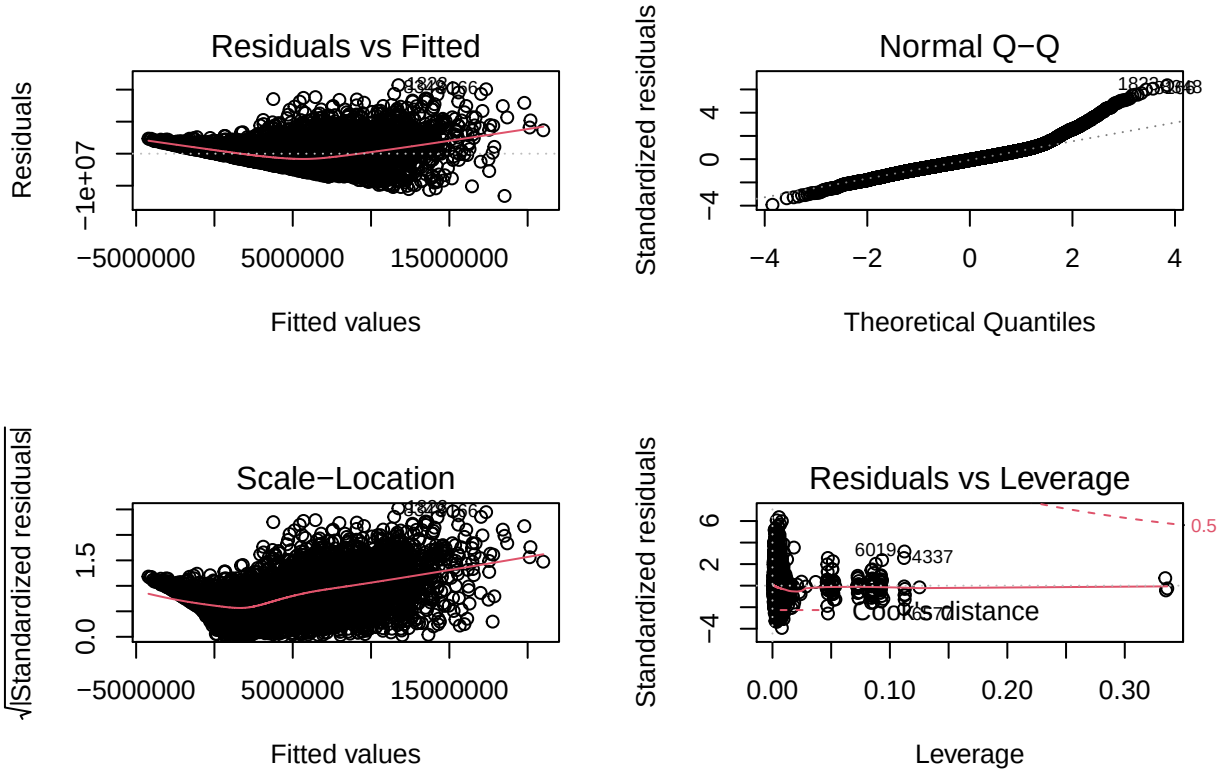
```
## years_of_exp     356823      23003  15.512  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3372000 on 8395 degrees of freedom
## Multiple R-squared:  0.5534, Adjusted R-squared:  0.5517
## F-statistic: 315.3 on 33 and 8395 DF,  p-value: < 2.2e-16
```

Base model using all predictors has multiple insignificant variables, as seen by the high p values of a lot of the predictors. However, this model as a whole is significant, that is it has multiple coefficients with values not equal to 0 as shown by the low p-value of the F statistic. Moreover, these coefficients show how a unit change in the predictor variable affects salary, for example every additional year of experience adds $356,823 to one's salary. But, I wouldn't rely on these predictors as many of the are insignificant and have wrong signs, possibly due to multi-collinearity. These predictors explain about ~ 55% of the variation in Salary.

Question 3

```
par(mfrow = c(2,2))
plot(lin_reg_1)
```

```
## Warning: not plotting observations with leverage one:
##   3999
```



```
leverage <- hatvalues(lin_reg_1)
leverage_points <- df_updated$X1[leverage>=4/8429]
outliers <- df_updated$X1[abs(rstandard(lin_reg_1))>=4]
length(intersect(leverage_points, outliers))
```

```
## [1] 44
```

```
points_to_remove <- intersect(leverage_points, outliers)
df_updated <- df_updated %>%
              filter(!row_number() %in% points_to_remove)
```

We identified 44 bad leverage points. As one can see in the last plot, there are a few points with really high leverage, hence we decided to remove them. Also, it is only 44 points out of 8429 so it should be fine to remove them.

```
lin_reg_1 <- lm(Salary~., data = df_updated[,-1])
summary(lin_reg_1)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = df_updated[, -1])
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -12691072  -1949116   -233686   1565671  13904267
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -408882728    9908056 -41.268  < 2e-16 ***
## SalStartYr       205572       4874  42.178  < 2e-16 ***
## PosC-PF         -650446     959417  -0.678 0.497816
## PosPF           -304159     130298  -2.334 0.019602 *
## PosPF-C          777903     919950   0.846 0.397804
## PosPF-SF        -949901     962497  -0.987 0.323714
## PosPG           -869474     194803  -4.463 8.17e-06 ***
## PosPG-SF        6196494    3171576   1.954 0.050763 .
## PosPG-SG        -813226     694877  -1.170 0.241907
## PosSF           -532454     153827  -3.461 0.000540 ***
## PosSF-PF        1846454    1063170   1.737 0.082469 .
## PosSF-SG         922580     722562   1.277 0.201702
## PosSG           -558714     169693  -3.293 0.000997 ***
## PosSG-PF       -1185741    1835641  -0.646 0.518326
## PosSG-PG        -479729     929270  -0.516 0.605699
## PosSG-SF        -993502     859284  -1.156 0.247633
## Age              -86201      20736  -4.157 3.26e-05 ***
## G                -14224       2057  -6.915 5.04e-12 ***
## FG               2248864    1101310   2.042 0.041184 *
## `FG%`           17775868    2530536   7.025 2.32e-12 ***
## `3P`               89718     988010   0.091 0.927648
## `3P%`             217790     283985   0.767 0.443160
## `2P`            -1622647     787186  -2.061 0.039303 *
## `2P%`            -255792    1262670  -0.203 0.839468
```

```
## `eFG%`        -19910020    2201962  -9.042  < 2e-16 ***
## FT               688070     470222   1.463 0.143426
## `FT%`           -631321     368050  -1.715 0.086325 .
## TRB              302080      28685  10.531  < 2e-16 ***
## AST              420775      40326  10.434  < 2e-16 ***
## STL             -775084     120957  -6.408 1.55e-10 ***
## BLK              704678     104271   6.758 1.49e-11 ***
## TOV             -326222     109490  -2.979 0.002896 **
## PTS             -107000     467597  -0.229 0.819006
## years_of_exp     348422      21607  16.126  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3164000 on 8351 degrees of freedom
## Multiple R-squared:  0.5576, Adjusted R-squared:  0.5558
## F-statistic: 318.9 on 33 and 8351 DF,  p-value: < 2.2e-16
```

There is an increase in R^2 on the base model.

Question 4

```
library(Boruta)
library(car)
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.0.5
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(AER)
```

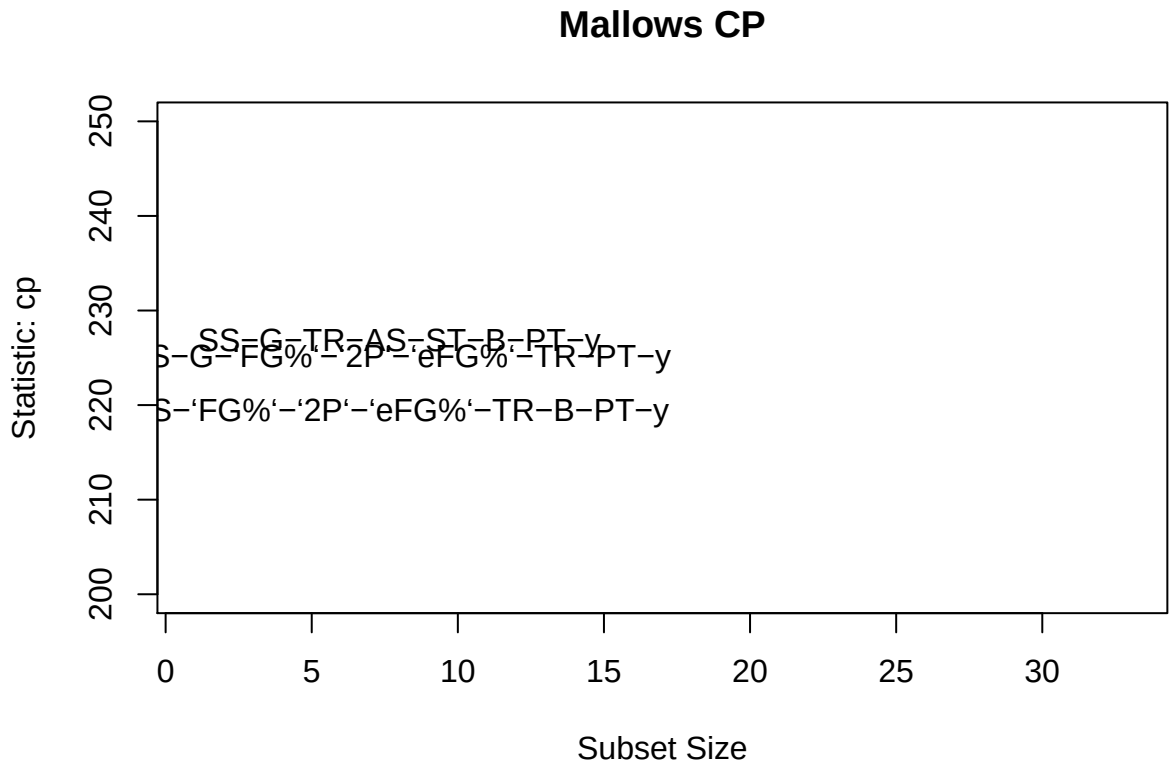
```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival
```

```
library(leaps)
ss <- regsubsets(Salary~.,method=c("exhaustive"),nbest=3,data=df_updated[,-1])
subsets(ss,statistic="cp",legend=F,main="Mallows CP",col="steelblue4", ylim= c(200,250))
```




```
##              Abbreviation
## SalStartYr             SS
## PosC-PF                PC
## PosPF                PsPF
## PosPF-C             PPF-C
## PosPF-SF            PPF-S
## PosPG                PsPG
## PosPG-SF           PPG-SF
## PosPG-SG           PPG-SG
## PosSF                PsSF
## PosSF-PF            PSF-P
## PosSF-SG            PSF-S
## PosSG                PsSG
```

```
## PosSG-PF           PSG-PF
## PosSG-PG           PSG-PG
## PosSG-SF            PSG-S
## Age                   Ag
## G                      G
## FG                    FG
## `FG%`               `FG%`
## `3P`                `3P`
## `3P%`               `3P%
## `2P`                `2P`
## `2P%`               `2P%
## `eFG%`             `eFG%`
## FT                    FT
## `FT%`               `FT
## TRB                   TR
## AST                   AS
## STL                   ST
## BLK                    B
## TOV                   TO
## PTS                   PT
## years_of_exp           y
```

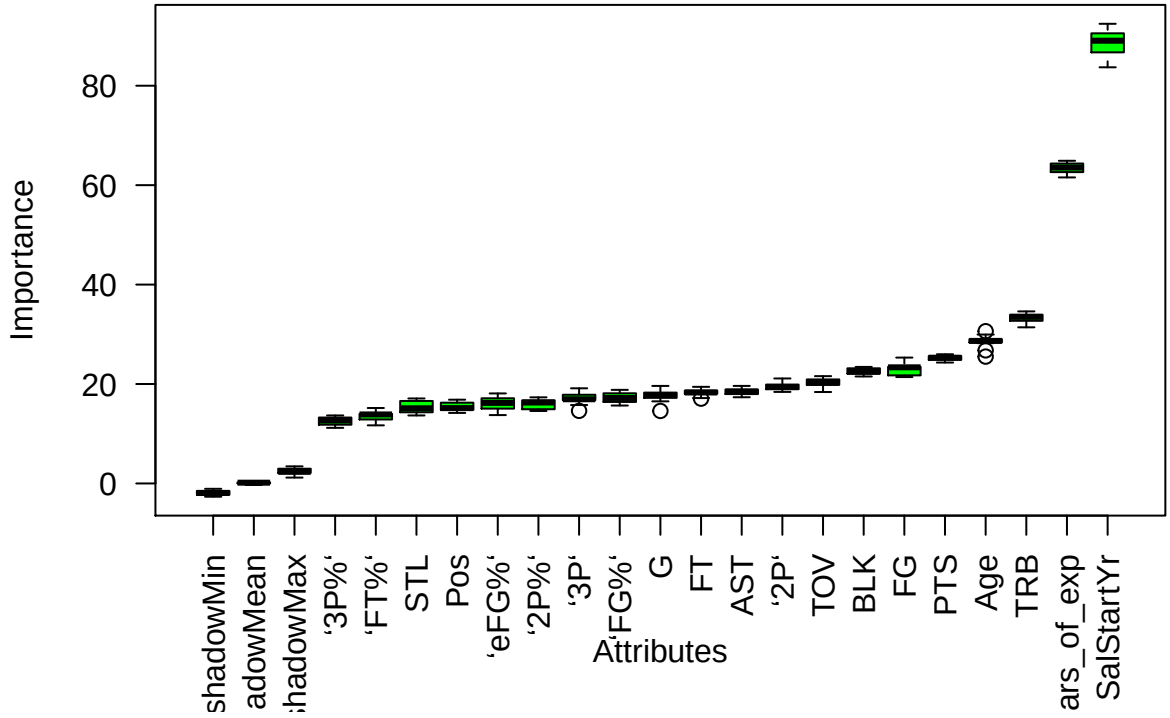
Salary Start Year, FG%, 2P, eFG%, TR, B, PTS, years of experience are the main variables identified by Mallows CP

```
Bor.res <- Boruta(Salary ~., data = df_updated[,-1], doTrace = 2)
```

```
##  1. run of importance source...

##  2. run of importance source...

##  3. run of importance source...

##  4. run of importance source...

##  5. run of importance source...

##  6. run of importance source...

##  7. run of importance source...

##  8. run of importance source...

##  9. run of importance source...

##  10. run of importance source...

##  11. run of importance source...

## After 11 iterations, +2 mins:
```

```
##  confirmed 20 attributes: ‘2P%‘, ‘2P‘, ‘3P%‘, ‘3P‘, ‘eFG%‘ and 15 more;

##  no more attributes left.
```

```
plot(Bor.res,sort=T, las = 2)
```




```
boruta_signif_conf <- names(Bor.res$finalDecision[Bor.res$finalDecision %in% c("Confirmed")])
print(boruta_signif_conf)
```

```
##  [1] "SalStartYr"   "Pos"          "Age"          "G"            "FG"
##  [6] "‘FG%‘"        "‘3P‘"         "‘3P%‘"        "‘2P‘"         "‘2P%‘"
## [11] "‘eFG%‘"       "FT"           "‘FT%‘"        "TRB"          "AST"
## [16] "STL"          "BLK"          "TOV"          "PTS"          "years_of_exp"
```

The list above are the main variables identified by the Boruta Algorithm

```
lin_reg_2 <- lm(Salary ~ years_of_exp+TRB+PTS + BLK + SalStartYr + G + ‘FG%‘+ TOV + ‘eFG%‘+ Age + ‘3P‘ +
summary(lin_reg_2)
```

```
##
## Call:
## lm(formula = Salary ~ years_of_exp + TRB + PTS + BLK + SalStartYr +
##     G + ‘FG%‘ + TOV + ‘eFG%‘ + Age + ‘3P‘ + ‘2P‘ + AST, data = df_updated)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -12361060  -1978094   -239190   1577196  14240803
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -418720669    9731322 -43.028  < 2e-16 ***
## years_of_exp     354314      21588  16.412  < 2e-16 ***
## TRB              330493      25083  13.176  < 2e-16 ***
## PTS              570712      46950  12.156  < 2e-16 ***
## BLK              808829      98460   8.215 2.44e-16 ***
## SalStartYr       209790       4790  43.801  < 2e-16 ***
## G                -16118       2042  -7.895 3.27e-15 ***
## `FG%`          20029259    2088110   9.592  < 2e-16 ***
## TOV             -337357     109136  -3.091    0.002 **
## `eFG%`        -21514642    2003774 -10.737  < 2e-16 ***
## Age               -84090      20655  -4.071 4.72e-05 ***
## `3P`              250957     191915   1.308    0.191
## `2P`             -783002     119232  -6.567 5.44e-11 ***
## AST               257059      32125   8.002 1.39e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3179000 on 8371 degrees of freedom
## Multiple R-squared:  0.5524, Adjusted R-squared:  0.5517
## F-statistic: 794.6 on 13 and 8371 DF,  p-value: < 2.2e-16
```

Used the most significant variables as demonstrated by the 2 methods above. This model looks to be better than the previous model as almost all variables are significant, and R^2 is almost the same as before despite removing 10+ predictors. However, some signs don't make sense, for example according to the model the more 2 pointers a player makes, their salary decreases. The issue in signs might indicate high multicollinearity in the model.

Question 5

```
vif(lin_reg_2)
```

```
## years_of_exp          TRB          PTS          BLK   SalStartYr            G
##     5.995734     3.368195    64.687751     2.112584     1.248309     1.380865
##        `FG%`          TOV       `eFG%`          Age         `3P`         `2P`
##    12.972713     6.472492    11.398914     5.950237    13.414591    48.584525
##          AST
##     3.245734
```

Multiple VIFs above 5, indicating multicollinearity. This maybe due to the fact that a lot of these metrics take into account other variables in the equation. For example, PTS (points) scored would be directly related with 3 pointers made and 2 pointers made. So we have to drop variables which are connected to each other.

```
lin_reg_3 <- lm(Salary ~ years_of_exp+TRB+PTS + BLK + SalStartYr + G + `FG%`+ TOV + `eFG%`  , data = df
summary(lin_reg_3)
```

```
##
## Call:
## lm(formula = Salary ~ years_of_exp + TRB + PTS + BLK + SalStartYr +
##     G + `FG%` + TOV + `eFG%`, data = df_updated)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -11773619  -1969637   -255893   1590061  15834949
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -443745476    9247302 -47.986  < 2e-16 ***
## years_of_exp     288045       9162  31.438  < 2e-16 ***
## TRB              233267      24240   9.623  < 2e-16 ***
## PTS              352258      12081  29.159  < 2e-16 ***
## BLK              620705      97918   6.339 2.43e-10 ***
## SalStartYr       221306       4617  47.937  < 2e-16 ***
## G                -12610       2031  -6.208 5.63e-10 ***
## `FG%`           2794694    1233779   2.265 0.023529 *
## TOV              289266      80942   3.574 0.000354 ***
## `eFG%`          -5806217    1207920  -4.807 1.56e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3218000 on 8375 degrees of freedom
## Multiple R-squared:  0.5411, Adjusted R-squared:  0.5406
## F-statistic:  1097 on 9 and 8375 DF,  p-value: < 2.2e-16
```

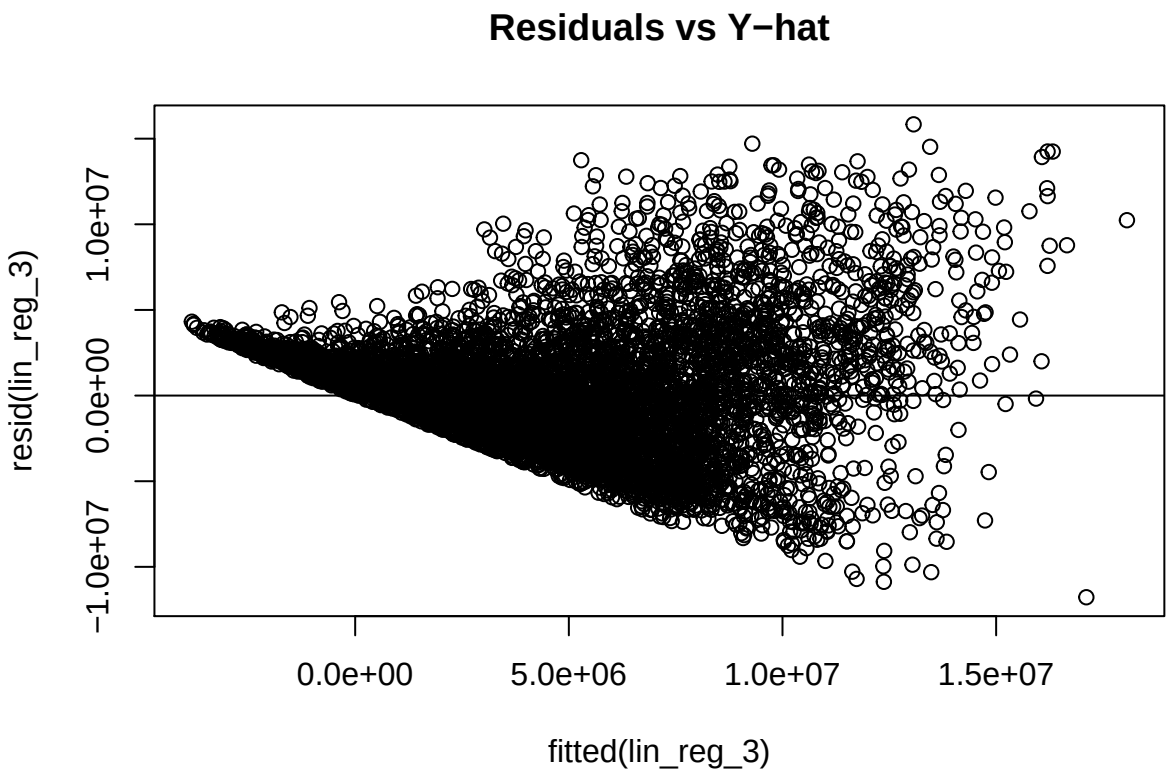
```
vif(lin_reg_3)
```

```
## years_of_exp          TRB          PTS          BLK   SalStartYr            G
##     1.053973     3.069641     4.179701     2.039028     1.131803     1.334245
##        `FG%`          TOV        `eFG%`
##     4.419841     3.474460     4.042505
```

The VIFs here look way better. None of them cross 5, indicating no multi-collinearity. All the predictors are significant, and R^2 is not far off from the base model.

Question 6

```
plot(fitted(lin_reg_3), resid(lin_reg_3), main = "Residuals vs Y-hat")
abline(0,0)
```



This pattern depicts a funnel, that is the spread of residuals increases as y-hat increases. This shows evidence of heteroskedasticity.

Question 7

```
resettest(lin_reg_3, powers= 2:3,  type = "regressor", data = df_updated)
```

```
##
##  RESET test
##
## data:  lin_reg_3
## RESET = 34.514, df1 = 18, df2 = 8357, p-value < 2.2e-16
```

As the p-value of this test is <0.05, it means that this model would be better off with squared or cubed predictors. To determine which variables should have power terms, we should try different combinations of the most significant variables to see the impact they have on the model. This test shows that at least one of the squared continuous predictors is a statistically significant preictor for Salary.

Question 8

```
ncvTest(lin_reg_3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 3494.862, Df = 1, p = < 2.22e-16
```

```
bptest(lin_reg_3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lin_reg_3
## BP = 1781.8, df = 9, p-value < 2.2e-16
```

Both these tests have p-values have <0.05, it shows evidence of heteroskedasticity. To tackle this problem we should use feasible generalized linear squares.

```
ehatsq <- resid(lin_reg_3)^2
sighatsq.ols  <- lm(log(ehatsq)~log(years_of_exp+TRB+PTS + BLK + SalStartYr + G + `FG%`+ TOV + `eFG%`),
vari <- exp(fitted(sighatsq.ols))
```

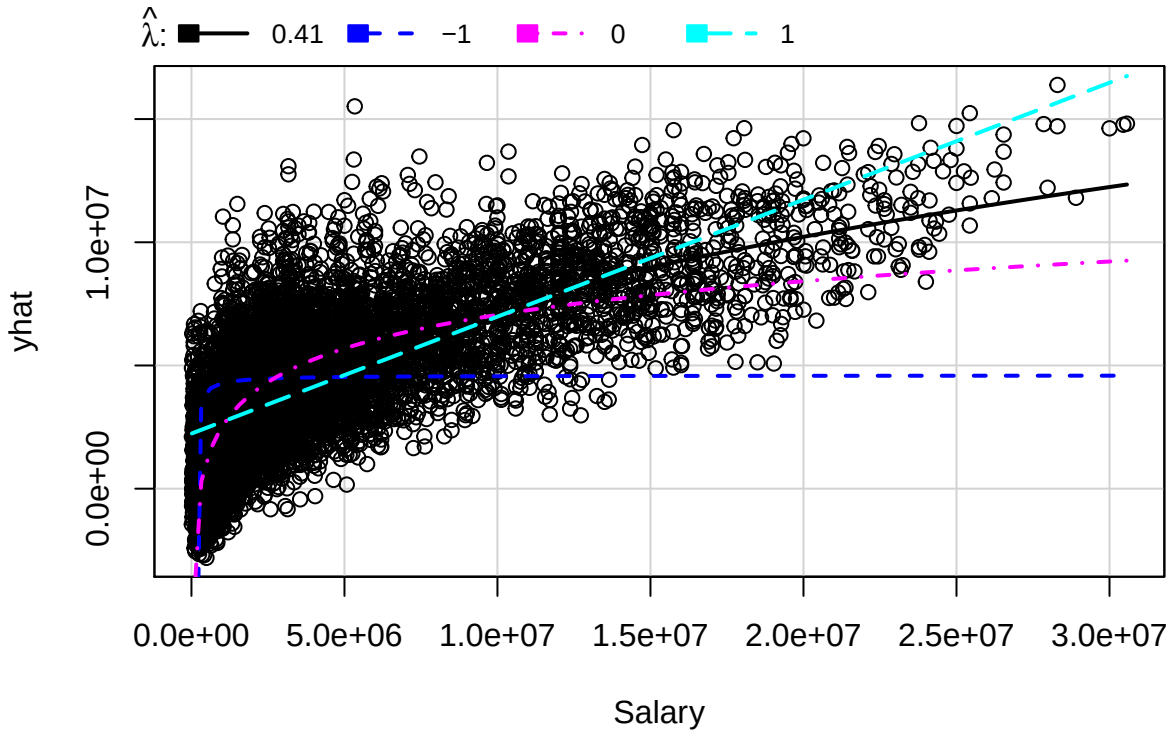
```
lin_reg_3_fgls <- lm(Salary ~ years_of_exp+TRB+PTS + BLK + SalStartYr + G + `FG%`+ TOV + `eFG%` , weigh
summary(lin_reg_3_fgls)
```

```
##
## Call:
## lm(formula = Salary ~ years_of_exp + TRB + PTS + BLK + SalStartYr +
##     G + `FG%` + TOV + `eFG%`, data = df_updated, weights = 1/vari)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9334 -1.2795 -0.2294  0.9554 13.3768
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -348692010    8175566 -42.651  < 2e-16 ***
## years_of_exp     255268       8266  30.880  < 2e-16 ***
## TRB              202262      23731   8.523  < 2e-16 ***
## PTS              320252      11815  27.105  < 2e-16 ***
## BLK              654190      97373   6.718 1.96e-11 ***
## SalStartYr       174150       4080  42.683  < 2e-16 ***
## G                -10091       1585  -6.367 2.03e-10 ***
## `FG%`           2499340    1067013   2.342  0.01918 *
## TOV              266333      75059   3.548  0.00039 ***
## `eFG%`         -5692835    1020619  -5.578 2.51e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.017 on 8375 degrees of freedom
## Multiple R-squared:  0.5164, Adjusted R-squared:  0.5159
## F-statistic: 993.8 on 9 and 8375 DF,  p-value: < 2.2e-16
```

There is a drop in R^2 and the coefficients of some of the predictors have changed due to the weights.

Question 9

```
inverseResponsePlot(lin_reg_3_fgls)
```




```
##       lambda          RSS
## 1  0.4092635 3.217152e+16
## 2 -1.0000000 7.362235e+16
## 3  0.0000000 3.637301e+16
## 4  1.0000000 3.646922e+16
```

The Inverse Response Plot shows that a log tranformation to Salary could be helpful in reducing RSS. This will be added to the final model.

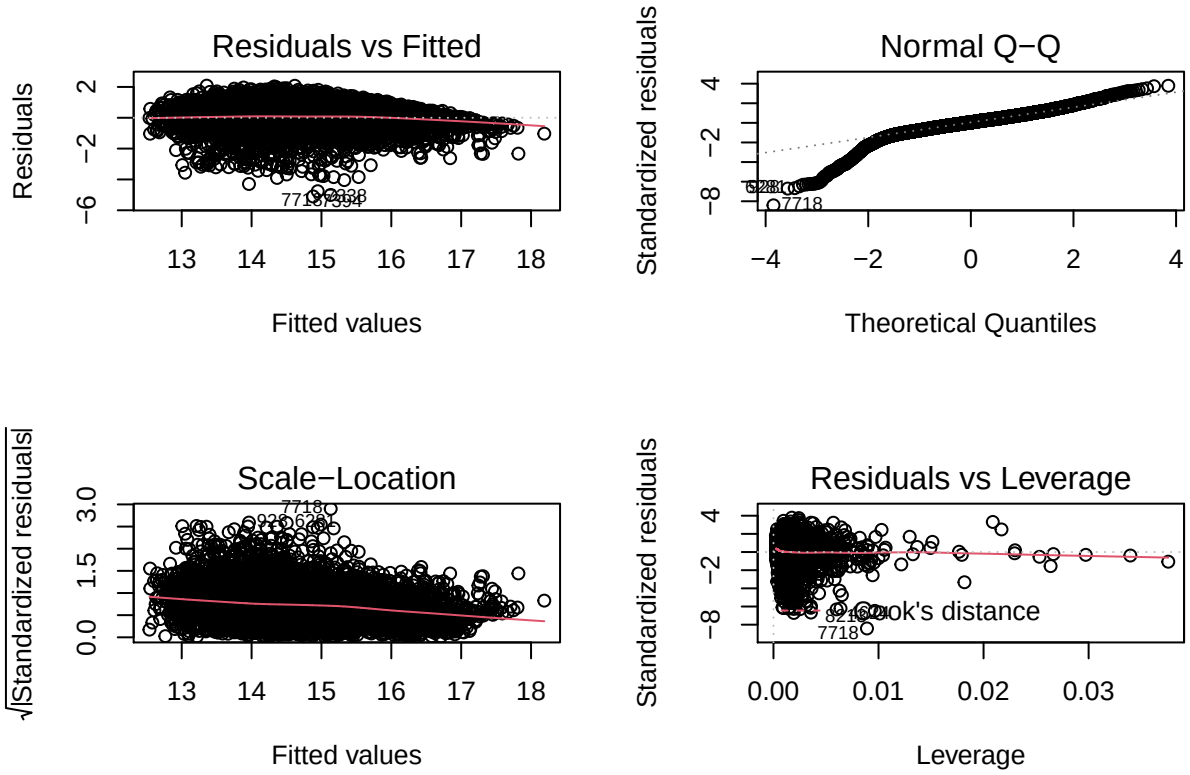
```
lin_reg_4_fgls <- lm(log(Salary) ~ years_of_exp+I(TRB^2)+PTS + BLK + SalStartYr + `FG%` + I(G^2) + TOV +
summary(lin_reg_4_fgls)
```

```
##
## Call:
## lm(formula = log(Salary) ~ years_of_exp + I(TRB^2) + PTS + BLK +
##     SalStartYr + `FG%` + I(G^2) + TOV + I(`eFG%`^2), data = df_updated,
##     weights = 1/vari)
##
## Weighted Residuals:
```

```
##        Min         1Q     Median         3Q        Max
## -4.927e-06 -2.782e-07  3.090e-08  3.329e-07  2.224e-06
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -9.283e+01  2.388e+00 -38.880  < 2e-16 ***
## years_of_exp  8.168e-02  2.395e-03  34.108  < 2e-16 ***
## I(TRB^2)      2.809e-03  5.285e-04   5.315 1.09e-07 ***
## PTS           7.025e-02  3.366e-03  20.869  < 2e-16 ***
## BLK           2.500e-01  2.728e-02   9.165  < 2e-16 ***
## SalStartYr    5.275e-02  1.184e-03  44.549  < 2e-16 ***
## `FG%`         8.015e-01  2.763e-01   2.901  0.00373 **
## I(G^2)        5.228e-05  4.720e-06  11.077  < 2e-16 ***
## TOV           1.674e-01  2.196e-02   7.624 2.74e-14 ***
## I(`eFG%`^2)  -1.545e+00  2.991e-01  -5.167 2.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.882e-07 on 8375 degrees of freedom
## Multiple R-squared:  0.5215, Adjusted R-squared:  0.5209
## F-statistic:  1014 on 9 and 8375 DF,  p-value: < 2.2e-16
```

We decided to add quadratic terms for Total Rebounds, Field Goal percentage and Effective Field Goal Percentage as these were variables that were highly significant, ranked highly in the Boruta Algorithm or were present in Mallow's CP and didn't cause multicollinearity issues. We also notice a slight increase in R^2.

```
par(mfrow = c(2,2))
plot(lin_reg_4_fgls)
```



Looks like adding the log to Salary and weighting the regression, helped with heteroskedasticy as the distribution of residuals looks way better.

```
vif(lin_reg_4_fgls)
```

```
## years_of_exp     I(TRB^2)          PTS          BLK   SalStartYr        `FG%`
##     1.056288     2.172280     4.245127     1.868648     1.117559     4.459189
##       I(G^2)          TOV I(`eFG%`^2)
##     1.385093     3.529884     4.226838
```

This model looks like our best model, as there are fewer signs of heteroskedasticity due to weighting, and addition of logs. Moreover, all predictors are signifiant and there is no multicollinearity. There is a slight dip in R^2 but it is not too far off the base model. Hence, we feel this is our best valid model.

```
AIC(lin_reg_1)
```

```
## [1] 274835.8
```

```
AIC(lin_reg_2)
```

```
## [1] 274894.1
```

```
AIC(lin_reg_3)
```

```
## [1] 275094.6
```

```
AIC(lin_reg_3_fgls)
```

```
## [1] 273859.3
```

```
AIC(lin_reg_4_fgls)
```

```
## [1] 21510.73
```

```
BIC(lin_reg_1)
```

```
## [1] 275082
```

```
BIC(lin_reg_2)
```

```
## [1] 274999.7
```

```
BIC(lin_reg_3)
```

```
## [1] 275172
```

```
BIC(lin_reg_3_fgls)
```

```
## [1] 273936.7
```

```
BIC(lin_reg_4_fgls)
```

```
## [1] 21588.1
```

Both the AIC and BIC show that this model has the lowest value, hence adding more proof to the fact that it is our best model yet, despite the slight dip in R^2.

Question 10

```
library(lmvar)
cv_fit <- lm(log(Salary) ~ years_of_exp+I(TRB^2)+PTS + BLK + SalStartYr + `FG%` + I(G^2) + TOV + I(`eFG%`
cv.lm(cv_fit, k = 5)
```

```
## Mean absolute error        :  0.5773786
## Sample standard deviation  :  0.004300257
##
## Mean squared error         :  0.5863352
## Sample standard deviation  :  0.02061988
##
## Root mean squared error    :  0.7656298
## Sample standard deviation  :  0.01351775
```

CV was done again without logs, so that the estimates of RMSE were interpretable.

```
cv_fit <- lm((Salary) ~ years_of_exp+I(TRB^2)+PTS + BLK + SalStartYr + `FG%` + I(G^2) + TOV + I(`eFG%`^
cv.lm(cv_fit, k = 5)
```

```
## Mean absolute error        :  2369104
## Sample standard deviation  :  31419.32
##
## Mean squared error         :  1.038797e+13
## Sample standard deviation  :  282795456823
##
## Root mean squared error    :  3222803
## Sample standard deviation  :  43510.83
```

Looking at our MAE and RMSE, on average each prediction was off by ~ $3 million. This is slightly concerning as Salary was right skewed with a mean of around ~ 4 million. This is mostly concerning while predicting the salaries of players who are on cheaper contracts.

```
set.seed(1234)
row.number <- sample(1:nrow(df_updated), (2/3)*nrow(df_updated))
train = df_updated[row.number,]
test = df_updated[-row.number,]
dim(train)
```

```
## [1] 5590   22
```

```
dim(test)
```

```
## [1] 2795   22
```

```
lin_reg_train <- lm(log(Salary) ~ years_of_exp+I(TRB^2)+PTS + BLK + SalStartYr + `FG%` + I(G^2) + TOV +
sqrt(mean((train$Salary - predict(lin_reg_train, train)) ^ 2))
```

```
## [1] 6421088
```

```
sqrt(mean((test$Salary - predict(lin_reg_train, test)) ^ 2))
```

```
## [1] 6489944
```

This shows that our average training error was ~ $6.4 million and testing error was slightly more at ~ 6.5 million USD. The fact that there is such a significant difference in our cross valiaiton errors and training/testing errors shows the fact that this model needs more observations to succeed.

Question 11

To conclude, we used 9 predictors and weights to predict NBA players' salaries. Our best model had a Rˆ2 of 0.52, but it was as valid as we could make it. We see Salary Start year and Years of Experience are our most powerful predictors, closely followed by more basketball related metrics such as Points, Rebounds, Games Played and Blocks. Our Rˆ2 isn't super high but there are maybe a few ways we can try to increase it in the future. We can remove/split the data as younger players with <2 years of experience have salaries that are structured differently. We can also try incorporating an inflation factor as salaries have increased over the years due to inflation and the popularity of basketball. Finally, this was a great experience to put theoretical tools we have learned in class to solve a real life question.