

Rookie sensations vs. NBA God's- NBA Salary Predictions

Stat 432 Final Project

Bryan Ramirez, Junyu Liu, Neil (Suritaneil) Sahota

II. Abstract

Our project is about trying to predict the salaries of the best NBA player of the year, known as the MVP, and the best rookie of the year. In our project, we were trying to see what the most influential player stats were in an NBA player that contributed to their salary. We found out that the most important stats that had the greatest influence on salary was GP (the amount of games a person played), GS (the amount of games a player started in a game), AST (when a person passes the ball and that person scores), TRB (when a person misses a shot and a person grabs the ball of the rim of the basket), PPM (a stat we created points divided by minutes played), Points(how many points a player scored).

III. Problem and Motivation

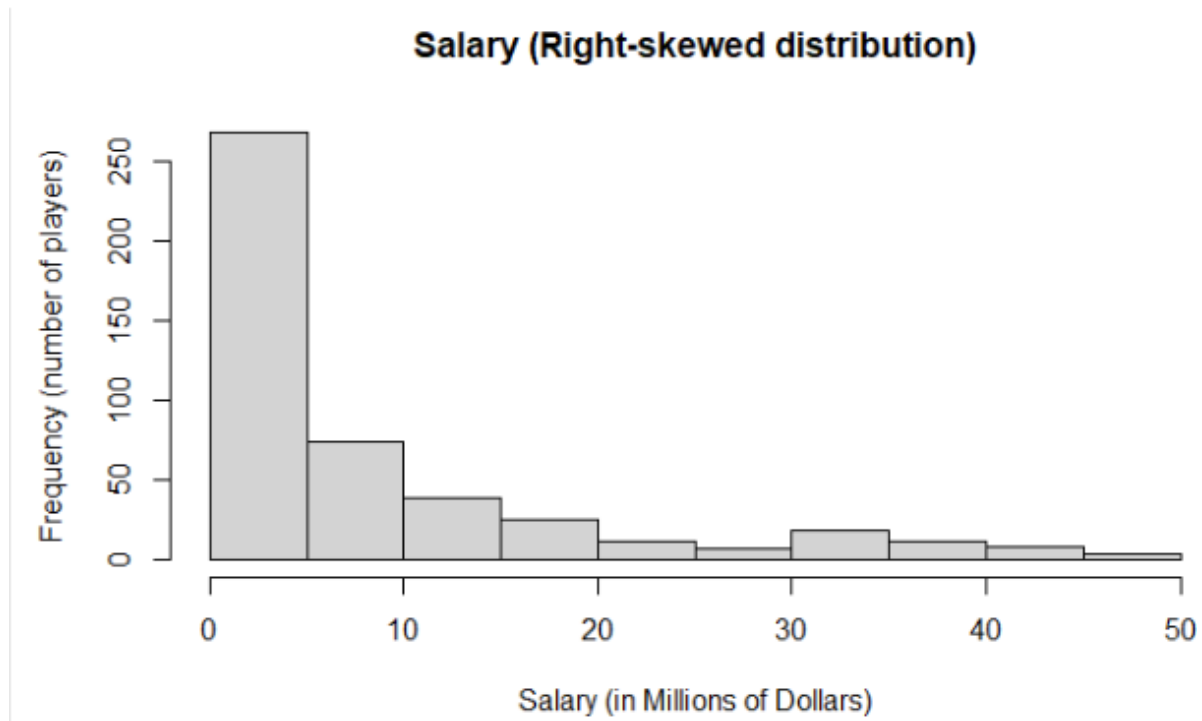
Our dataset was based on the 2022-2023 NBA season, It contained 42 NA values. Instead of removing them, we replaced them with the median values of their respective columns. After this step, we next removed some player stats that were directly influenced by other stats. For example, we had a predictor that had 3 pointers attempted by a player, 3 pointers made by a player, and the 3 point percentage, which was just the 3 pointers made by a player divided by the 3 pointers attempted. This was also done to eFG%, which was just all the shots they made divided by the shots they took, adjusting the weight of 3 pointers being worth more points. Lastly, there was TRB (total rebounds when a person gets possession of the basketball after a missed shot). For this specific player stat there were two types of rebounds recorded, offensive and defensive, so we decided to use the TRB as it accounted for both types of rebounds. We chose this project because we all love basketball and played it from a very young age. Also it was a really fun activity to do to pass the time. This project is interesting because it gives insight into what factors really determine a player's worth, and you could apply this to many things, not just sports.

IV. Data Description

The first thing we wanted to do was clean up our data and make sure no empty values(NA) or inconsistencies were in our data. Our data focused specifically on NBA data from the 2022-2023 season. When we got deep into the data we discovered that there were 42 missing values, but like I mentioned before, instead of replacing them, we took the median value for each respective column. After this step, we were thinking about what factors would be the most important in predicting a player's salary. At first we thought maybe AST which is assists (when a person passes the ball and the person who gets that ball immediately scores), PTS which is points (when a person scores a point), TRB (total rebounds when a person grabs the ball from a missed shot), X3P which is 3 point percentage (a shot shot from behind the half circle line since it's a farther, therefore a harder shot, it counts as 3 points instead of 2, so this

takes how many 3 pointers they made and divide it by the total attempts), FG which is field goal percentage(the percentage of shots(overall) a percent made vs the amount they missed), and lastly STL which is steals(when a person steals or takes the ball away from another player). A lot of the predictors we had were very skewed, so some transformations on the stats were needed.

Lastly, we removed 13 players from our dataset for very specific reasons since they were affecting our model's prediction accuracy. The first 3 players we removed were Shaquille Harrison(\$12,260), Skylar Mays(\$116,574), and Stanely Umude(\$58,493). These players only played for 10 days or less and made a very small salary since the average salary at the time in the NBA was 1 million dollars. Next we had Mac McClung(\$160,856), Gabe York(\$32,171), Justin Minaya(\$35,096), Jay Scybb(\$49,719), Jay Huff(\$116,986), and Lindell Wigginton(\$99,438). These players also had a very low salary and they were on 2 way contracts, so they were splitting their time playing in the NBA and the G-League, which is like the league right before the NBA. We also had two players with Exhibit 10 contracts Jacob Gilyard(\$5,849) and RaiQuan Gray(\$5,849), which are almost the same thing as two-way contracts and they made a tiny salary. Lastly, we had Jeenathan Williams(\$52,644) and Kobi Simmons(\$32,795), who had a standard contract but still made a very low salary. These players did not accurately represent our models in the context of the data, and our model improved after removing them.



V. Questions of Interest

Our project focused on two main research questions: whether we can predict the current rookie of the year salary, and if we can predict the current MVP salary. Out of further interest, we want to predict the salary for the previous years' MVPs and rookies of the year.

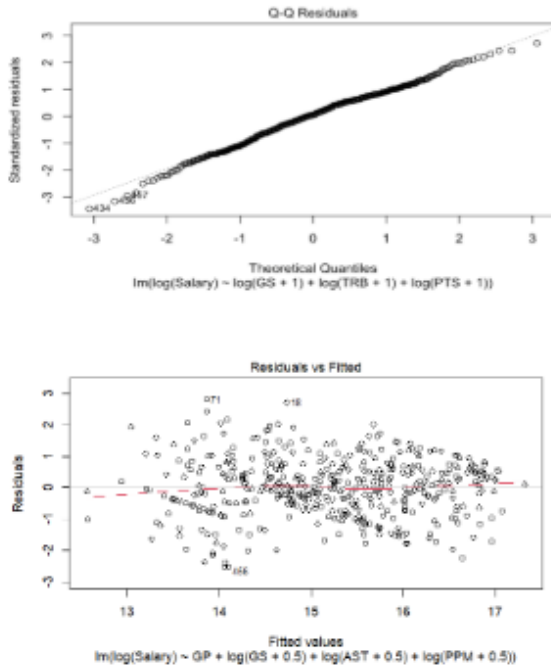
VI. Regression Analysis, Results and Interpretation

Since we are trying to predict NBA player salaries and the salary distribution is right-skewed, we ended up performing a log transformation on the response from the beginning to address this issue. We also had multicollinearity issues with some highly correlated predictors; we ended up combining certain player stats or removing some player stats that were used to calculate other player stats. We also used a corplot to help visualize which predictors are highly correlated with each other and to further investigate those specific predictors. In the rookie of the year model, we combined total points scored and minutes played to get the points per minute scored by a player, which we found to be a better predictor in our model for rookie of the year salary. However, in the MVP model, we found that combining the number of games played and the number of minutes played gives the total minutes played by a player. After removing predictors that don't make sense to include in the model based on our research question and addressing multicollinearity. After addressing all multicollinearity issues all the VIF values we had for our predictors were under 5, whereas some of the VIF values were significantly above 10 before addressing these issues. We next utilized the forward and backwards stepwise algorithm to help finalize a model using both the AIC and BIC criteria. For the rookie of the year model, our anova test had a p-value of 0.0359. With a H_0 : the reduced model is a better fit and H_1 : the full model is a better fit; since the p_value is < 0.05 , we reject H_0 , that is we conclude the full model is a better fit for our model and move forward with the BIC stepwise model. For the MVP model we also used the forward and backward stepwise algorithm with AIC and BIC criterions. Since the AIC value for the reduced model is 1479.202 and the AIC or the full model is 1477.755, we decided to use the full model to predict MVP salary.

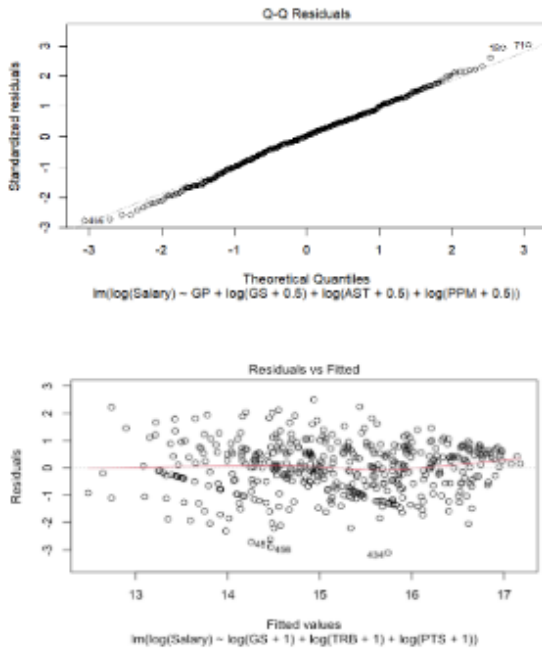
Another step we took was to further investigate bad leverage points. After looking into these points we found 13 NBA players that we decided to remove from our dataset due to various issues, like playing for 10 days or less, or players with 2 way contracts. After removing those specific players we then looked into possible transformations on the predictors as we did notice some trend and a possible funneling pattern on the residuals vs fitted values plot. Next we utilized the powerTransform function on the rookie of the year model to find the optimal transformations for the predictors in our model including: GP, GS, AST, and PPM. We ended up needing to shift the values on the predictors that included any non-positive values. After the transformation we end up with $\log(\text{Salary}) \sim \text{GP} + \log(\text{GS} + .05) + \log(\text{AST} + .05) + \log(\text{PPM} + .05)$. Note some of the rounded powers for this model included 0.08 and -0.13;

however, the respective lower and upper bounds were less than 0.04 away from 0. After taking these steps, the QQ-plot shows that the points approximately follow the line; indicating the normality assumption is met in our model. Furthermore, the residuals vs fitted values plot has no obvious trend or patterns, and the residuals have a constant range across the fitted values. That is to say, the linearity and equal variances assumptions are met in our model. We also utilized the powerTransform function on the MVP model including: GS, TRB, and PTS. We shifted the predictors for a similar issue with non-negative numbers in this model also, and thus ended up with the model: $\log(\text{Salary}) \sim \log(\text{GS}+1) + \log(\text{TRB}+1) + \log(\text{PTS}+1)$. After these steps we also noticed the QQ-plot for the MVP model had the standardized residuals vs theoretical quantiles closely follow the line, indicating that the normality assumption is met. Also we observe that from the residuals vs fitted values plot there is no obvious trend or pattern and the residuals maintain a relatively constant range across the fitted values, showing that the linearity and equal variances assumptions are met. Once we got our final MVP and rookie of the year models, we observed that the rookie of the year had an adjusted R2 of 0.541 and the MVP model had an adjusted R2 of 0.544. Furthermore, for the MVP model we found that points was the most significant predictor and for every 1% increase in points scored the predicted salary is expected to increase by about 0.95% when holding all other predictors constant. For the rookie of the year model we found that for every 1% increase in PPM, the predicted salary is expected to increase by 1.19%, holding all other predictors constant.

MVP model plots



Rookie of the Year model plots



VII. Conclusions:

Summary of your findings and any comments you may have about the reliability or generalizability of your analysis.

In the conclusion section of your project, you should summarize your findings from the final model in clear, non-statistical terms. What is the primary message derived from your analysis? Additionally, this section can address any further questions raised, problems encountered, or potential extensions of the analysis. You may also include final remarks and reflections on your project. For instance, do you trust your results? How generalizable are your results, and to what situations do they apply? Feel free to share any other relevant thoughts or comments

In conclusion, our final was decently accurate at predicting the salaries of mvp's and rookies of the year. It was extremely accurate at times, but also performed poorly when certain player stats were either too high or all the stats were extremely balanced. This is due to the fact that player stats like those do not happen very often in real life. Another thing our MVP model did not account for was the bonuses and contract extensions that really good players like MVP's get. As for the rookies of the year the model was very accurate but again could not account for the salary cap which is just a limit put on a rookies salary so that a franchise does not put themselves in financial ruin by signing a rookie for a lot of money who may or may not produce because you never know if that college talent will transfer to the NBA. Overall our models were extremely accurate at times, for example when we checked last year's rookie of the year (2023-2024) his salary was \$12,160,680 and our model predicted \$12,330,470; which was only off by 2%. However, when we checked the MVP, our model predicted \$24,994,091 and his actual salary was \$47,607,350; so we were off by 47%. This was due to many factors like the bonus he received, in addition to, the contract extension and just a really good balance of those factors has never been achieved before.

Sources

NBA Player Salaries.” HoopsHype, www.hoopshype.com/salaries/players/

NBA Player Contracts, Salaries, and Transactions.” Spotrac, www.spotrac.com/nba/

VIII. Appendix

```
library(tidyverse)
```

Warning: package 'purrr' was built under R version 4.4.3

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.0.4
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(car)
```

Loading required package: carData

Attaching package: 'car'

The following object is masked from 'package:dplyr':

recode

The following object is masked from 'package:purrr':

some

Loading the data

```
nba <- read.csv("nba_salaries.csv")
# head(nba)
# str(nba)
# unique(nba$Position)
```

```
#View(nba[which(nba$MP < 5),])
```

```
dim(nba)
```

```
[1] 467 32
```

Removing unnecessary columns

```
nba2 <- nba |> select(-c(X, Position, Team, Player.additional))
```

```
# salary data right skewed, log transform
```

```
head(nba2)
```

	Player.Name	Salary	Age	GP	GS	MP	FG	FGA	FG.	X3P	X3PA	X3P.	X2P		
1	Stephen Curry	48070014	34	56	56	34.7	10.0	20.2	0.493	4.9	11.4	0.427	5.1		
2	John Wall	47345760	32	34	3	22.2	4.1	9.9	0.408	1.0	3.2	0.303	3.1		
3	Russell Westbrook	47080179	34	73	24	29.1	5.9	13.6	0.436	1.2	3.9	0.311	4.7		
4	LeBron James	44474988	38	55	54	35.5	11.1	22.2	0.500	2.2	6.9	0.321	8.9		
5	Kevin Durant	44119845	34	47	47	35.6	10.3	18.3	0.560	2.0	4.9	0.404	8.3		
6	Bradley Beal	43279250	29	50	50	33.5	8.9	17.6	0.506	1.6	4.4	0.365	7.3		
	X2PA	X2P.	eFG.	FT	FTA	FT.	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
1	8.8	0.579	0.614	4.6	5.0	0.915	0.7	5.4	6.1	6.3	0.9	0.4	3.2	2.1	29.4
2	6.7	0.459	0.457	2.3	3.3	0.681	0.4	2.3	2.7	5.2	0.8	0.4	2.4	1.7	11.4
3	9.7	0.487	0.481	2.8	4.3	0.656	1.2	4.6	5.8	7.5	1.0	0.5	3.5	2.2	15.9
4	15.3	0.580	0.549	4.6	5.9	0.768	1.2	7.1	8.3	6.8	0.9	0.6	3.2	1.6	28.9
5	13.4	0.617	0.614	6.5	7.1	0.919	0.4	6.3	6.7	5.0	0.7	1.4	3.3	2.1	29.1
6	13.2	0.552	0.551	3.8	4.6	0.842	0.8	3.1	3.9	5.4	0.9	0.7	2.9	2.1	23.2

```
nba_model2 <- lm(log(Salary) ~. -Player.Name, nba2)
```

Removing X, Position, Team, Player.additional from our model.

```
# str(nba2)
```

```
# model2.full <- lm(Salary ~ ., nba2)
```

```
# model2.0 <- lm(Salary ~ 1, nba2)
```

```
# n <- nrow(nba2)
```

```
#
```

```
# # step(model2.full, direction = "backward", k = log(n), trace = 0)
# # step(model2.0, scope = list(lower = model2.0, upper = model2.full), direction = "forward"
#
# summary(model2.full)
```

Plotting data:

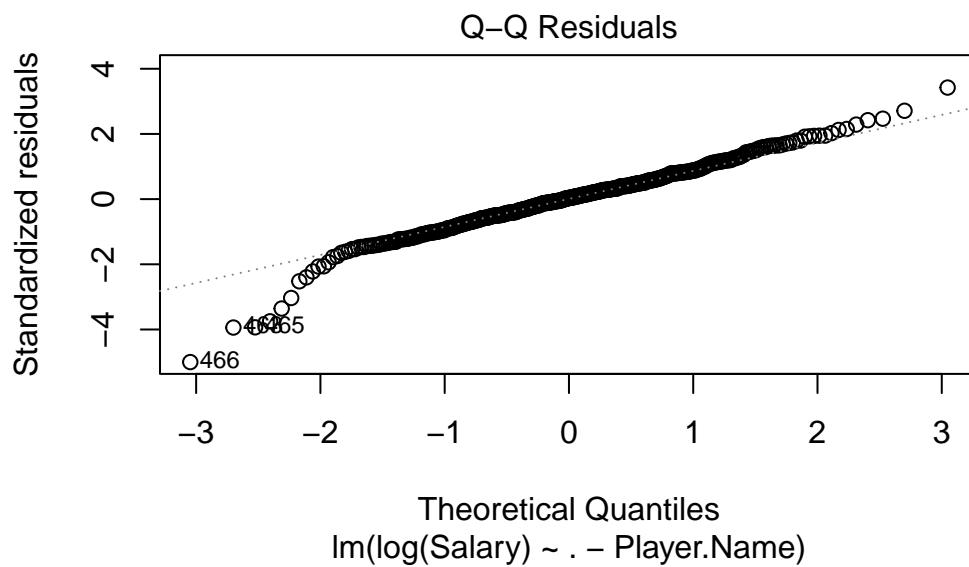
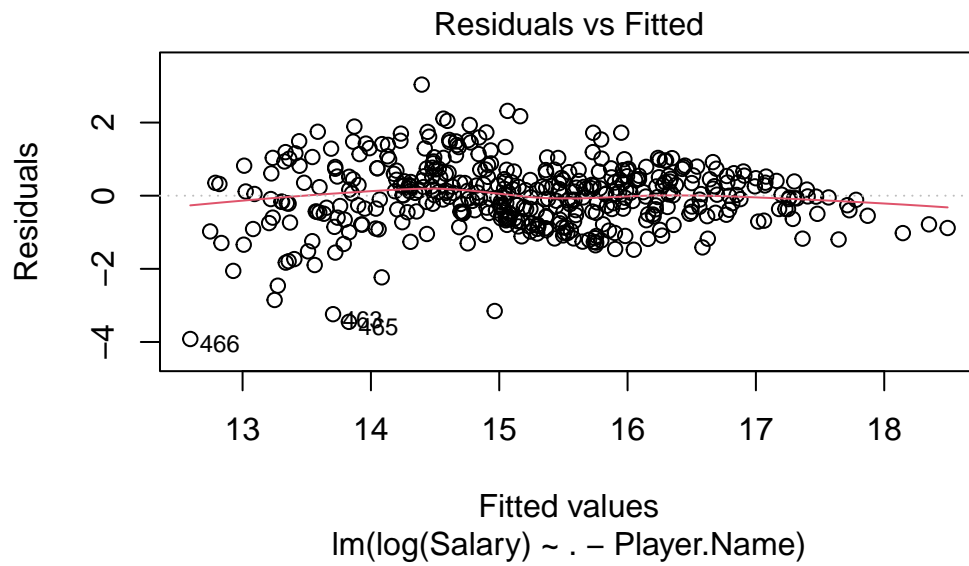
```
# nba_model2 <- lm(Salary ~ ., nba2)

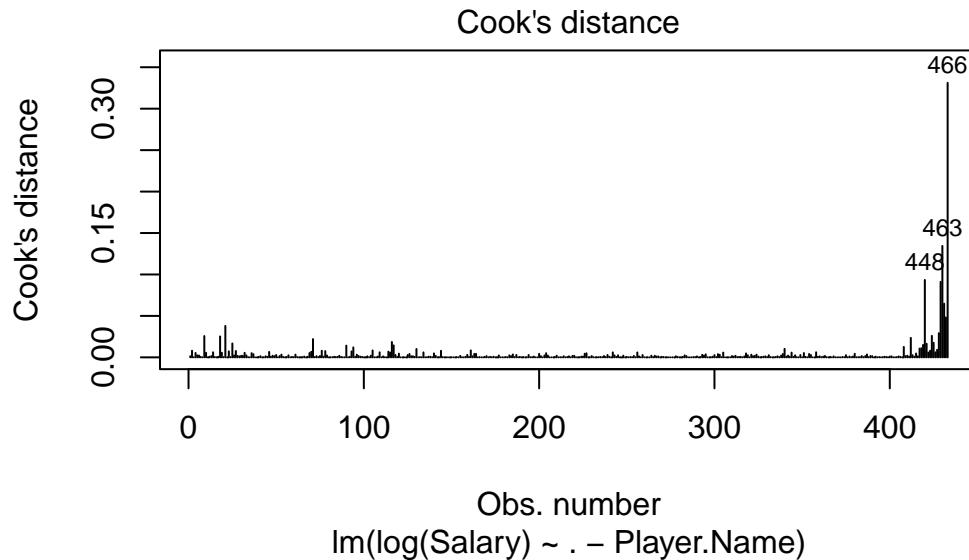
# pairs(Salary ~ ., nba2)
# scatterplotMatrix(~ Salary + Age + GP + GS + MP + FG + FGA + X3P + X3PA + X2P + X2PA + FT -
# avPlots(nba_model2)

# round(cor(nba2, use = "pairwise.complete.obs"), 3)
# corrplot::corrplot(cor(nba2, use = "pairwise.complete.obs"))
```

Assumptions checking

```
plot(nba_model2, c(1,2,4))
```





```
vif_val <- vif(nba_model2)
vif_val
```

Age	GP	GS	MP	FG	FGA
1.146404	1.960957	4.072049	17.246312	6747.834968	11600.101435
FG.	X3P	X3PA	X3P.	X2P	X2PA
20.184407	598.664362	2376.423868	2.032373	1991.882496	6077.558330
X2P.	eFG.	FT	FTA	FT.	ORB
3.231736	17.617354	542.061354	93.448157	1.793573	187.326162
DRB	TRB	AST	STL	BLK	TOV
1017.289543	1772.433420	5.225552	2.398148	2.114279	8.282015
PF	PTS				
3.148362	9695.975359				

Remove shots made and shots attempted columns

Shots made and shots attempted columns result in the percentage of shots made column ($\text{pred} / \text{predA} = \text{pred.}$)

FG. = FG / FGA, same with X3P., X2P., and FT. So we remove the correlated values and keep the percentage column.

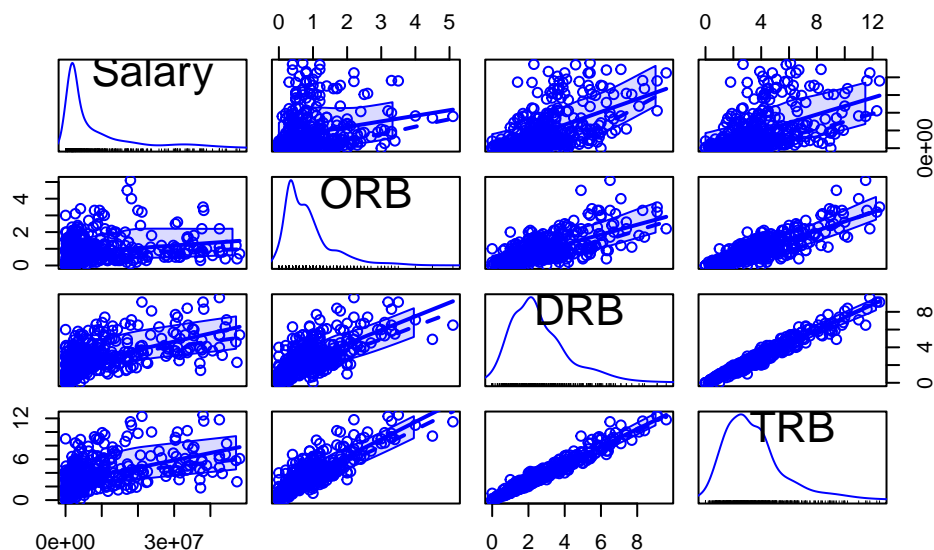
```
nba3 <- nba2 |> select(-c(FG, FGA, X3P, X3PA, X2P, X2PA, FT, FTA,))
# head(nba3)

nba_model3 <- lm(log(Salary) ~ . -Player.Name, nba3)
# vif(nba_model3)
```

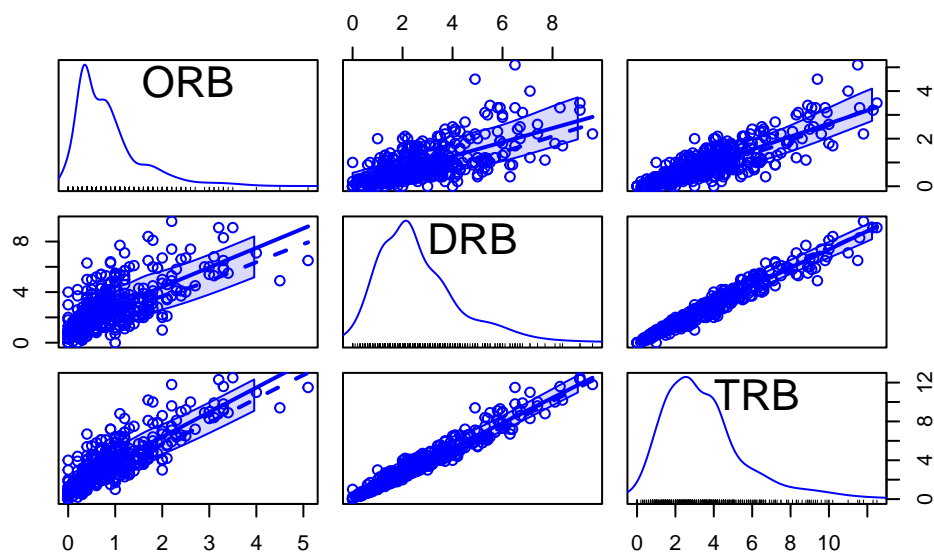
visualizing plots for rebounds (correlated columns)

Since ORB and DRB are highly correlated to TRB, we select TRB as a predictor and remove ORB and DRB. FG. is the field goal percentage; however eFG. is a similar measurement for FG. but adjusts scores depending on if it is 2 or 3 points. We choose to select

```
scatterplotMatrix(~ Salary + ORB + DRB + TRB, nba3)
```



```
# scatterplotMatrix(~Salary + PTS, nba3)
scatterplotMatrix(~ ORB + DRB + TRB, nba3)
```



$$\text{ORB} + \text{DRB} = \text{TRB}$$

```
nba4 <- nba3 |> select(-c(ORB, DRB, FG.))
nba_model4 <- lm(Salary ~. -Player.Name, nba4)
# head(nba4)
```

Multicollinearity

```
vif(nba_model4)
```

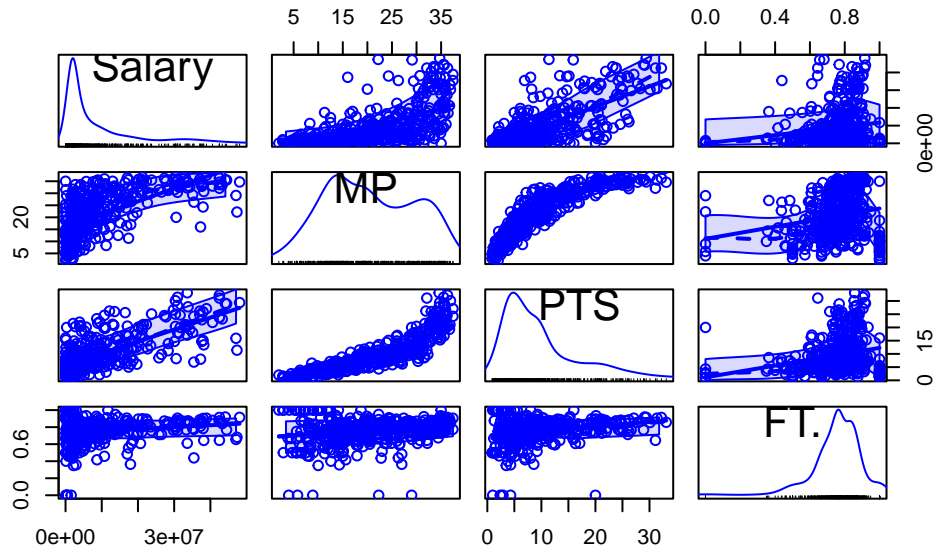
Age	GP	GS	MP	X3P.	X2P.	eFG.	FT.
1.090539	1.871292	3.974785	13.420690	1.615515	2.127983	2.620344	1.236538
TRB	AST	STL	BLK	TOV	PF	PTS	
3.594226	5.030234	2.337795	1.927315	7.661694	3.031019	8.263502	

Variables with high VIF (multicollinearity):

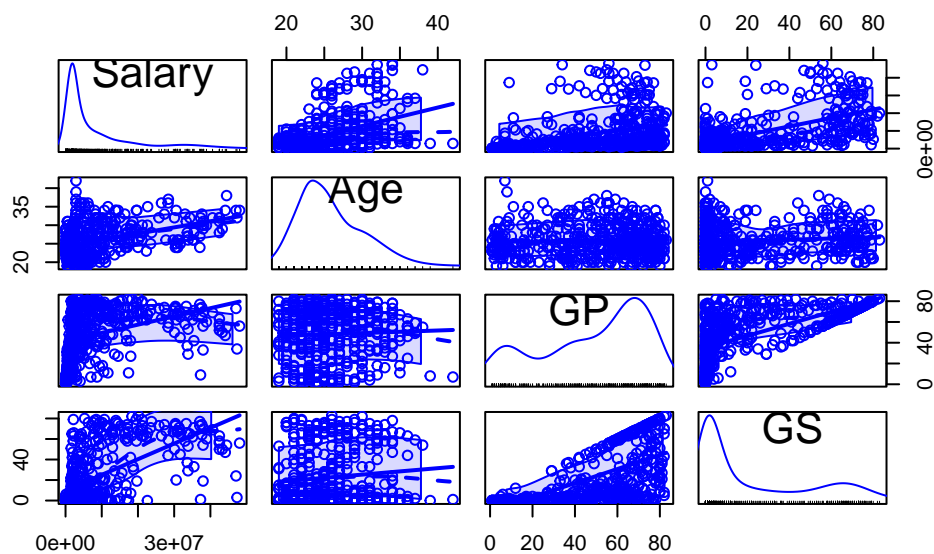
> 10: MP,

> 5: AST, TOV, PTS

```
# MP and PTS
scatterplotMatrix(~Salary + MP + PTS + FT., nba4)
```

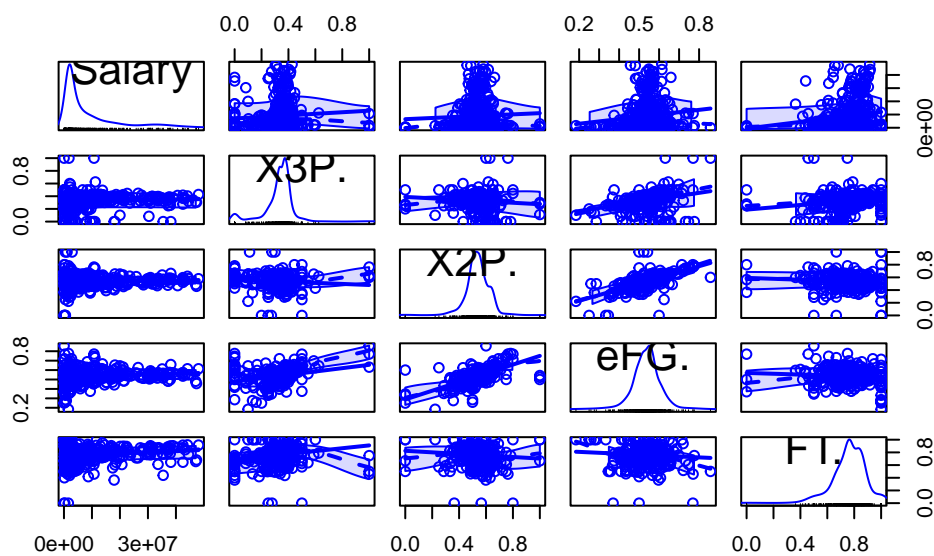


```
scatterplotMatrix(~ Salary + Age + GP + GS, nba4)
```

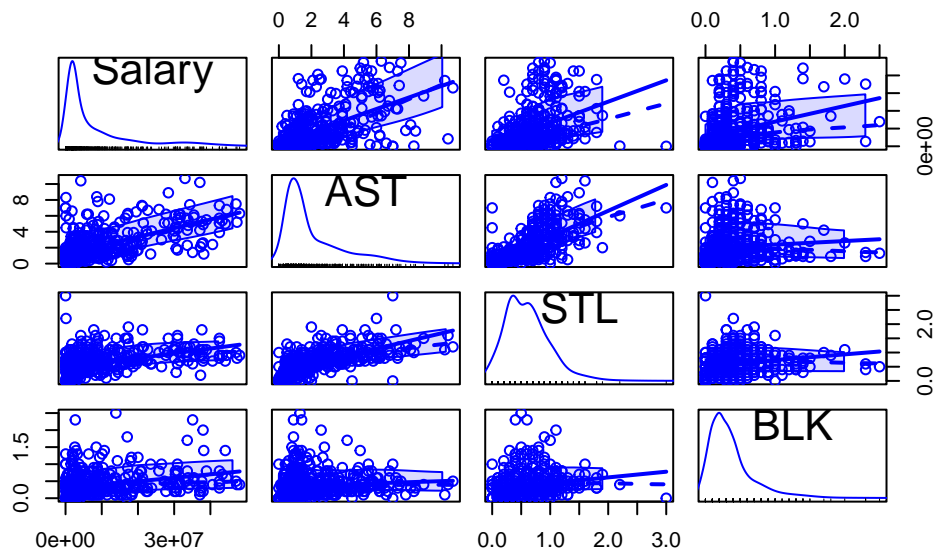


```
# scatterplotMatrix(~ Salary + MP + X3P., nba4)
```

```
scatterplotMatrix(~ Salary + X3P. + X2P. + eFG. + FT., nba4)
```



```
scatterplotMatrix(~ Salary + AST + STL + BLK, nba4)
```



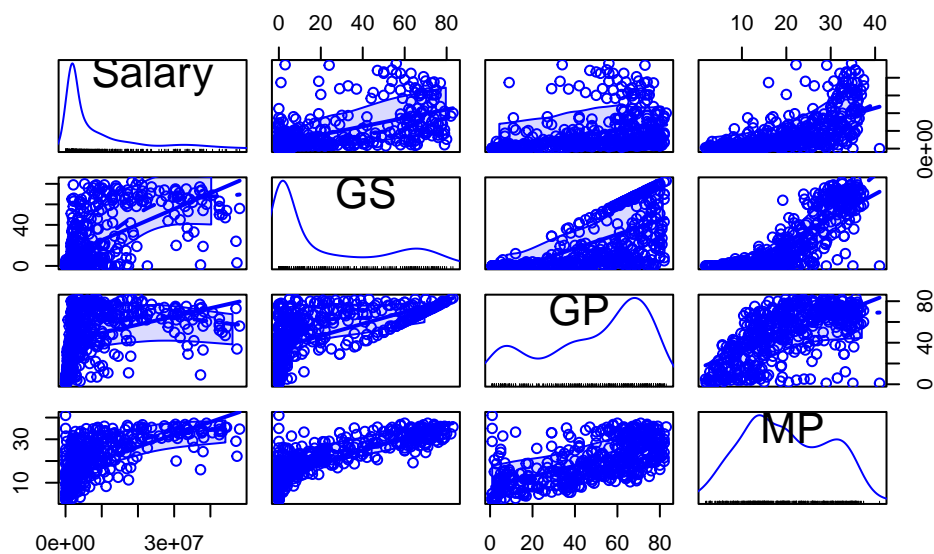
```
# scatterplotMatrix(~ Salary + TOV + PF + PTS, nba4)
```

```
# highly correlated
```

```
# scatterplotMatrix(~ Salary + X3P. + X2P. + eFG. + FT. + PTS, nba4)
```

```
# scatterplotMatrix(~Salary + PF, nba4)
```

```
scatterplotMatrix(~ Salary + GS + GP + MP, nba4)
```



Replacing N/A values with median value

(Note: only accuracy/percentage columns have NA which represents no shots were attempted)

```
# NAs are in percentage is because the values are 0 shots made / 0 shots attempted.

df <- data.frame(NAs = colSums(is.na(nba4)))
df %>% filter(NAs > 0)
```

```
      NAs
X3P.   13
X2P.    4
eFG.    1
FT.   23
```

```
sum(is.na(nba4))
```

```
[1] 41
```

```
dim(nba4)
```

```
[1] 467 17
```

```
# Get rows with at least one NA
na_rows <- nba4[rowSums(is.na(nba4)) > 0, ]
# na_rows
```

```
# replace NA with median value of columnn
nba5_med <- nba4
nba5_med$X3P.[is.na(nba5_med$X3P.)] <- median(nba5_med$X3P., na.rm = TRUE)
nba5_med$X2P.[is.na(nba5_med$X2P.)] <- median(nba5_med$X2P., na.rm = TRUE)
nba5_med$eFG.[is.na(nba5_med$eFG.)] <- median(nba5_med$eFG., na.rm = TRUE)
nba5_med$FT.[is.na(nba5_med$FT.)] <- median(nba5_med$FT., na.rm = TRUE)

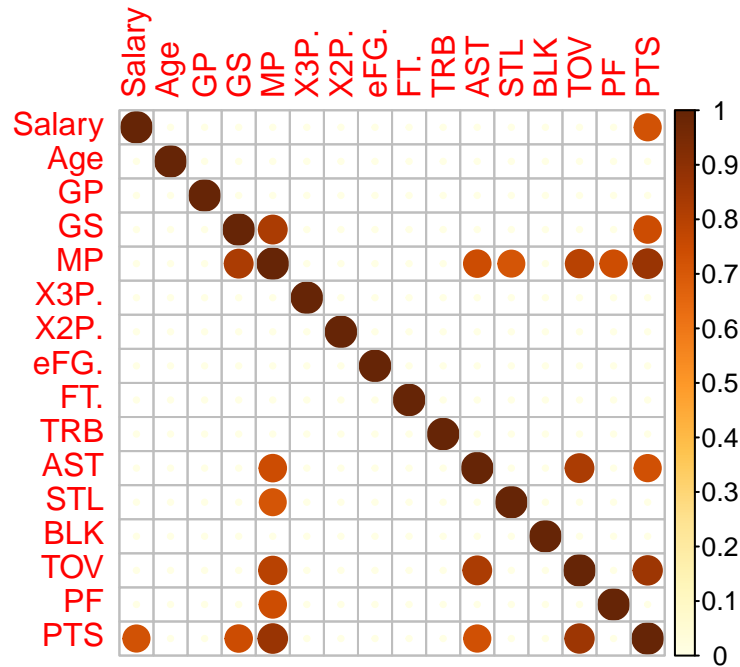
sum(is.na(nba5_med))
```

```
[1] 0
```

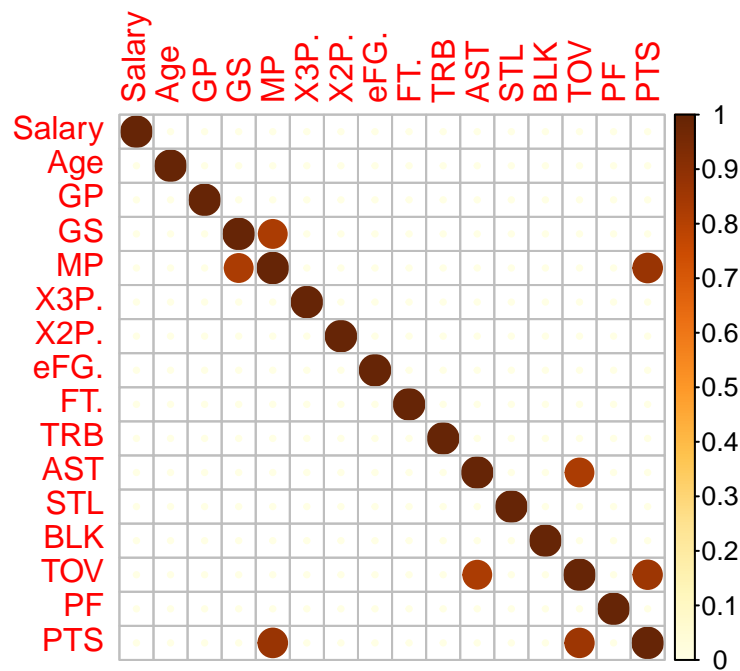
```
# dim(nba5_med)
# head(nba4)
# head(nba5_med)
```

Correlation plots

```
# correlations over 0.7
corrplot::corrplot(cor(nba5_med[, -1])*(abs(cor(nba5_med[, -1])) > 0.7), is.corr = FALSE)
```



```
# correlations over 0.8
corrplot::corrplot(cor(nba5_med[, -1])*(abs(cor(nba5_med[, -1])) > 0.8), is.corr = FALSE)
```



Further Variable Selection

X3p. and X2p. is accounted for by eFG.

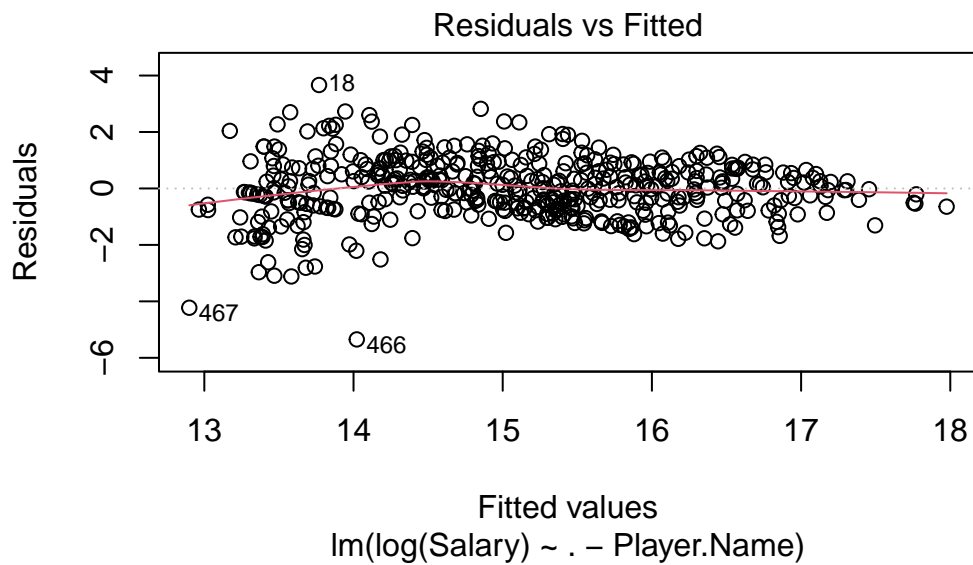
PF - personal fouls don't affect salary

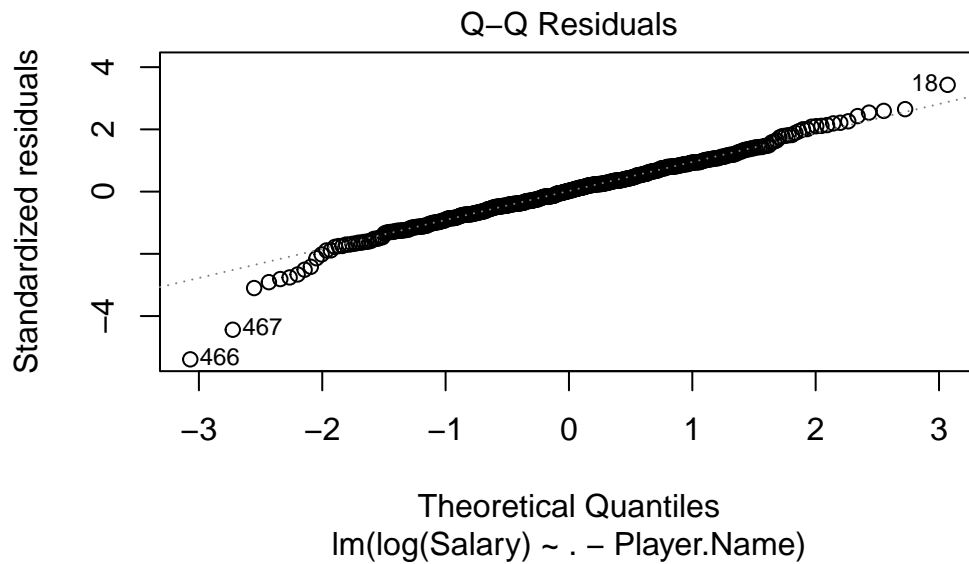
age- doesn't indicate salary

Removing turnover (Luka) - similar reason as PF, doesn't affect salary as long as not excessive, but other player statistics

```
nba6 <- nba5_med |> select(-c(X3P., X2P., PF, Age, TOV))  
# head(nba6)  
  
nba_model6 <- lm(log(Salary) ~ . -Player.Name, nba6)  
# dim(nba6)
```

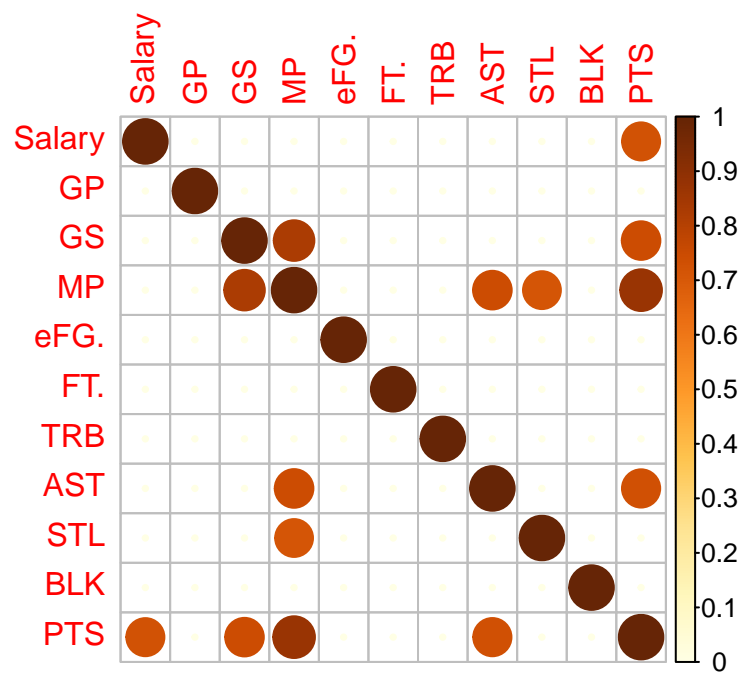
```
plot(nba_model6, 1:2)
```





correlations over 0.8

```
corrplot::corrplot(cor(nba6[, -1])*(abs(cor(nba6[, -1])) > 0.7), is.corr = FALSE)
```



```
vif(nba_model6)
```

	GP	GS	MP	eFG.	FT.	TRB	AST	STL
	2.016655	3.492714	10.690572	1.155351	1.189095	3.225686	3.149864	2.432912
	BLK	PTS						
	1.925231	5.134776						

Rookie Model

Points/Minute

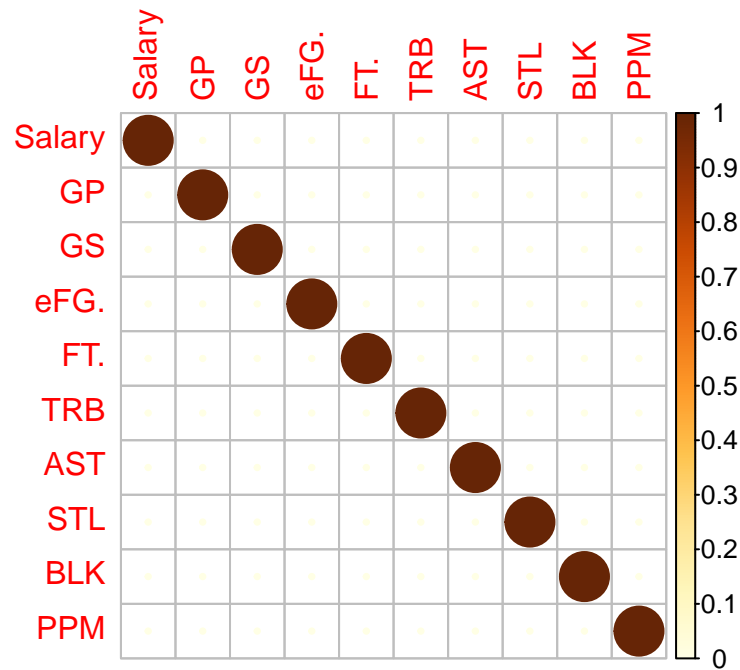
```
# points per minute (pts / mp) - # of points div minutes played
nba_ppm <- nba6
# head(nba_ppm)
# dim(nba7)
# dim(nba_ppm)
nba_ppm$PPM <- (nba6$PTS / nba6$MP)
nba_ppm <- nba_ppm |> select(-c(MP, PTS))
head(nba_ppm)
```

	Player.Name	Salary	GP	GS	eFG.	FT.	TRB	AST	STL	BLK	PPM
1	Stephen Curry	48070014	56	56	0.614	0.915	6.1	6.3	0.9	0.4	0.8472622
2	John Wall	47345760	34	3	0.457	0.681	2.7	5.2	0.8	0.4	0.5135135
3	Russell Westbrook	47080179	73	24	0.481	0.656	5.8	7.5	1.0	0.5	0.5463918
4	LeBron James	44474988	55	54	0.549	0.768	8.3	6.8	0.9	0.6	0.8140845
5	Kevin Durant	44119845	47	47	0.614	0.919	6.7	5.0	0.7	1.4	0.8174157
6	Bradley Beal	43279250	50	50	0.551	0.842	3.9	5.4	0.9	0.7	0.6925373

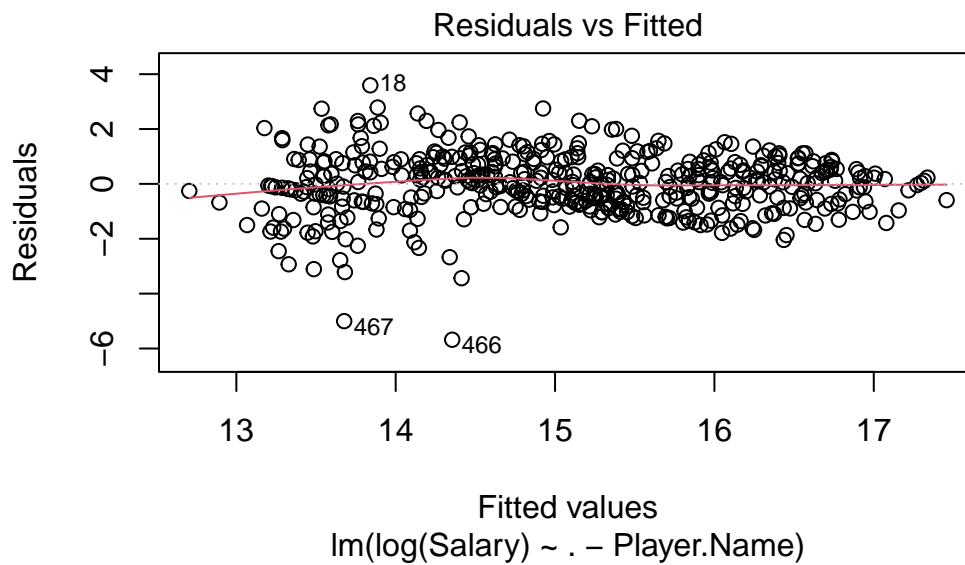
```
nba_ppm.mod <- lm(log(Salary) ~ . -Player.Name, nba_ppm)
vif(nba_ppm.mod)
```

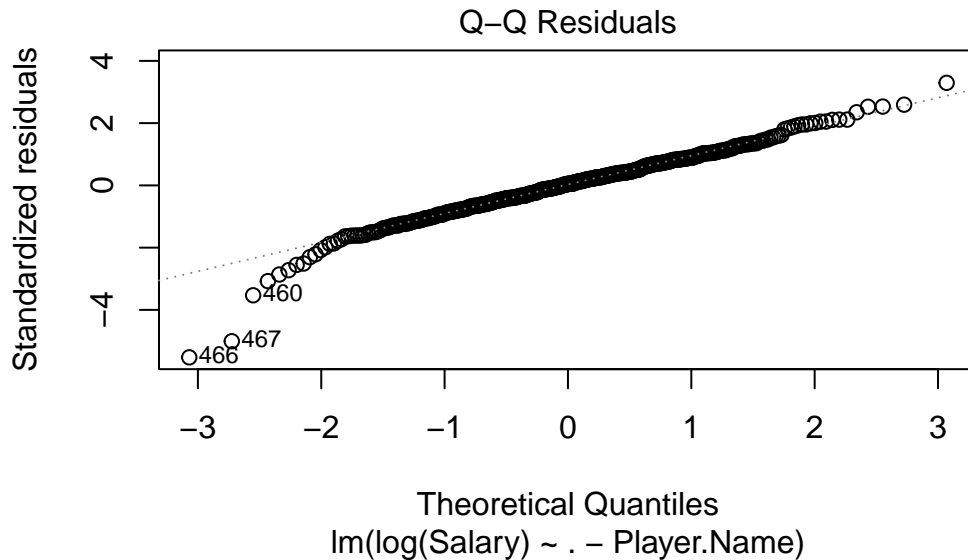
	GP	GS	eFG.	FT.	TRB	AST	STL	BLK
	1.775753	2.828684	1.260029	1.190533	2.754505	2.446919	2.020937	1.923260
	PPM							
	1.562822							

```
corrplot::corrplot(cor(nba_ppm[, -1])*(abs(cor(nba_ppm[, -1])) > 0.7), is.corr = FALSE)
```



```
plot(nba_ppm.mod, 1:2)
```





stepwise variable selection

```
nba_ppm.mod <- lm(log(Salary) ~ . -Player.Name, nba_ppm)
nba_ppm.mod.0 <- lm(log(Salary) ~ 1, nba_ppm)
n <- nrow(nba_ppm)

mod2 <- step(nba_ppm.mod.0, scope = list(lower = nba_ppm.mod.0, upper = nba_ppm.mod), trace = TRUE)
mod1 <- step(nba_ppm.mod.0, scope = list(lower = nba_ppm.mod.0, upper = nba_ppm.mod), k = log(n))

# mod1
# mod2
anova(mod1, mod2)
```

Analysis of Variance Table

```
Model 1: log(Salary) ~ GP + GS + PPM + AST
Model 2: log(Salary) ~ GP + GS + PPM + AST + TRB
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	462	557.98				
2	461	552.67	1	5.3059	4.4258	0.03594 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

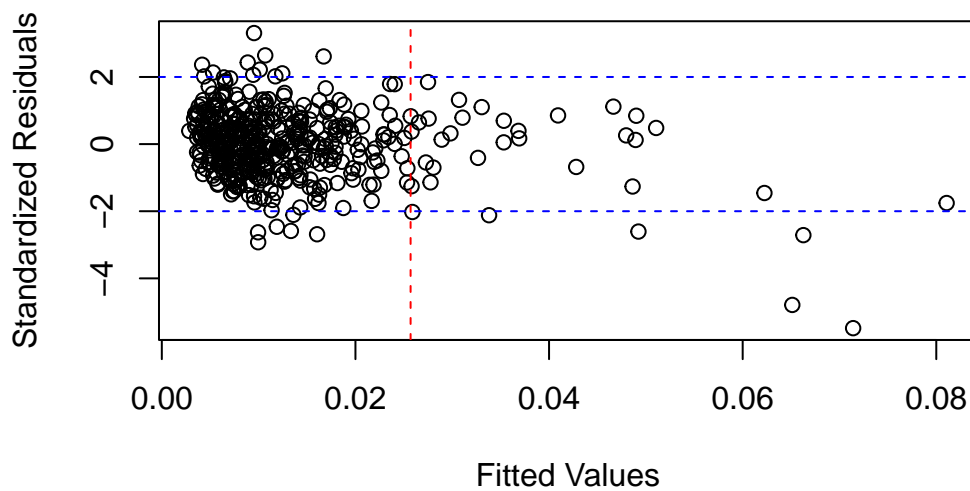
```
AIC(mod1, mod2)
```

```
      df      AIC
mod1  6 1420.409
mod2  7 1417.947
```

```
# summary(mod1)
# summary(mod2)
```

```
plot(hatvalues(mod2), rstandard(mod2), xlab = "Fitted Values", ylab = "Standardized Residuals",
      abline(h = c(2,-2), lty = 2, col = "blue"))

p <- 5
n <- nrow(nba_ppm)
abline(v = 2 * (p + 1) / n, col = "red", lty = 2)
```



outliers

```
outlier_ind <- which(abs(rstandard(mod2)) > 2)
nba_ppm[outlier_ind, ]
```

	Player.Name	Salary	GP	GS	eFG.	FT.	TRB	AST	STL	BLK	PPM
2	John Wall	47345760	34	3	0.457	0.681	2.7	5.2	0.8	0.4	0.51351351
14	Khris Middleton	37984276	33	19	0.499	0.902	4.2	4.9	0.7	0.2	0.62139918
18	Kemba Walker	37281261	9	1	0.482	0.810	1.8	2.1	0.2	0.2	0.50000000
69	Evan Fournier	18000000	27	7	0.443	0.857	1.8	1.3	0.6	0.1	0.35882353
71	Jonathan Isaac	17400000	11	0	0.472	0.556	4.0	0.5	1.3	0.4	0.44247788
77	Duncan Robinson	16902000	42	1	0.504	0.906	1.6	1.1	0.3	0.0	0.38787879
87	Derrick Rose	14520730	27	0	0.443	0.917	1.5	1.7	0.3	0.2	0.44800000
106	Danny Green	11710528	11	0	0.612	1.000	1.3	0.5	0.5	0.3	0.44000000
131	Nerlens Noel	9391069	17	4	0.375	0.667	2.7	0.6	0.9	0.6	0.18260870
142	Daniel Theis	8694369	7	1	0.500	0.417	3.1	1.3	0.3	0.9	0.44871795
163	Cody Martin	7000000	7	0	0.431	0.571	3.4	1.6	0.6	0.1	0.26178010
446	Mac McClung	160856	2	0	0.550	0.600	5.0	4.5	0.0	0.0	0.60975610
448	Shaquille Harrison	134862	5	0	0.458	0.733	4.4	6.0	2.2	0.4	0.36666667
453	Skyлар Mays	116574	6	6	0.588	0.923	3.2	8.3	1.0	0.2	0.48571429
458	Lindell Wigginton	99438	7	1	0.568	0.889	1.0	2.0	0.0	0.3	0.57258065
460	Stanley Umude	58493	1	0	0.000	1.000	0.0	0.0	1.0	1.0	1.00000000
461	Jeenathan Williams	52644	5	4	0.654	0.667	3.0	2.0	0.6	0.4	0.41732283
462	Jay Scrubb	49719	2	0	0.857	0.500	3.0	0.5	1.0	0.0	0.43333333
463	Justin Minaya	35096	4	0	0.370	0.000	3.8	1.0	0.5	1.3	0.19282511
464	Kobi Simmons	32795	5	0	0.250	1.000	0.8	1.0	0.0	0.4	0.17857143
465	Gabe York	32171	3	0	0.524	1.000	2.0	1.7	0.7	0.0	0.42780749
466	RaiQuan Gray	5849	1	0	0.583	1.000	9.0	7.0	0.0	1.0	0.45714286
467	Jacob Gilyard	5849	1	0	0.500	0.769	4.0	7.0	3.0	0.0	0.07317073

high lev pts

```
# points with high leverage
ind_lev <- which(hatvalues(mod2) > 2*(p+1)/n)
nba_ppm[ind_lev, ]
```

	Player.Name	Salary	GP	GS	eFG.	FT.	TRB	AST	STL	BLK
3	Russell Westbrook	47080179	73	24	0.481	0.656	5.8	7.5	1.0	0.5
4	LeBron James	44474988	55	54	0.549	0.768	8.3	6.8	0.9	0.6

9	Giannis Antetokounmpo	42492492	63	63	0.572	0.645	11.8	5.7	0.8	0.8
10	Damian Lillard	42492492	58	58	0.564	0.914	4.8	7.3	0.9	0.3
13	Rudy Gobert	38172414	70	70	0.659	0.644	11.6	1.2	0.8	1.4
15	Anthony Davis	37980720	56	54	0.573	0.784	12.5	2.6	1.1	2.0
19	Trae Young	37096500	73	73	0.485	0.886	3.0	10.2	1.1	0.1
21	Ben Simmons	35448672	42	33	0.566	0.439	6.3	6.1	1.3	0.6
28	Joel Embiid	33616770	66	66	0.573	0.857	10.2	4.2	1.0	1.7
30	James Harden	33000000	58	58	0.536	0.867	6.1	10.7	1.2	0.5
36	Deandre Ayton	30913750	67	67	0.592	0.760	10.0	1.7	0.6	0.8
43	Chris Paul	28400000	59	59	0.513	0.831	4.3	8.9	1.5	0.4
48	Draymond Green	25806468	73	73	0.570	0.713	7.2	6.8	1.0	0.8
57	Domantas Sabonis	21100000	79	79	0.632	0.742	12.3	7.3	0.8	0.5
61	Jarrett Allen	20000000	68	68	0.645	0.733	9.8	1.7	0.8	1.2
68	Clint Capela	18206896	65	63	0.653	0.603	11.0	0.9	0.7	1.2
70	Steven Adams	17926829	42	42	0.597	0.364	11.5	2.3	0.9	1.1
76	Mitchell Robinson	17045454	59	58	0.671	0.484	9.4	0.9	0.9	1.8
105	Ja Morant	12119440	61	59	0.504	0.748	5.9	8.1	1.1	0.3
114	Bobby Portis	10843350	70	22	0.555	0.768	9.6	1.5	0.4	0.2
118	Cade Cunningham	10552800	12	12	0.453	0.837	6.2	6.0	0.8	0.6
122	Ivica Zubac	10123457	76	76	0.634	0.697	9.9	1.0	0.4	1.3
138	Darius Garland	8920794	69	69	0.537	0.863	2.7	7.8	1.2	0.1
143	LaMelo Ball	8623920	36	36	0.510	0.836	6.4	8.4	1.3	0.3
217	Tyrese Haliburton	4215120	56	56	0.586	0.871	3.7	10.4	1.6	0.4
438	Donovan Williams	239822	2	0	0.400	0.769	1.0	0.0	0.0	0.0
440	Chris Silva	211045	1	0	1.000	0.769	0.0	0.0	0.0	0.0
441	Tyler Dorsey	201802	3	0	0.900	0.769	0.7	0.0	0.0	0.0
446	Mac McClung	160856	2	0	0.550	0.600	5.0	4.5	0.0	0.0
448	Shaquille Harrison	134862	5	0	0.458	0.733	4.4	6.0	2.2	0.4
453	Skylar Mays	116574	6	6	0.588	0.923	3.2	8.3	1.0	0.2
454	Frank Jackson	113114	1	0	0.000	0.769	2.0	1.0	0.0	0.0
460	Stanley Umude	58493	1	0	0.000	1.000	0.0	0.0	1.0	1.0
466	RaiQuan Gray	5849	1	0	0.583	1.000	9.0	7.0	0.0	1.0
467	Jacob Gilyard	5849	1	0	0.500	0.769	4.0	7.0	3.0	0.0

PPM

3	0.54639175
4	0.81408451
9	0.96884735
10	0.88705234
13	0.43648208
15	0.76176471
19	0.75287356
21	0.26235741
28	0.95664740

```

30 0.57065217
36 0.59210526
43 0.43437500
48 0.26984127
57 0.55202312
61 0.43865031
68 0.45112782
70 0.31851852
76 0.27407407
105 0.82131661
114 0.54230769
118 0.59759760
122 0.37762238
138 0.60845070
143 0.66193182
217 0.61607143
438 1.00000000
440 0.66666667
441 1.11111111
446 0.60975610
448 0.36666667
453 0.48571429
454 0.00000000
460 1.00000000
466 0.45714286
467 0.07317073

```

bad leverage pts

```

ind <- which(abs(rstandard(mod2)) > 2 & hatvalues(mod2) > 2*(p+1)/n)
leverage_bad <- nba_ppm[ind, ]
leverage_bad

```

	Player.Name	Salary	GP	GS	eFG.	FT.	TRB	AST	STL	BLK	PPM
446	Mac McClung	160856	2	0	0.550	0.600	5.0	4.5	0.0	0.0	0.60975610
448	Shaquille Harrison	134862	5	0	0.458	0.733	4.4	6.0	2.2	0.4	0.36666667
453	Skylar Mays	116574	6	6	0.588	0.923	3.2	8.3	1.0	0.2	0.48571429
460	Stanley Umude	58493	1	0	0.000	1.000	0.0	0.0	1.0	1.0	1.00000000
466	RaiQuan Gray	5849	1	0	0.583	1.000	9.0	7.0	0.0	1.0	0.45714286
467	Jacob Gilyard	5849	1	0	0.500	0.769	4.0	7.0	3.0	0.0	0.07317073

```
leverage_bad$Original_sal <- exp(leverage_bad$Salary)
leverage_bad[, c(1,2,11)]
```

	Player.Name	Salary	PPM
446	Mac McClung	160856	0.60975610
448	Shaquille Harrison	134862	0.36666667
453	Skylar Mays	116574	0.48571429
460	Stanley Umude	58493	1.00000000
466	RaiQuan Gray	5849	0.45714286
467	Jacob Gilyard	5849	0.07317073

```
nba_ppm[446, 1]
```

```
[1] "Mac McClung"
```

```
nba_ppm[448, 1]
```

```
[1] "Shaquille Harrison"
```

```
nba_ppm[453, 1]
```

```
[1] "Skylar Mays"
```

```
nba_ppm[460, 1]
```

```
[1] "Stanley Umude"
```

```
nba_ppm[466, 1]
```

```
[1] "RaiQuan Gray"
```

```
nba_ppm[467, 1]
```

```
[1] "Jacob Gilyard"
```

```
nba_ppm[465, 1]
```

```
[1] "Gabe York"
```

```
nba_ppm[463, 1]
```

```
[1] "Justin Minaya"
```

```
nba_ppm[461, 1]
```

```
[1] "Jeenathan Williams"
```

```
nba_ppm[464, 1]
```

```
[1] "Kobi Simmons"
```

```
nba_ppm[462, 1]
```

```
[1] "Jay Scrubb"
```

```
nba_ppm[452, 1]
```

```
[1] "Jay Huff"
```

```
nba_ppm[458, 1]
```

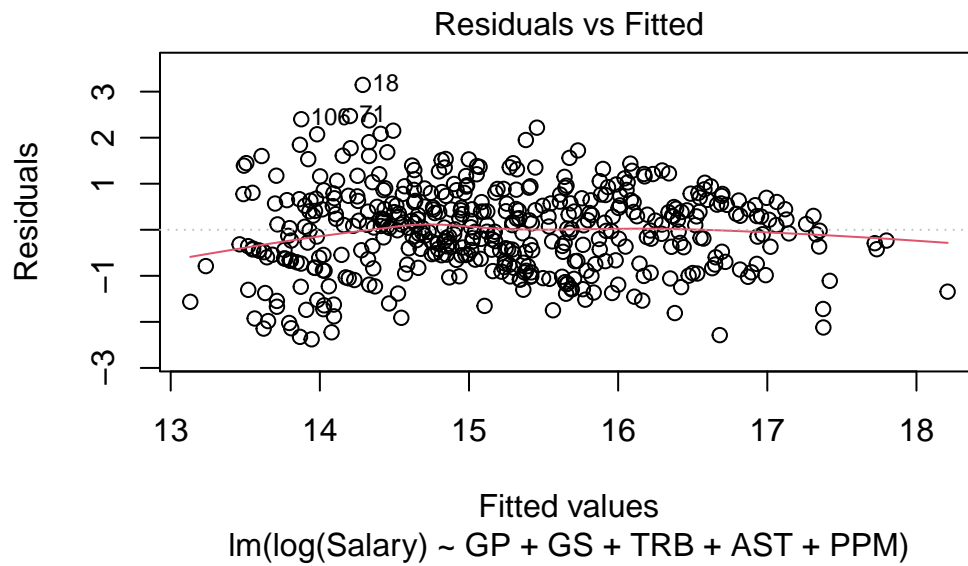
```
[1] "Lindell Wigginton"
```

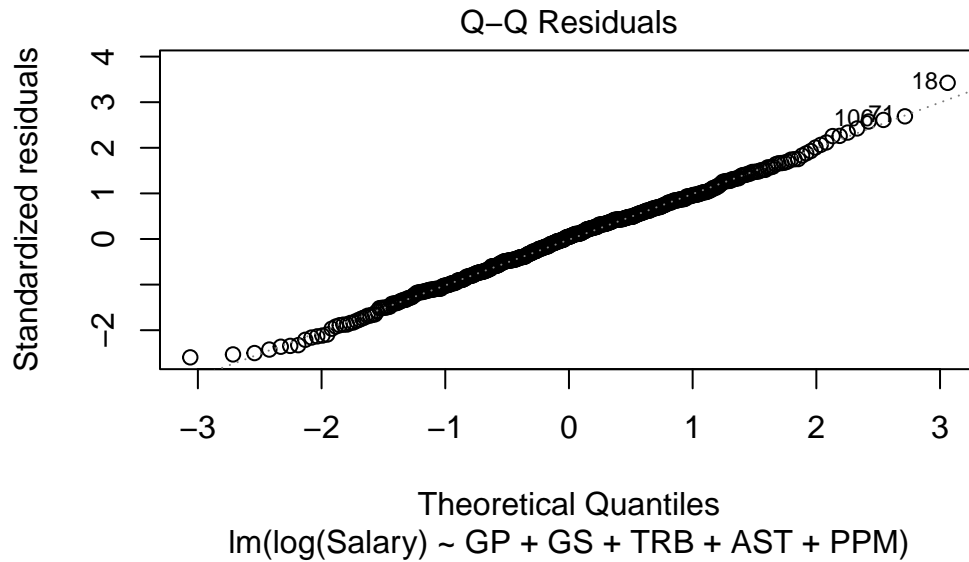
```
# Shaquille Harrison signed a 10 day contract.(5 games)
# justin manaya - 10 day contract outlier
# jay scrubb -> split time between NBA and G league (salary)
# louis king -> split time between NBA and G league (salary)
# Mac McClung, Gabe York - Two way contract G-League and NBA(2 and 3 games, respectively)
```

```
# plot(mod2, 1:2)

rookie <- nba_ppm[-c(446, 448, 453, 460, 466, 467, 465, 463, 461, 464, 462, 452, 458), ]
# rookie <- final_df

rookie.mod <- lm(log(Salary) ~ GP + GS + TRB + AST + PPM, rookie)
plot(rookie.mod, c(1, 2))
```





```
# sum(is.na(test))
summary(rookie.mod)
```

Call:

```
lm(formula = log(Salary) ~ GP + GS + TRB + AST + PPM, data = rookie)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2.37836	-0.63632	0.04609	0.59940	3.14644

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	13.099622	0.153464	85.359	< 2e-16	***
GP	0.015833	0.002352	6.732	5.14e-11	***
GS	0.005565	0.002657	2.095	0.0367	*
TRB	0.120123	0.025224	4.762	2.59e-06	***
AST	0.207183	0.030756	6.736	5.00e-11	***
PPM	0.777148	0.327207	2.375	0.0180	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9238 on 448 degrees of freedom
 Multiple R-squared: 0.5506, Adjusted R-squared: 0.5456
 F-statistic: 109.8 on 5 and 448 DF, p-value: < 2.2e-16

```
# rookie
# plot(rookie.mod, 1:2)
```

```
# scatterplotMatrix(~log(Salary) + GP + GS + TRB + AST + PPM, test)
#
# scatterplotMatrix(~log(Salary) + GS + AST + PPM, test)
```

```
# pft_1 <- powerTransform(cbind(GP, GS+1, AST+1, PPM+1) ~ 1, data = rookie)
# summary(pft_1)

pft_05 <- powerTransform(cbind(GP, GS+.5, AST+.5, PPM+.5) ~ 1, data = rookie)
summary(pft_05)
```

bcPower Transformations to Multinormality

	Est Power	Rounded Pwr	Wald Lwr Bnd	Wald Up Bnd
GP	1.0870	1.00	0.9596	1.2144
	0.0843	0.08	0.0302	0.1384
	-0.1337	-0.13	-0.2468	-0.0205
	-0.0114	0.00	-0.3642	0.3414

Likelihood ratio test that transformation parameters are equal to 0
 (all log transformations)

	LRT	df	pval
LR test, lambda = (0 0 0 0)	435.2481	4	< 2.22e-16

Likelihood ratio test that no transformations are needed

	LRT	df	pval
LR test, lambda = (1 1 1 1)	1141.059	4	< 2.22e-16

```
# pft_025 <- powerTransform(cbind(GP, GS+.25, AST+.25, PPM+.25) ~ 1, data = rookie)
# summary(pft_025)
```

```
rookie_bic05 <- lm(log(Salary) ~ GP + log(GS+.5) + log(AST+.5) + log(PPM+.5), rookie) ##
summary(rookie_bic05)
```

```
Call:
lm(formula = log(Salary) ~ GP + log(GS + 0.5) + log(AST + 0.5) +
    log(PPM + 0.5), data = rookie)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2.53782	-0.58050	0.02469	0.58483	2.80151

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	13.950626	0.119037	117.196	< 2e-16 ***
GP	0.010766	0.002717	3.962	8.64e-05 ***
log(GS + 0.5)	0.185247	0.040579	4.565	6.46e-06 ***
log(AST + 0.5)	0.643450	0.090544	7.106	4.71e-12 ***
log(PPM + 0.5)	1.184514	0.293745	4.032	6.49e-05 ***

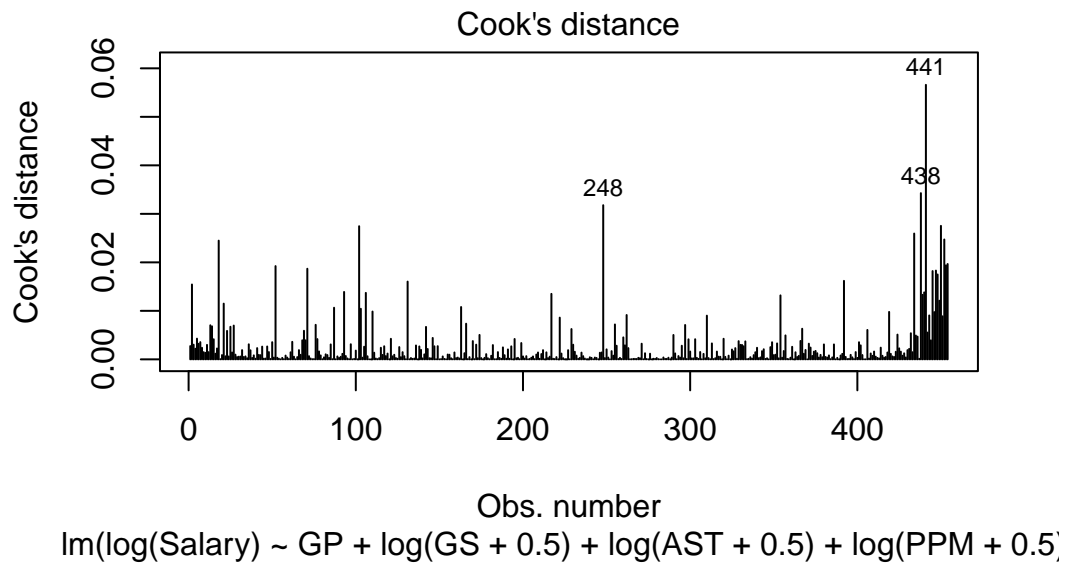
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9287 on 449 degrees of freedom

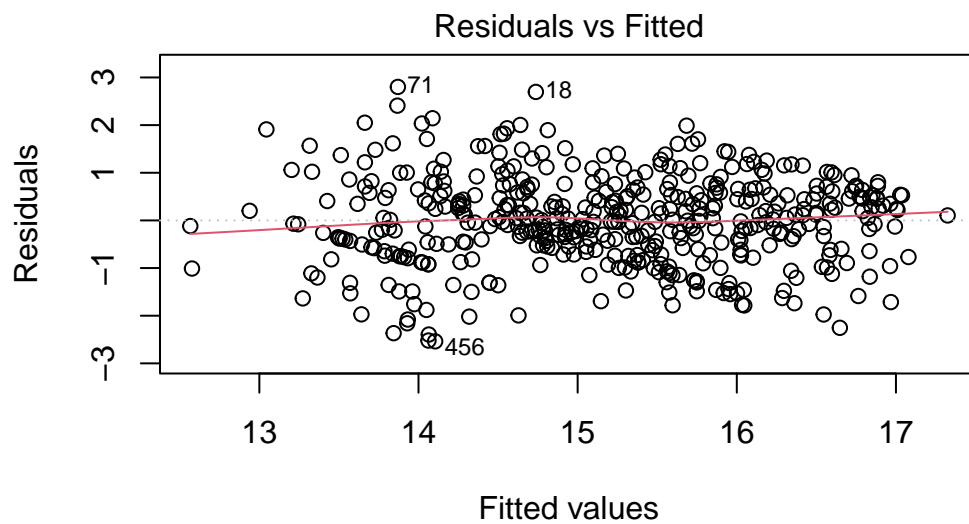
Multiple R-squared: 0.5448, Adjusted R-squared: 0.5408

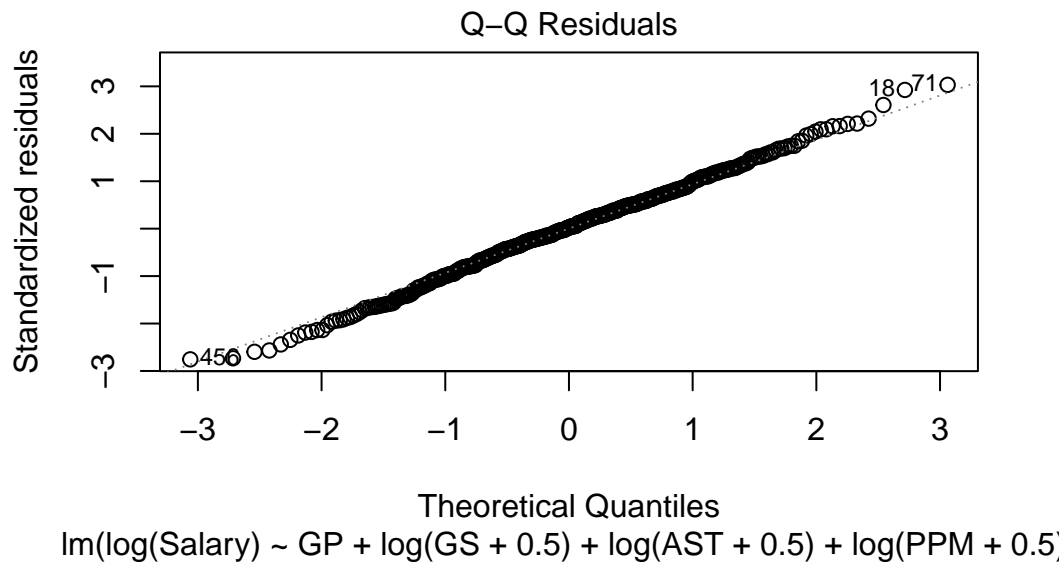
F-statistic: 134.4 on 4 and 449 DF, p-value: < 2.2e-16

```
plot(rookie_bic05,4)
```



```
plot(rookie_bic05,1:2)
```





```
rookie[rookie$Player.Name == "Paolo Banchero", ] #11]
```

	Player.Name	Salary	GP	GS	eFG.	FT.	TRB	AST	STL	BLK	PPM
112	Paolo Banchero	11055120	72	72	0.465	0.738	6.9	3.7	0.8	0.5	0.591716

* Ja Morant 2019-2020

```
# Ja Morant 2019 - 2020
new_val <- data.frame(GP = 67, GS = 67, TRB = 3.9, AST = 7.3, PPM = 17.8/31)
exp(predict(rookie_bic05, new_val))
```

1
20972839

* Paolo Banchero (2022-2023) - rookie of year

```
# Paolo Banchero (2022 - 2023) (year won rookie of year)
new_val <- data.frame(GP = 72, GS = 72, TRB = 6.9, AST = 3.7, PPM = 0.591716)
exp(predict(rookie_bic05, new_val))
```

```
1
15350401
```

* Victor Wembanyama (2023-2024)

```
# new rookie of year - Victor Wembanyama (2023-2024) (year won rookie of year)

new_val <- data.frame(GP = 46, GS = 46, TRB = 11, AST = 3.7, PPM = 24.3/33.2)
exp(predict(rookie_bic05, new_val))
```

```
1
12330470
```

```
# rookie[rookie$Player.Name == "Ja Morant", ]
```

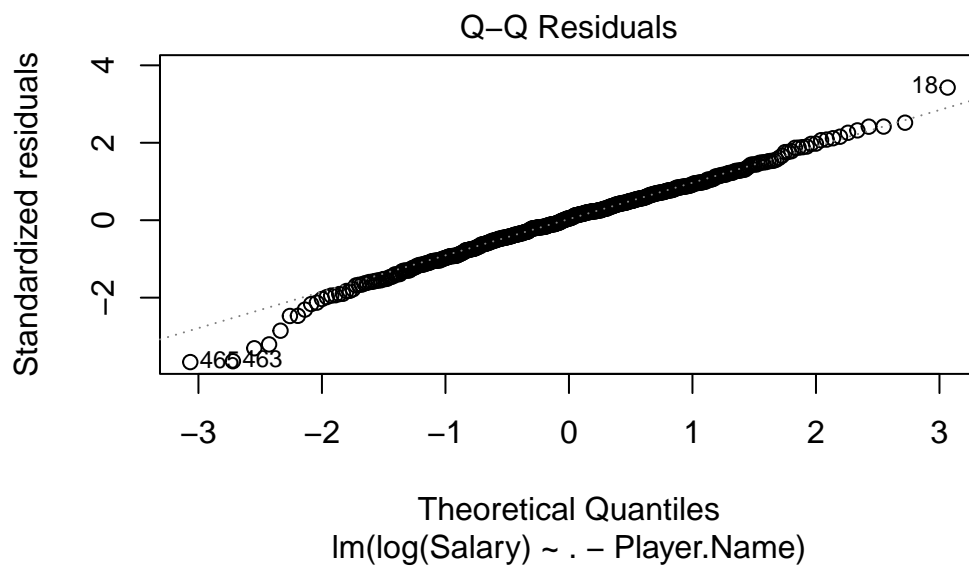
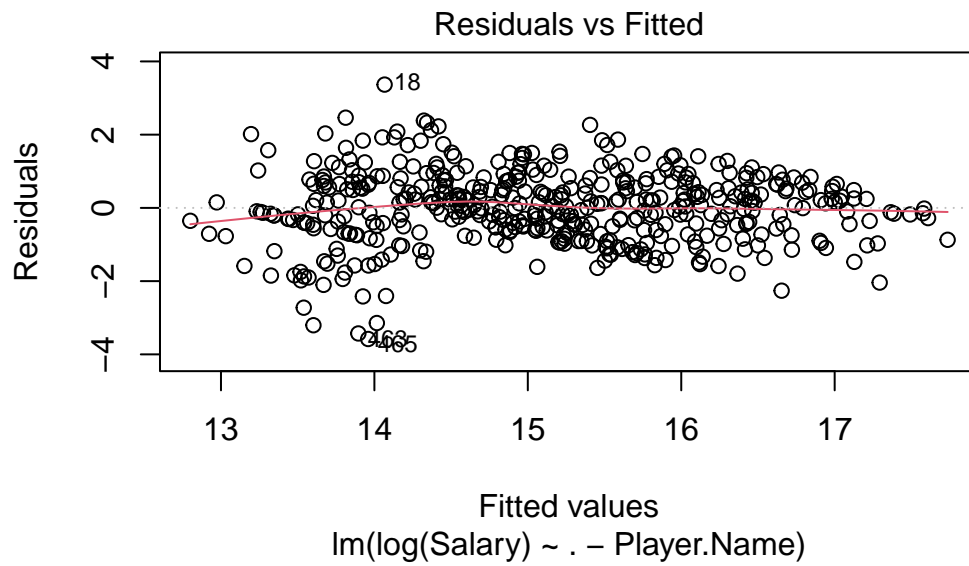
```
# #Joel Embiid 2024
#
# new_val <- data.frame(GP = 39, GS = 39, TRB = 8.2, AST = 4.5, PPM = 23.8/30.2)
# exp(predict(rookie_bic05, new_val))
```

```
# # joker 2019-2020
# new_val <- data.frame(GS =70, PTS =29.6, TRB=12.7)
# # exp(predict(Final_test_bic05, new_val))
# predict(Final_test_aic, new_val)
# exp(17.13199 )
```

```
nba_rm_lev <- nba_ppm[-c(446, 448, 453, 460, 466, 467), ]
dim(nba_rm_lev)
```

```
[1] 461 11
```

```
fin.mod <- lm(log(Salary) ~ . -Player.Name, nba_rm_lev)
plot(fin.mod, c(1,2))
```



MVP Model

take MP * GP (total minutes played)

```
nba_tmp <- nba6
nba_tmp$mp_gs <- nba6$MP * nba6$GP
nba_tmp <- nba6 |> select(-c(GP, MP))
# nba_tmp
```

blocks are not significant

```
nba_tmp2 <- nba_tmp |> select(-c(BLK))
# nba_tmp2
# head(nba_tmp2)
```

stepwise variable selection

```
nba_tmp_model <- lm(log(Salary) ~ . -Player.Name, nba_tmp2)
nba_tmp_model.0 <- lm(log(Salary) ~ 1, nba_tmp2)
n <- nrow(nba_tmp2)

mod1.1 <- step(nba_tmp_model.0, scope = list(lower = nba_tmp_model.0, upper = nba_tmp_model)
mod2.1 <- step(nba_tmp_model.0, scope = list(lower = nba_tmp_model.0, upper = nba_tmp_model)
# summary(mod1)
# summary(mod2)
anova(mod2.1, mod1.1)
```

Analysis of Variance Table

```
Model 1: log(Salary) ~ PTS + GS
Model 2: log(Salary) ~ PTS + GS + TRB
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     464 638.28
2     463 633.59  1     4.6939 3.4301 0.06465 .
---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

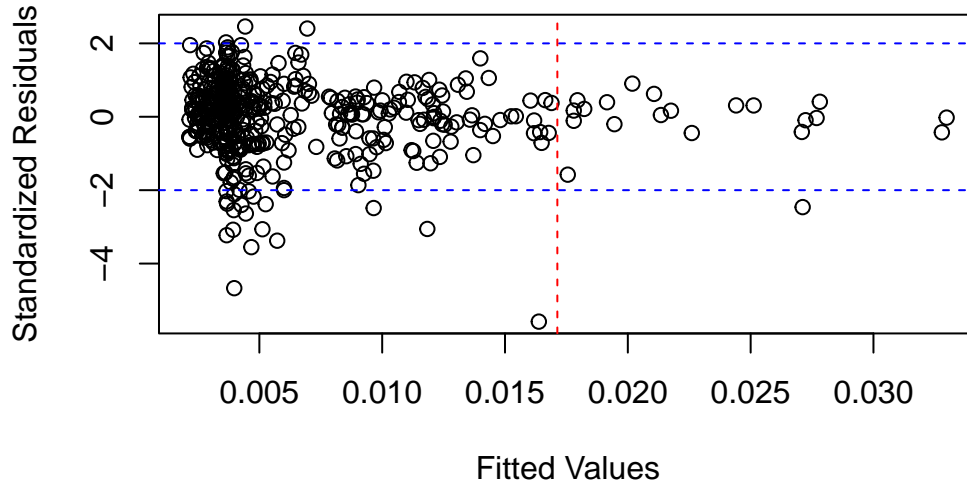
```
AIC(mod2.1, mod1.1)
```

```
      df      AIC
mod2.1  4 1479.202
mod1.1  5 1477.755
```

```
# anova(mod2)
```

```
plot(hatvalues(mod2.1), rstandard(mod2.1), xlab = "Fitted Values", ylab = "Standardized Residuals",
      abline(h = c(2,-2), lty = 2, col = "blue")
```

```
p <- 3
n <- nrow(nba_tmp2)
abline(v = 2 * (p + 1) / n, col = "red", lty = 2)
```



```
ind.1 <- which(abs(rstandard(mod2.1)) > 2 & hatvalues(mod2.1) > 2*(p+1)/n)
leverage_bad <- nba_tmp2[ind.1, ]
leverage_bad
```

	Player.Name	Salary	GS	eFG.	FT.	TRB	AST	STL	PTS
434	Louis King	307089	0	0.769	0	4	2	1	20

removing players

```
mvp <- nba_tmp2[-c(446, 448, 453, 460, 466, 467, 465, 463, 461, 464, 462, 452, 458), ]

mvp.mod <- lm(log(Salary) ~ GS + TRB + PTS, mvp)
# sum(is.na(test))
summary(mvp.mod)
```

Call:

```
lm(formula = log(Salary) ~ GS + TRB + PTS, data = mvp)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-3.2968	-0.6356	0.1226	0.6617	2.7288

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	13.839124	0.088937	155.606	< 2e-16 ***
GS	0.010211	0.002652	3.850	0.000135 ***
TRB	0.094059	0.026448	3.556	0.000416 ***
PTS	0.085817	0.010301	8.331	9.79e-16 ***

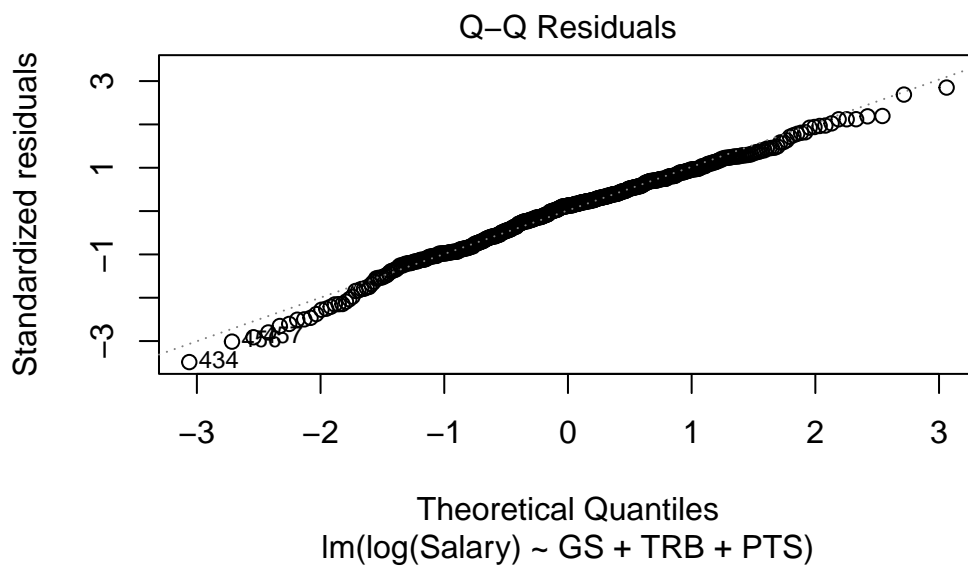
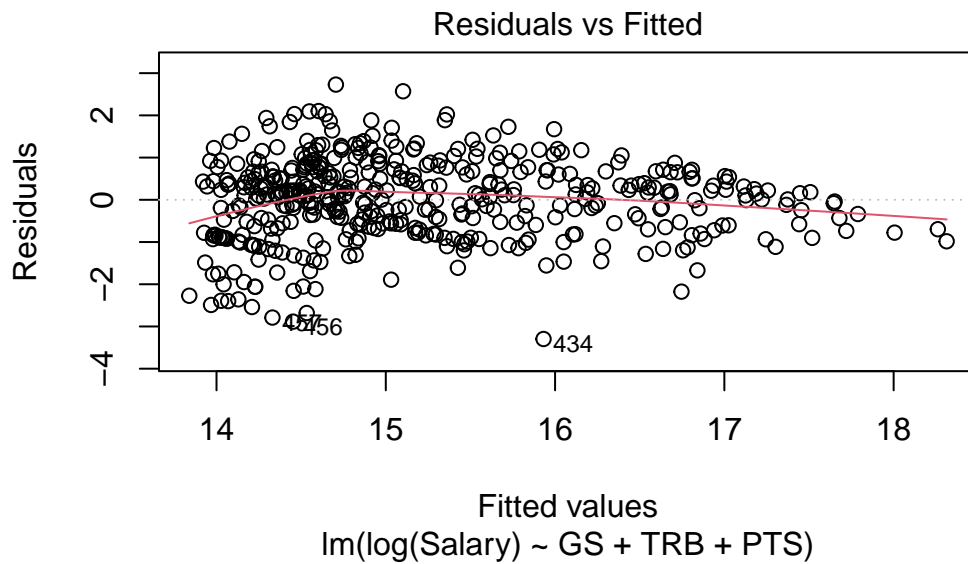
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9599 on 450 degrees of freedom

Multiple R-squared: 0.5127, Adjusted R-squared: 0.5094

F-statistic: 157.8 on 3 and 450 DF, p-value: < 2.2e-16

```
# mvp
plot(mvp.mod, 1:2)
```



transformations (Predictor)

```
pft <- powerTransform(cbind(GS+1,PTS +1 ,TRB +1) ~ 1, mvp)
summary(pft)
```

bcPower Transformations to Multinormality

	Est Power	Rounded Pwr	Wald Lwr Bnd	Wald Up Bnd
Y1	0.0548	0.0	-0.0087	0.1182
Y2	0.0963	0.1	0.0098	0.1829
Y3	0.0123	0.0	-0.1129	0.1376

Likelihood ratio test that transformation parameters are equal to 0
(all log transformations)

	LRT	df	pval
LR test, lambda = (0 0 0)	6.852211	3	0.076761

Likelihood ratio test that no transformations are needed

	LRT	df	pval
LR test, lambda = (1 1 1)	1065.717	3	< 2.22e-16

```
mvp_model <- lm(log(Salary) ~ log(GS+1) + log(TRB+1) + log(PTS+1), mvp)
summary(mvp_model)
```

Call:

```
lm(formula = log(Salary) ~ log(GS + 1) + log(TRB + 1) + log(PTS + 1), data = mvp)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.10419	-0.58135	0.05698	0.64001	2.50088

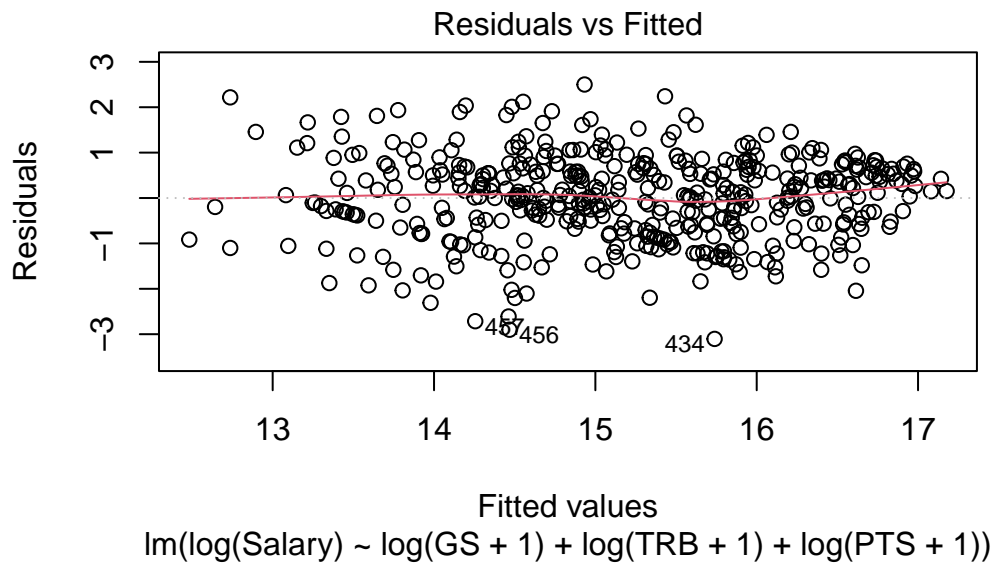
Coefficients:

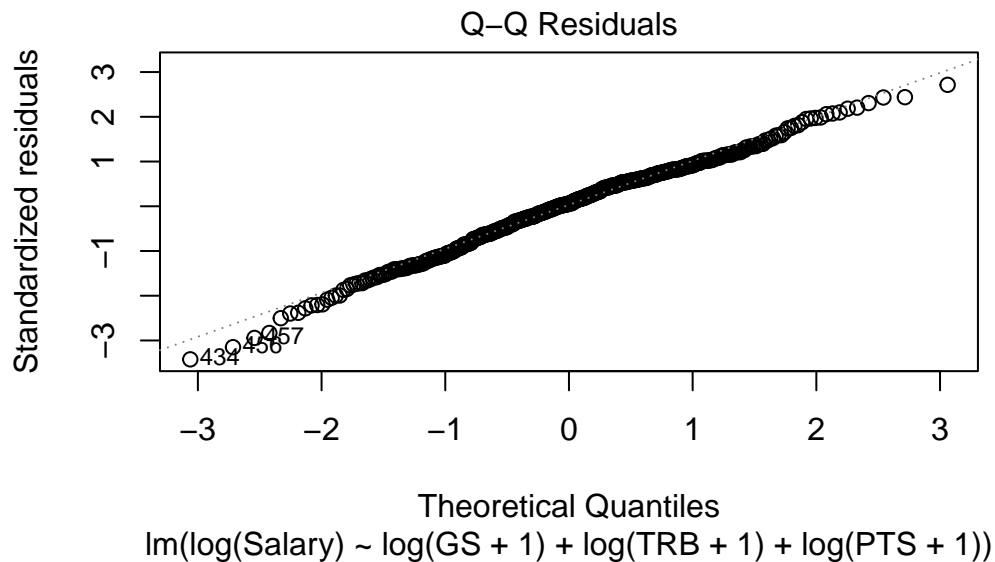
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	12.4831	0.1661	75.159	< 2e-16 ***
log(GS + 1)	0.1885	0.0454	4.152	3.95e-05 ***
log(TRB + 1)	0.2309	0.1347	1.714	0.0872 .
log(PTS + 1)	0.9474	0.1091	8.685	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9257 on 450 degrees of freedom
Multiple R-squared: 0.5468, Adjusted R-squared: 0.5437
F-statistic: 180.9 on 3 and 450 DF, p-value: < 2.2e-16

```
plot(mvp_model, 1:2)
```





correlation

```
# corplot::corplot(cor(mvp[, -1])*(abs(cor(mvp[, -1])) > 0.7), is.corr = FALSE)
```

```
mvp[mvp$Player.Name == "Joel Embiid", ]
```

	Player.Name	Salary	GS	eFG.	FT.	TRB	AST	STL	PTS
28	Joel Embiid	33616770	66	0.573	0.857	10.2	4.2	1	33.1

Giannis Antetokounmpo (19-20)

```
#giannis(MVP 2019-2020)
new_val <- data.frame(GS =63, PTS =29.5, TRB=13.6)
predict(mvp_model, new_val)
```

1
17.12401

```
exp(17.12401)
```

```
[1] 27344063
```

Joel Embiid (22-24)

```
#Joel Embiid(MVP 2022-2023)
new_val <- data.frame(GS =66, PTS =33.1, TRB =10.2)
predict(mvp_model, new_val)
```

```
      1
17.17712
```

```
exp(17.17712)
```

```
[1] 28835563
```

Nikola Jokic (Joker) (24-25)

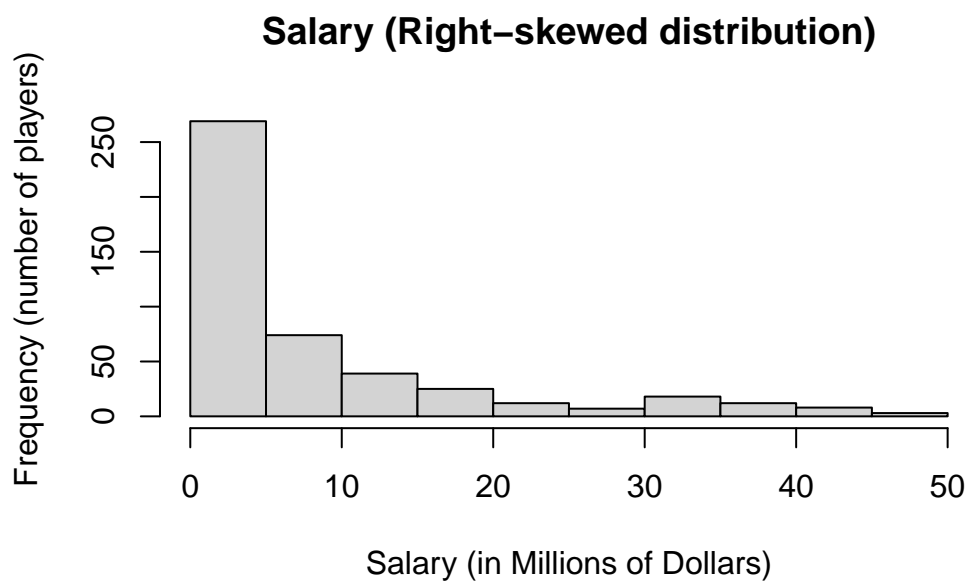
```
#Joker Nikola Jokic(2024-2025)
new_val <- data.frame(GS =70, PTS =29.6, TRB=12.7)
predict(mvp_model, new_val)
```

```
      1
17.13199
```

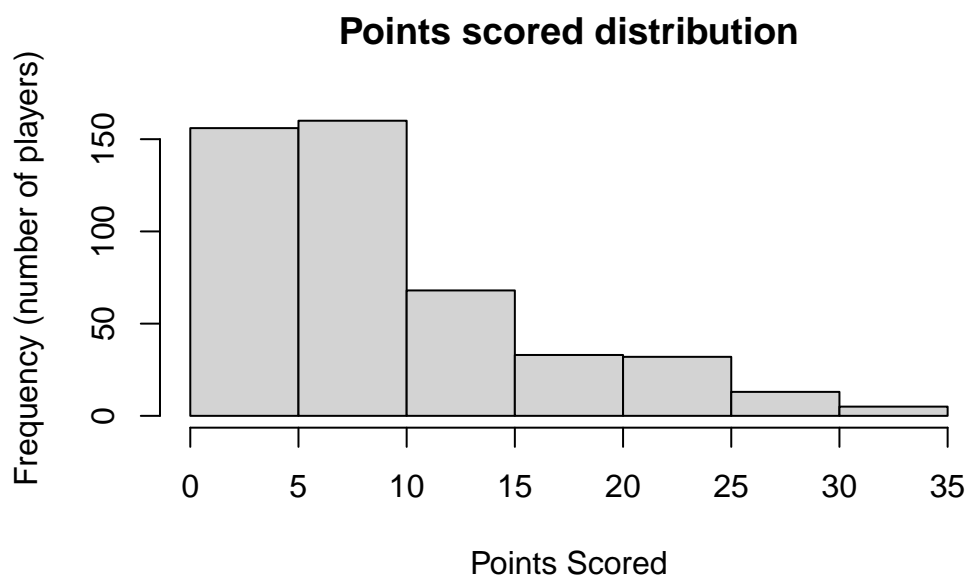
```
exp(17.13199)
```

```
[1] 27563142
```

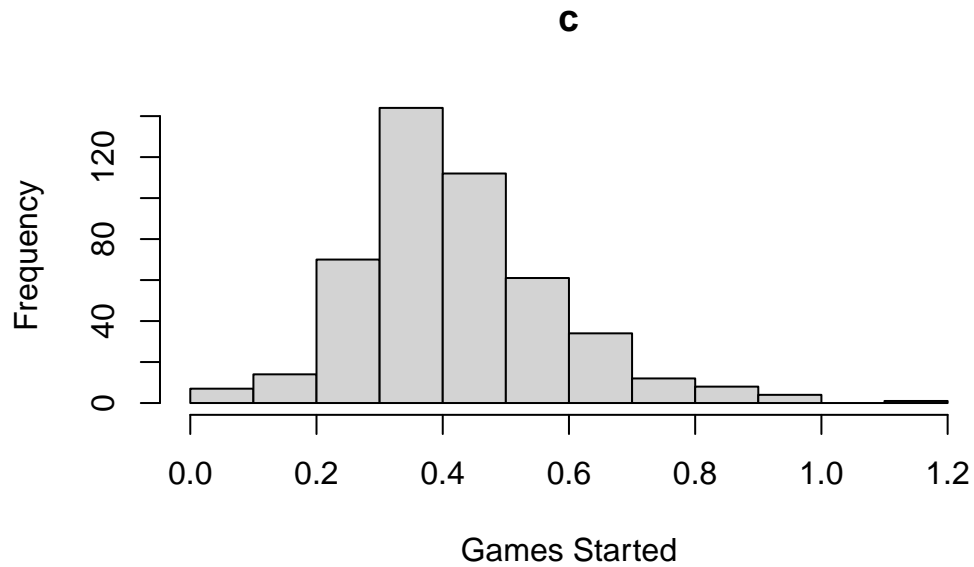
```
hist(nba3$Salary/1000000, xlab = "Salary (in Millions of Dollars)", ylab = "Frequency (number of players)")
```



```
hist(nba3$PTS, xlab = "Points Scored", ylab = "Frequency (number of players)", main = "Points
```



```
hist(nba_ppm$PPM, xlab = "Games Started", ylab = "Frequency", main = "c")
```



```
# library(ggplot2)
#
#
# ggplot(nba3, aes(x = nba3$Salary/1000000)) +
#   geom_density(fill="blue", alpha=0.5) +
#   labs(title="Salary Distribution (Right-skewed tail)", x = "Salary (in Millions of Dollars)")
#
# ggplot(nba3, aes(x = log(nba3$Salary))) +
#   geom_density(fill="blue", alpha=0.5) +
#   labs(title="Density Plot")
```