NBA Salary Prediction through Integrated Player Statistics and Contract Data

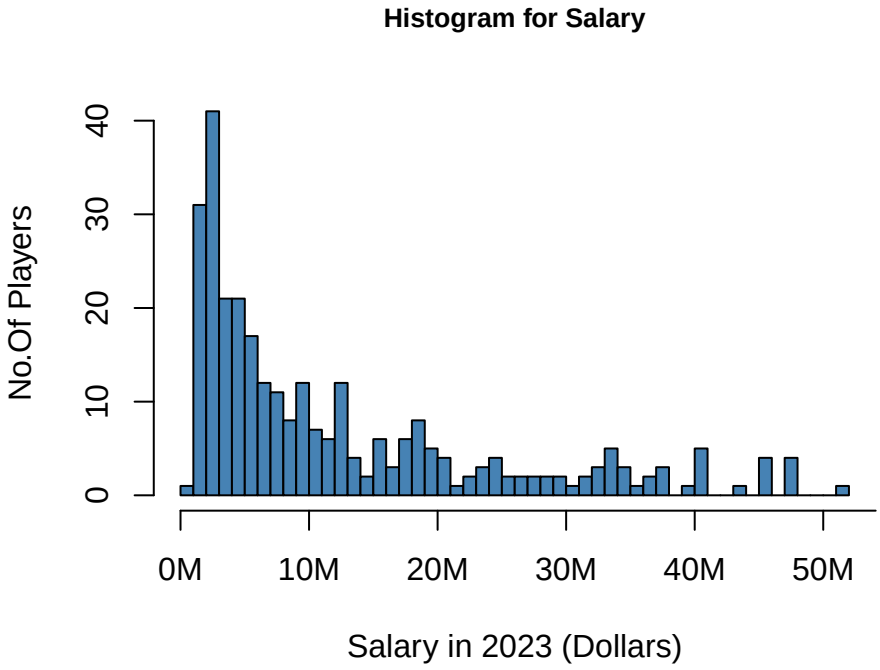
Gagan Ullas Nirgun, Pramodh Sairam Punjai Venkataraman, Ganesh Akula, Viswadeep Mallarapu Bhaskar

Introduction

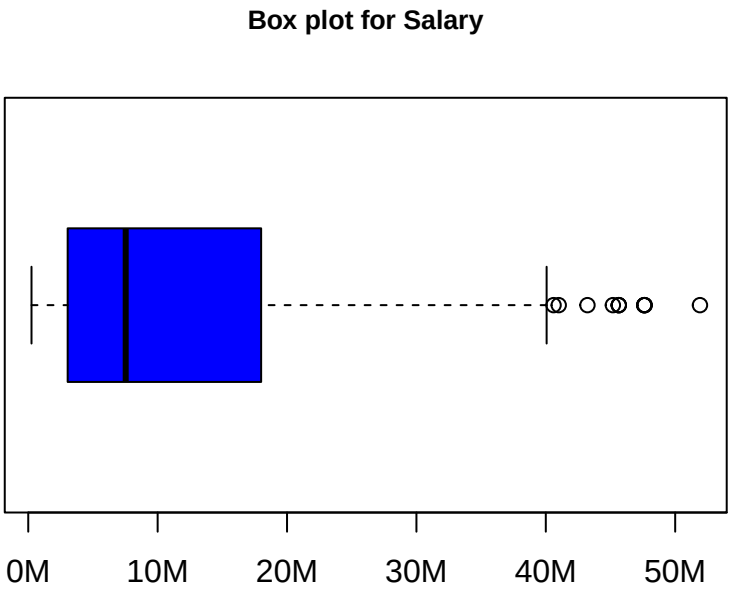
- **Motivation :** Our project question is motivated by a recognition of the complex interplay between on-court performance and monetary rewards. We aim to construct a model capable of estimating player salaries, thereby in the future offering a powerful tool for stakeholders to make informed decisions concerning player contracts.
- **Data Description :** The dataset for our project was sourced from two primary sources, Basketball-Reference's NBA 2023 Totals (https://www.basketball-reference.com/leagues/NBA_2023_totals.html) and Player Contracts (https://www.basketball-reference.com/contracts/players.html), forms the backbone of our comprehensive analysis. The amalgamation of these datasets is crucial for a holistic understanding of the factors influencing player salaries. This website provides detailed statistics for NBA players in the 2023 season. The data includes player names, teams, games played, minutes played, assists, rebounds, steals and other relevant metrics. The dataset is comprehensive, covering a wide range of player performance indicators. The original curator of the data is Basketball-Reference, a reputable source for NBA statistics.
- **Data Dictionary :**
  - `Salary`– Salary of the Player. This is the response variable.
  - `Player` – Player Name
  - `Pos` – Position
  - Age – Player's age on February 1 of the season
  - `Tm` – Team
  - `G` – Games
  - `GS` – Games Started
  - `MP` – Minutes Played
  - `FG` – Field Goals
  - `FGA` – Field Goal Attempts
  - `FG%` – Field Goal Percentage
  - `3P` – 3-Point Field Goals
  - `3PA` – 3-Point Field Goal Attempts
  - `3P%` – 3-Point Field Goal Percentage
  - `2P` – 2-Point Field Goals
  - `2PA` – 2-point Field Goal Attempts
  - `2P%` – 2-Point Field Goal Percentage
  - `eFG%` – Effective Field Goal Percentage. This statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal.

- `FT` – Free Throws
- `FTA` – Free Throw Attempts
- `FT%` – Free Throw Percentage
- `ORB` – Offensive Rebounds
- `DRB` – Defensive Rebounds
- `TRB` – Total Rebounds
- `AST` – Assists
- `STL` – Steals
- `BLK` – Blocks
- `TOV` – Turnovers
- `PF` – Personal Fouls
- `PTS` – Points

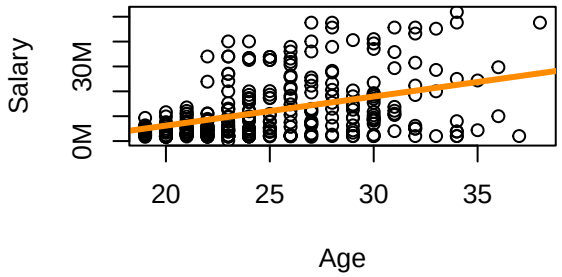
- **Exploratory Data Analysis**

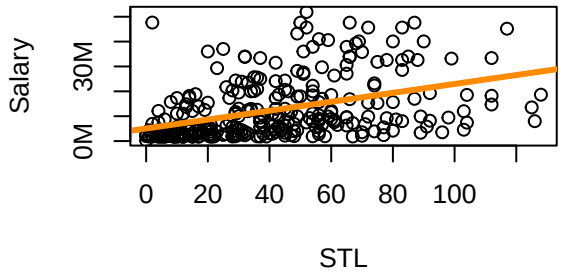


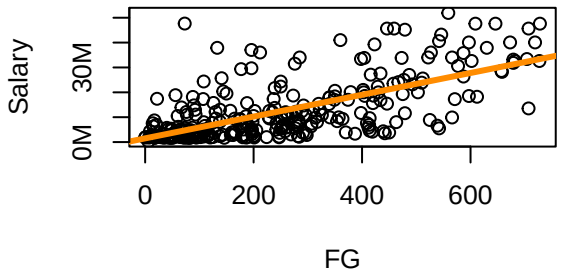
- We can observe that Histogram looks right skewed,While most of the players salary lie in the range until 10M , There are only few players (less than 10) who are in higher salary range (around 50 M).

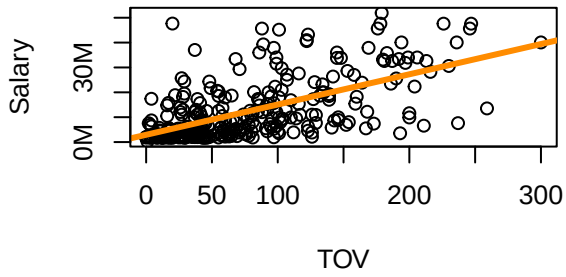


- Based on the Box-Plot we can see that the median salary of players is roughly around 5M to 10M range and there are a few players having salary greater than the upper range of the boxplot.
- **Linear Plots**

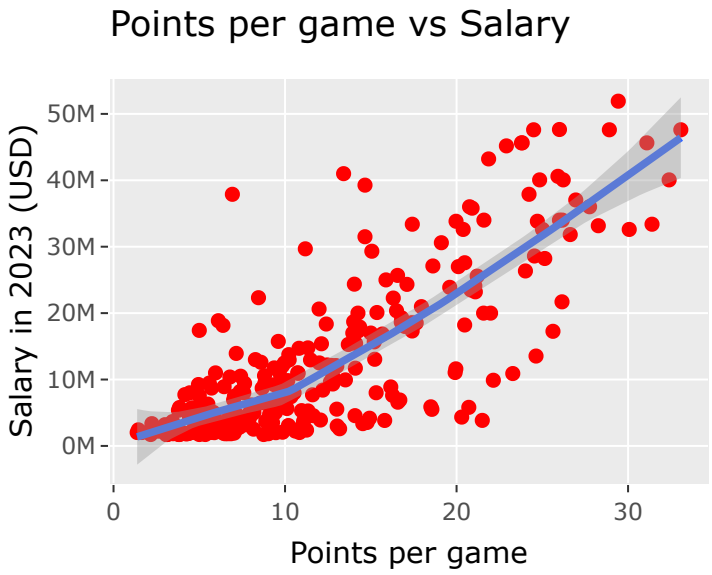




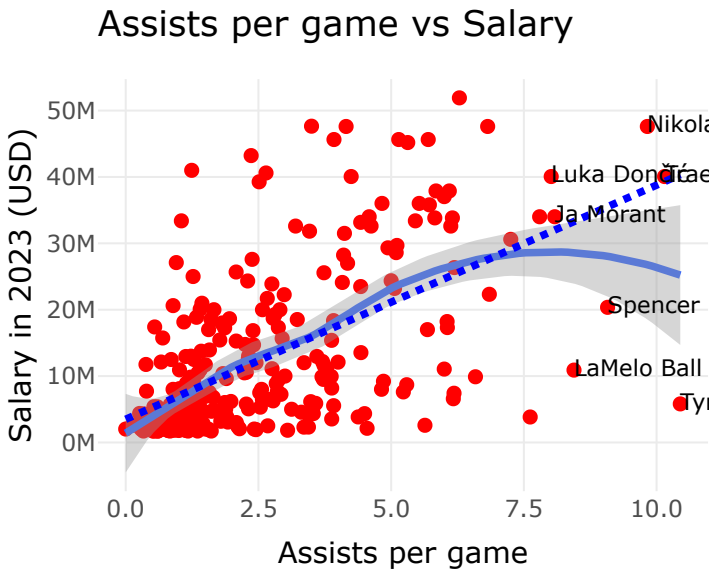




- Age,Steals,Field Goals and Turnovers metrics show a promising relationship with Salary.
- Other potential significant variables might be : Minutes Played, Assists, Points,Total Rebounds.



- As players consistently contribute more than nine points per game, the correlation strengthens, suggesting that higher-scoring players tend to command higher salaries.



- There is a positive correlation, suggesting that as players contribute more assists, their salaries tend to increase. However, beyond a threshold of around six assists per game, the correlation turns negative As players with high assist rates focus on facilitating plays rather than scoring, they may not be recognized as the team's top player, potentially influencing their compensation to be comparatively lower.

Regression Analysis

- **Data Cleaning**

1) There are 3 categorical columns(Player,Tm,Pos) in dataset, but for this particular analysis we are dropping all the categorical variables.

2) FT%, 2P%, and 3P% columns have NaN values, Since the FT% is calcuated by dividing the Free Throws by Free Throws Attempted, for those columns with NaN values in FT%, both the FT and FTA are 0 , thus we replace FT% with 0. This process is followed for the 2P% and 3P% columns as well as they follow the same pattern as FT%.

3) Data set contains players who have played less than 10 games during the season. These data points with a small sample size of playing time had high variance and were difficult to model, which made dropping them the right choice.

4) The final shape of the data set after cleaning is 278 observations with 27 columns.

- **Model Selection**

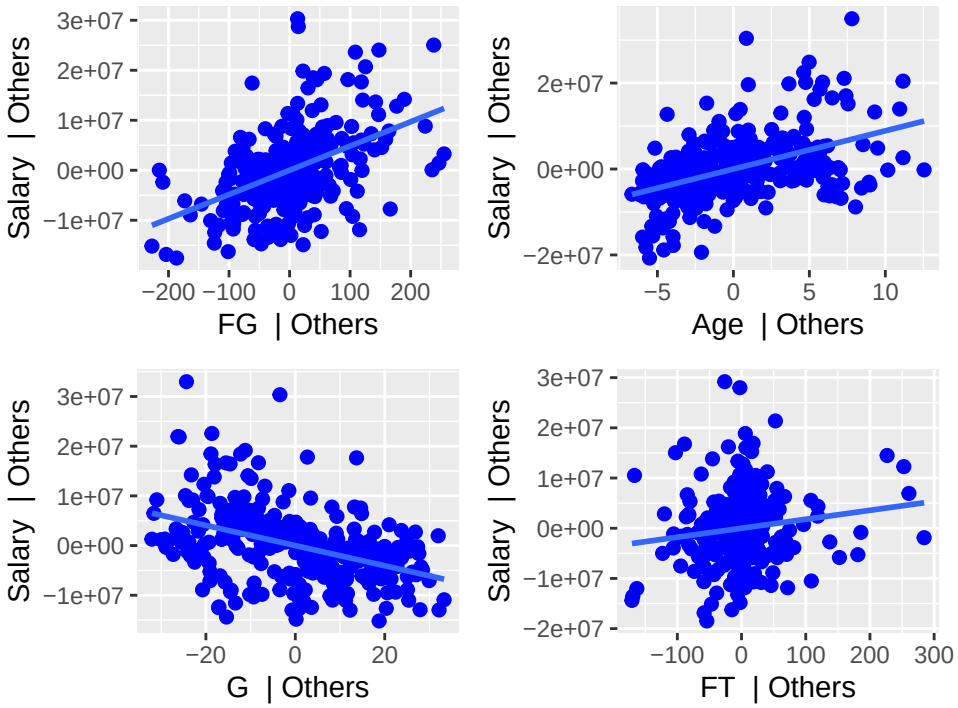
1) On fitting the model with all variables, we found out that, 2P, 2PA, TRB, and PTS are showing exact collinearity to Salary, so we choose to drop these columns from the model and re-fit with rest of the predictors.

2) After refining the model by addressing issues of complete collinearity and removing redundant columns, our analysis reveals that only the variables FG (Field Goals), Age, and G (Games Played) exhibit a statistically significant linear relationship with salary given the other predictors are in the model.

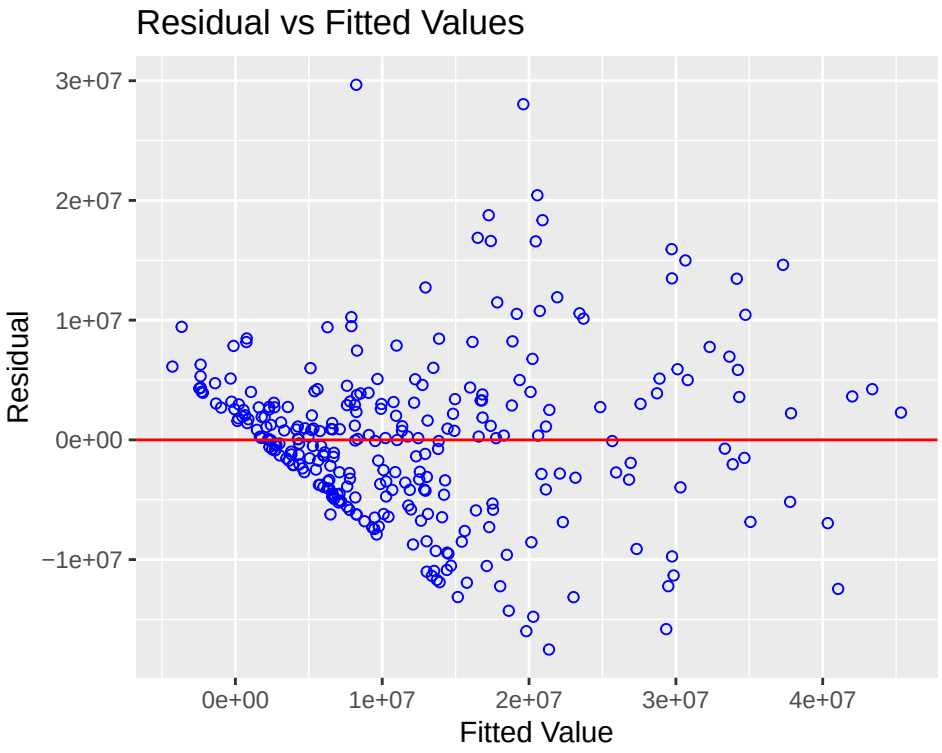
3) As we explore variable selection, it becomes apparent that numerous variables may not contribute significantly to our prediction model. In light of this, we proceed with the process of selecting relevant variables, aiming to streamline the model and focus on those factors that have meaningful impact on the prediction of salaries in our context.

Model	**RMSE**	**Variables Selected**
Forward AIC model	6795297	FG,G,Age,FT
Forward BIC model	6795297	FG,G,Age,FT
Backward AIC model	6801285	Age,G,FG,FGA,FG.,X3P,X3P.,FTA,AST
Backward BIC model	6798484	Age,G,FG,FTA
Stepwise AIC model	6795297	FG,G,Age,FT
Stepwise BIC mode	6795297	FG,G,Age,FT
Exhaustive AIC model	6796606	Age,G,FG,FGA,FG.,X3P,X3P.,FT ,AST
Exhaustive BIC model	6795297	FG,G,Age,FT
Adjusted R2 model	6826957	Age,G,FG,FGA,FG.,X3P,X3P.,FTA,FT.,AST,BLK

- Considering the outcomes above, our preference leans towards the model with the lowest Leave one out Root Mean Squared Error ($\text{RMSE}_{\text{LOOCV}}$). This choice is motivated by the desire to have a model that exhibits better overall accuracy in predicting salary values, as indicated by the minimized RMSE.
- The selected model comprises relevant predictors, such as FG (Field Goals), Age, and G (Games Played) and Free Throws (FT) reflecting a robust approach to capturing the underlying relationships in our data. (Salary ~ Age + G + FG + FT) .
- After fitting the selected model, we observe that approximately 70% of the variability in the response variable (Salary) is accounted for by the predictors included in our chosen model.
- We also recognize that, following variable selection, the significance tests may experience a reduction in power. To validate and justify our chosen model, we may opt for a train-test split approach.
- **Model Diagnostics**
- Focusing on model diagnostics to assess the validity of our linear regression model and ensure that it meets the necessary assumptions.

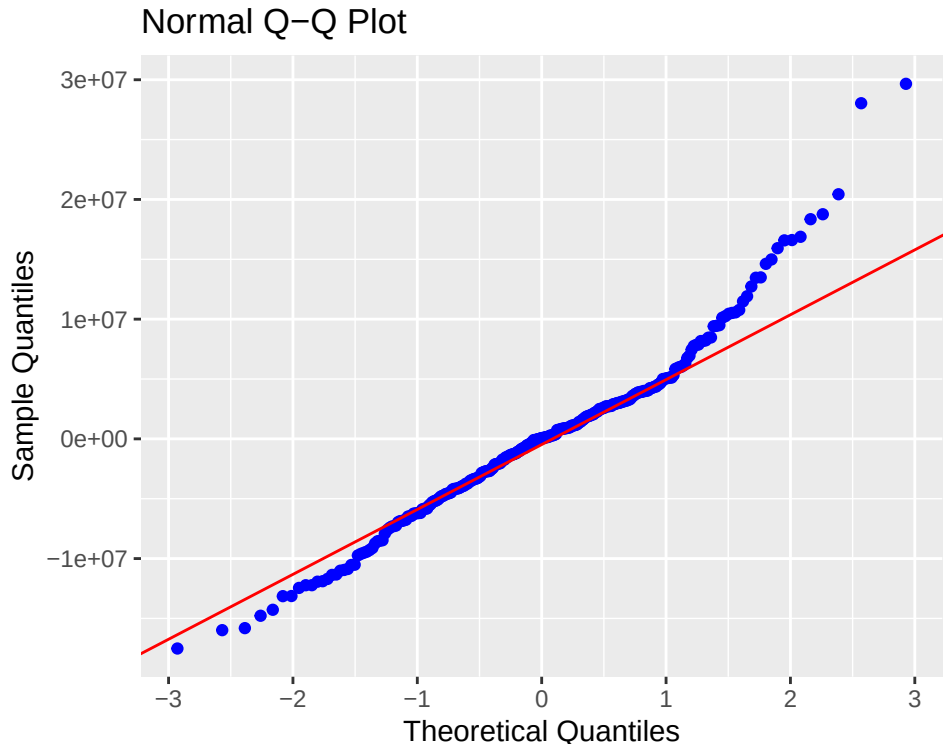


- From above plots, it looks like there is a significant linear relationship between each predictor and the response, given the other predictor are in the model.

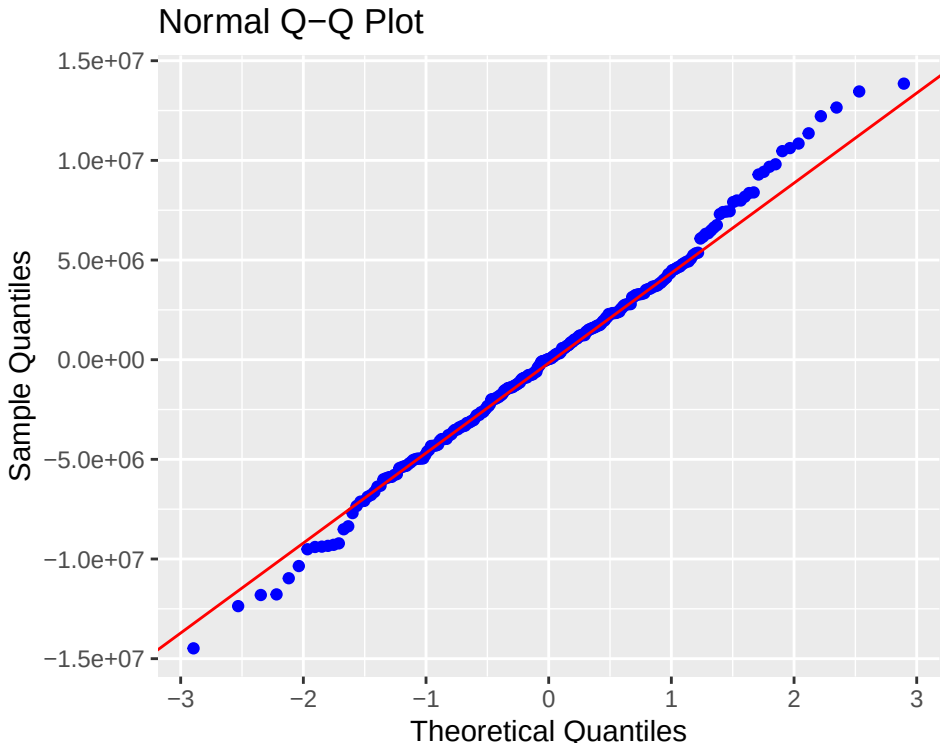


- The fitted-vs-residuals plot does not look good. In particular, the variance tends to increase as the fitted values increase. As such, The constant variance assumption has been violated.
- The value of the test statistic is 31.744 with a p-value of 2.157e-06. Hence, we reject the null hypothesis

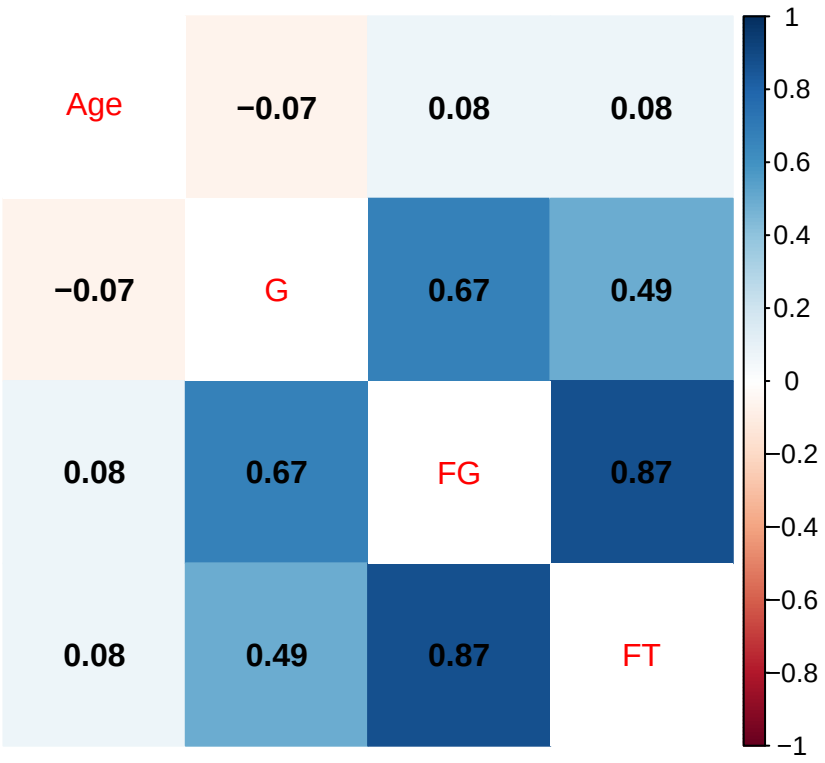
and conclude that the errors are heteroscedastic.



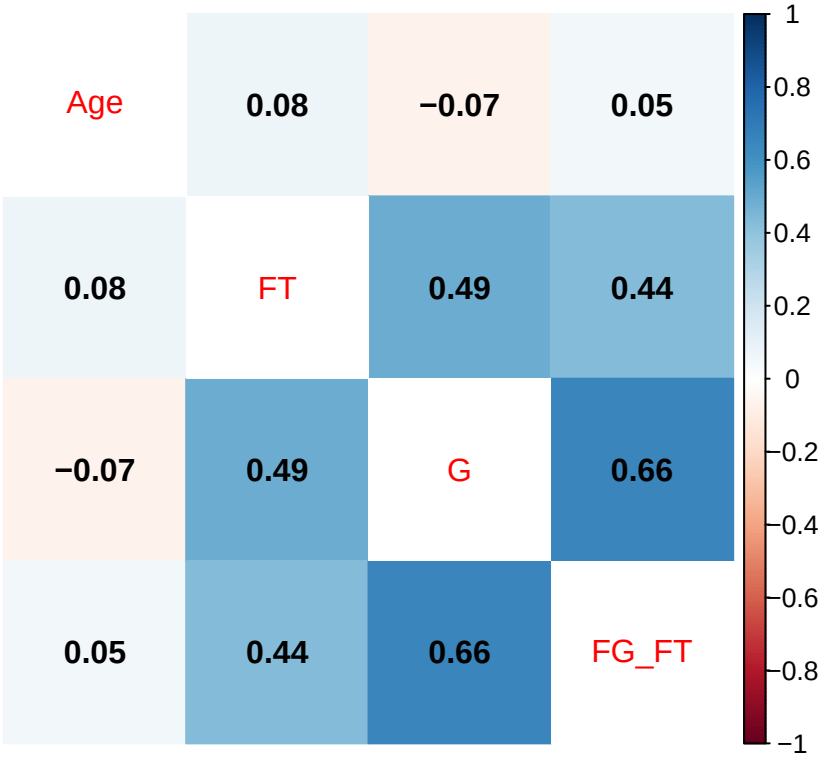
- Above Q-Q plot does not look good. Especially at end quantiles , the points do not follow the line closely at all.
- The test statistic of the Shapiro-Wilk test is 0.97348 and p-value is 4.965e-05.So, we reject the null hypothesis and conclude that the errors are not normally distributed.
- **Fixing Model Violations**
- To address the violation of the normality assumption in our model, we will investigate the presence of influential points within our dataset.
- There are 31 observations identified as highly influential points in our dataset. These points exert a notable influence on the regression model.



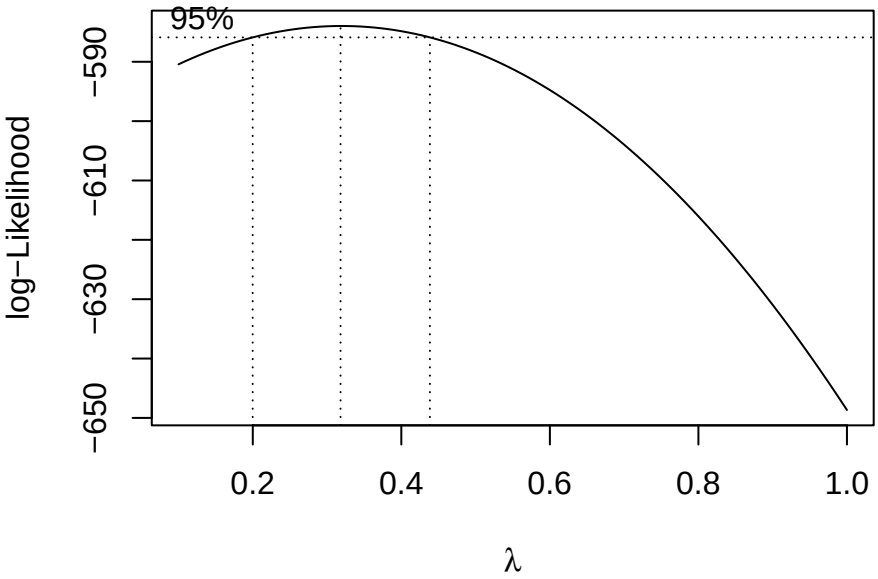
- The Q-Q plot looks very good. Most points fall around the line.
- The p-value of the Shapiro-Wilk test is 0.449, so we do not reject the null hypothesis and conclude that the errors are normally distributed.
- Removing influential points fixed the normality issues with the regression.
- **Fixing Collinearity Violations**



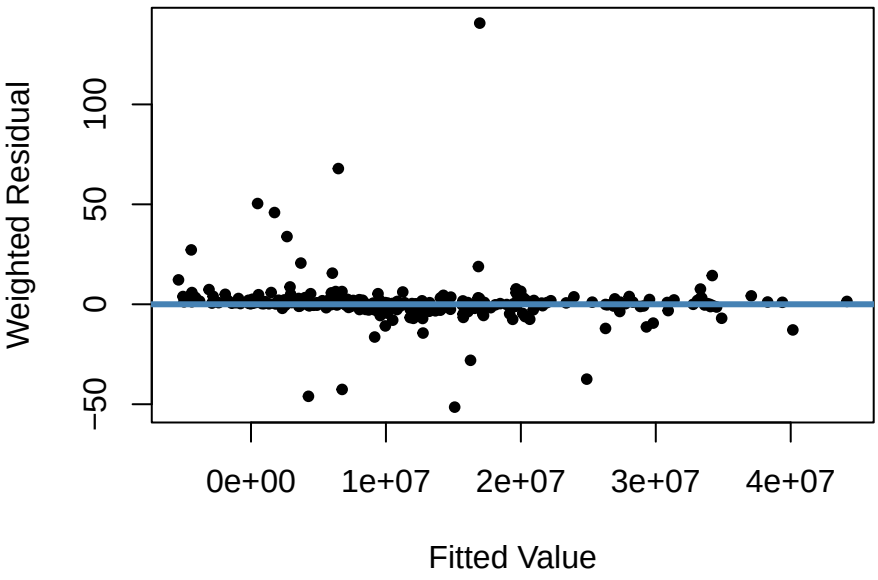
- From the above heatmap we can see that the variables FG and FT are highly correlated. Thus as the next step , we can perform feature engineering, to create new features.
- Generally, Field Goals are inclusive of the Free Throws , thus we can create a new variable which is the on field goals that are not free throws denoted as FG_FT(Field goals without free throws).



- After introducing the feature-engineered variable, we notice a reduction in the correlation among variables, effectively addressing the collinearity issues. Moving forward, our next steps involve conducting formal diagnostics to confirm that collinearity is no longer a concern among the model predictors.
- After performing formal diagnostics the condition number obtained is 20.22 and there are no condition indices greater than 30 , and also all the VIFs calculated are less than 5, thus we can alleviate concerns regarding multicollinearity as it is no longer a significant issue in our model.
- The model with the feature engineered variable as a R^2 value of 0.7848, but the constant variance violation still exists.
- To address the violation of the constant variance assumption in our model, we will do the Transformations.



- On observing above plot,the $\lambda = 0.3272727$ which is near to 0.5, so the Box-Cox method justifies our choice of a square root transformation of the response variable.
- The value of the test statistic is 11.428 with a p-value of 0.02216. Hence, we reject the null hypothesis and conclude that the errors are still heteroscedastic.
- As a next step , we move on to create a weighted least square model. Since our data are only recorded for a single year, we can reasonably assume that errors are independent, given that the data is not measured over time.



- The value of the Breusch Pagan test statistic is 3.4934e-11 with a p-value of 1. Hence, we fail to reject the null hypothesis and conclude that the errors are homoscedastic.
- WLS plots look better, we do not see any obvious pattern, and the spread about zero seems to be roughly the same. so constant variance assumption for this model is fixed.
- After fitting the WLS model, we observe that approximately 73% of the variability in the response variable (Salary) is accounted for by the predictors included in our chosen model.

Discussion

- The estimated regression equation of the Weighted Least Squared model is
- $\hat{Salary_i}$ = -25907926.1 + 92258.7 FG_FT_i + 20796.4 FT_i + 1482321.7 Age_i - -254044.2 G_i.
- We observe that approximately 73% of the variability in the response variable (Salary) is accounted for by the predictors included in our chosen model.
- The p-value of the test is 2.2e-16 and thus we reject null hypothesis and conclude that there is a significant linear relationship between the response and the predictors given all the predictors are in the model.
- All the individual predictors are in a significant linear relationship with the response when the other predictors are in the model.
- The initial model incorporated all predictors, providing a baseline for comparison. While achieving a R-squared value of 71%, it served as a starting point for further enhancement.

Model Explanation

Model	R2
OLS with all Predictors	0.71
OLS with all but except exact collinearity	0.7142
OLS with Variable selected model	0.6969
OLS with Featured engineered	0.7848
Squared Root Transformation model	0.5628
Weighted Least Squares	0.7363

- Our journey of model refinement began with an initial Ordinary Least Squares (OLS) model incorporating all predictors, yielding an R-squared of 0.71. Recognizing and addressing exact collinearity issues in the second step led to a marginal improvement (R-squared: 0.7142). Subsequent efforts focused on model streamlining through variable selection, resulting in a lower R-squared (0.6969). A significant breakthrough was achieved with feature engineering, specifically designed to address collinearity issues among the predictors, which elevated the R-squared to 0.7848. In an attempt to rectify the constant variance violation, a squared root transformation was introduced, but unfortunately, it did not fix the issue, leading to a decrease in R-squared (0.5628). This underscores the nuanced nature of transformations and the importance of adapting strategies to the unique characteristics of the data. Addressing heteroscedasticity through Weighted Least Squares significantly improved the model, yielding an R-squared of 0.7363.

Prediction Accuracy on Test Data

Model	RMSE
lasso_model_min	7434703
lasso_model_1se	7135546
Weighted Least Squares	6905880

- In our project report, we employed three key predictive models to forecast outcomes, each assessed by Root Mean Square Error (RMSE). The Lasso Model with Minimum RMSE exhibited a baseline accuracy with an error of 7,434,703. The Lasso Model with 1 Standard Error balanced accuracy and complexity, achieving improved predictions with an RMSE of 7,135,546. Our selected Weighted Least Squares Model outperformed the others, yielding the lowest RMSE at 6,905,880.

Limitations

- The omission of player position as a categorical variable represents a considerable limitation, as player salaries are inherently influenced by their positions on the basketball court. Recognizing this omission is essential, as it may compromise the model's ability to capture the unique contributions and positional demands that affect salary negotiations.
- Furthermore, the understanding that the model's RMSE could potentially be enhanced with the inclusion of additional categorical predictors emphasizes the need for a more comprehensive feature set.
- Additionally, it's crucial to note that the significance test may lose power post-variable selection, potentially affecting the model's robustness. To address this, a train-test split can be implemented offering a more rigorous assessment of the model's generalization in the context of player positions and other pertinent factors.

Conclusions

- FG_FT (Field Goals - Free Throws) reflects scoring efficiency, FT captures free throw accuracy, Age accounts for experience, and Games represent player durability. Research insights include understanding player valuation preferences (efficiency vs. volume scoring), trends over player age, the impact of game participation on salary, and considerations for model validation.The model serves as a valuable tool for researchers examining the nuanced factors influencing NBA player salaries.
- **Ideas about future work**
- Explore advanced feature engineering techniques to uncover complex relationships within the data. This may involve creating interaction terms, polynomial features, or other transformations to capture non-linear patterns and interactions among predictors.
- Consider the application of more advanced statistical techniques, such as machine learning algorithms (e.g., random forests, gradient boosting) or deep learning models, to capture intricate patterns and dependencies within the data that may be challenging for linear models to uncover.

Code Appendix

```
# Plotting Histogram for Salary

players_data=read.csv('/Users/gaganullas19/Downloads/AR_dataset_v1.csv')
hist(players_data$Salary,xlab='Salary in 2023 (Dollars)',
     ylab='No.Of Players',main='Histogram for Salary',
     breaks=50,col='steelblue',cex.main = 0.8, xaxt='n')
# Customize x-axis labels
axis(side = 1, at = seq(0, max(players_data$Salary) + 1e7, by = 1e7),
     labels = paste0(seq(0, max(players_data$Salary) + 1e7, by = 1e7) / 1e6, "M"))



# Plotting Box-Plot for Salary

boxplot(players_data$Salary,
        col='blue',main='Box plot for Salary',horizontal=TRUE,cex.main = 0.8,xaxt='n')
axis(side = 1, at = seq(0, max(players_data$Salary) + 1e7, by = 1e7),
     labels = paste0(seq(0, max(players_data$Salary) + 1e7, by = 1e7) / 1e6, "M"))
##str(players_data |> select(Pos))

# Linear Plots

par(mfrow = c(2, 2))
##EDA
mdl=lm(Salary~Age,players_data)
plot(Salary ~ Age, data = players_data,yaxt='n')
abline(mdl, lwd = 3, col = "darkorange")
axis(side = 2, at =
       seq(0, max(players_data$Salary) + 1e7, by = 1e7),
     labels = paste0(seq(0, max(players_data$Salary) + 1e7, by = 1e7) / 1e6, "M"))
#summary(mdl)

mdl=lm(Salary~STL,players_data)
plot(Salary ~ STL, data = players_data,yaxt='n')
abline(mdl, lwd = 3, col = "darkorange")
axis(side = 2, at =
       seq(0, max(players_data$Salary) + 1e7, by = 1e7),
     labels = paste0(seq(0, max(players_data$Salary) + 1e7, by = 1e7) / 1e6, "M"))
#summary(mdl)

mdl=lm(Salary~FG,players_data)
plot(Salary ~ FG, data = players_data,yaxt='n')
abline(mdl, lwd = 3, col = "darkorange")
axis(side = 2, at =
       seq(0, max(players_data$Salary) + 1e7, by = 1e7),
     labels = paste0(seq(0, max(players_data$Salary) + 1e7, by = 1e7) / 1e6, "M"))
#summary(mdl)

mdl=lm(Salary~TOV,players_data)
plot(Salary ~ TOV, data = players_data,yaxt='n')
abline(mdl, lwd = 3, col = "darkorange")
axis(side = 2, at =
```

```
       seq(0, max(players_data$Salary) + 1e7, by = 1e7),
     labels = paste0(seq(0, max(players_data$Salary) + 1e7, by = 1e7) / 1e6, "M"))
#summary(mdl)

# Plotting Points per game vs Salary

library(ggplot2)
library(plotly)
library(htmltools)

# Assuming 'players_data' is your data frame
ggplotly(ggplot
         (na.omit(players_data), aes(x = PTS/G, y = Salary)) +
    geom_point(col = "red") +
    geom_smooth(formula = y ~ x, method = "loess") +
    labs(title = "Points per game vs Salary",
         x = "Points per game", y = "Salary in 2023 (USD)") +
    scale_y_continuous(labels = function(x) paste0(x/1e6, "M"),
                       breaks = seq(0, 50000000, by = 10000000)))




# Plotting Assists per game vs Salary

library(ggplot2)
library(plotly)
library(htmltools)


ggplotly(
  ggplot(na.omit(players_data),
         aes(x = AST/G, y = Salary, label = ifelse(AST/G > 8, Player, ""))) +
    geom_point(col = "red") +
    geom_smooth
  (formula = y ~ x, method = "loess") +
    geom_smooth
  (formula = y ~ x, method = "glm", linetype = "dotted", col = "blue", se = FALSE) +
    labs(title = "Assists per game vs Salary",
         x = "Assists per game", y = "Salary in 2023 (USD)") +
    geom_text(size = 3) +
    scale_y_continuous
  (labels =

      function(x) paste0(x/1e6, "M"), breaks = seq(0, 50000000, by = 10000000)) +
    theme_minimal()  # Optional: Use a minimal theme for better aesthetics
) %>%
style(textposition = "right")

# DataSet
#head(players_data, n = 2)

# Dropping categorical columns(Player Name, Tm, Pos) in our dataset
```

```r
names(players_data[sapply(players_data, is.character)])

#library(fastDummies)

#players_data <- dummy_cols(players_data,   select_columns = "Pos")
#
players_data<-players_data[,-c(1,2,4)]
head(players_data)
# Checking for Nan values in our dataset

colSums(is.na(players_data))

#Null values in FT%, 2P%, and 3P% are replaced with 0's.

library(tidyr)
players_data[is.na(players_data$'FT.'),][,c('FT','FTA','FT.')]

players_data$'FT.'=players_data$'FT.' %>% replace_na(0)
players_data$'X2P.'=players_data$'X2P.' %>% replace_na(0)
players_data$'X3P.'=players_data$'X3P.' %>% replace_na(0)
# Shape of cleaned dataset
dim(players_data)
# Re-checking for Nan Values

colSums(is.na(players_data))


#Dropping players who played 10 or fewer games

players_data=players_data[players_data$G>10,]
dim(players_data)

#Fitting model with All variables in the dataset
full_OLS_model<-lm(Salary ~ ., data= players_data)
#summary(full_OLS_model)

# function to calculate RMSE

rmse = function(model){
  sqrt(mean((players_data$Salary - predict(model))^2))
}

#rmse(full_OLS_model)
# Dropping 2P, 2PA, TRB, and PTS

full_OLS_model_refitted_with_non_sig<-
  lm(Salary ~ . - X2P - X2PA - TRB - PTS , data=players_data)
summary(full_OLS_model_refitted_with_non_sig)

# Work code

str(players_data)
```

```r
# calculate RMSE for full refitted model

full_OLS_model_refitted_with_non_sig<-
  lm(Salary ~ . - X2P - X2PA - TRB - PTS , data=players_data)
rmse(full_OLS_model_refitted_with_non_sig)
#AIC (Backward selection)

library(faraway)
full_OLS_model_refitted_with_non_sig<-
  lm(Salary ~ . - X2P - X2PA - TRB - PTS , data=players_data)

mod_back_aic=step(full_OLS_model_refitted_with_non_sig,direction='backward')

calc_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}
calc_loocv_rmse(mod_back_aic)
summary(mod_back_aic)$adj.r.squared
#BIC (Backward selection)
n=nrow(players_data)
mod_back_bic=
  step(full_OLS_model_refitted_with_non_sig,direction='backward',k=log(n))
calc_loocv_rmse(mod_back_bic)
summary(mod_back_bic)$adj.r.squared
# Best Subset
library(leaps)
mod_exhaustive =
  summary(regsubsets

  (Salary ~ . - X2P - X2PA - TRB - PTS , data = players_data, nvmax = 23,really.big=T))
#mod_exhaustive$which
#mod_exhaustive$rss
best_r2_ind = which.max(mod_exhaustive$adjr2)
Adj_R2<-mod_exhaustive$which[best_r2_ind,]
Adj_R2
mod_exh_r2<-
  lm
(Salary ~ Age + G + FG + FGA + FG. + X3P + X3P. + FTA + FT. + AST + BLK,data=players_data)
calc_loocv_rmse(mod_exh_r2)

# Best AIC

p = ncol(mod_exhaustive$which)

mod_aic = n * log(mod_exhaustive$rss / n) + 2 * (2:p)

best_aic_ind = which.min(mod_aic)

mod_exhaustive$which[best_aic_ind, ]
mod_exh_aic<-
  lm(Salary ~ Age + G + FG + FGA + FG. + X3P + X3P. + FT + AST,data=players_data)
calc_loocv_rmse(mod_exh_aic)
n = nrow(players_data)
```

```r
p = ncol(players_data)

mod_bic = n * log(mod_exhaustive$rss / n) + log(n) * (2:p)

#mod_bic

best_bic_ind = which.min(mod_bic)
#best_bic_ind

mod_exhaustive$which[best_bic_ind, ]

mod_exhaust_bic = lm(Salary ~ . - X2P - X2PA - TRB - PTS, data = players_data)
mod_exh_bic<-lm(Salary ~ Age + G + FG + FT,data=players_data)
calc_loocv_rmse(mod_exh_bic)
# Stepwise - AIC
mod_start = lm(Salary ~ 1, data = players_data)
mod_forwd_aic = step(
  mod_start,
  scope = Salary ~ Age + G + GS + MP + FG + FGA + FG. + X3P +
    X3PA + X3P.  + X2P. + eFG. + FT + FTA +
    FT. + ORB + DRB  + AST + STL + BLK + TOV + PF ,
  direction = 'forward')

calc_loocv_rmse(mod_forwd_aic)


# Stepwise - BIC

mod_start = lm(Salary ~ 1, data = players_data)
n=nrow(players_data)
mod_forwd_bic = step(
  mod_start,
  scope = Salary ~ Age + G + GS + MP + FG + FGA + FG. + X3P +
    X3PA + X3P.  + X2P. + eFG. + FT + FTA +
    FT. + ORB + DRB  + AST + STL + BLK + TOV + PF ,
  direction = 'forward',k = log(n))


calc_loocv_rmse(mod_forwd_bic)

#AIC (Forwards selection)

mod_start = lm(Salary ~ 1, data=players_data)
forwd_aic = step(
    mod_start,
    scope = Salary ~ Age + G + GS + MP + FG + FGA + FG. + X3P +
    X3PA + X3P.  + X2P. + eFG. + FT + FTA +
    FT. + ORB + DRB  + AST + STL + BLK + TOV + PF,
    direction = 'forward')

calc_loocv_rmse(forwd_aic)
#BIC (Forward selection)
```

```
mod_start = lm(Salary ~ 1, data=players_data)
n = nrow(players_data)

forwd_bic = step(
    mod_start,
    scope = Salary ~ Age + G + GS + MP + FG + FGA + FG. + X3P +
    X3PA + X3P.  + X2P. + eFG. + FT + FTA +
    FT. + ORB + DRB  + AST + STL + BLK + TOV + PF,
    direction = 'forward',
    k = log(n))

calc_loocv_rmse(forwd_bic)
selected_model<-lm(Salary ~ Age + G + FG + FT, data=players_data)
#summary(selected_model)
rmse(selected_model)
#variable_plts_model
library(readr)
variable_plts_model = lm(Salary ~ FG + G + Age + FT,data = players_data)
library(olsrr)
ols_plot_added_variable(variable_plts_model)
# Additional plots
par(mfrow = c(2, 3))

plot(Salary ~ FG, data = players_data, pch = 20, col = 'darkblue')
plot(Salary ~ G, data = players_data, pch = 20, col = 'darkblue')
plot(Salary ~ FT, data = players_data, pch = 20, col = 'darkblue')
plot(Salary ~ Age, data = players_data, pch = 20, col = 'darkblue')



# BP test

library(lmtest)
bptest(selected_model)

# Residual vs Fitted plot
library(olsrr)
ols_plot_resid_fit(selected_model)

# Normality Violation Test

shapiro.test(resid(selected_model))
ols_plot_resid_qq(selected_model)
which(cooks.distance(selected_model) >
        4 / length(cooks.distance(selected_model)))
influence_rmvd=
  which(cooks.distance(selected_model)<= 4/length(cooks.distance(selected_model)))
non_inflnc_model =
  lm(Salary ~ FG + G + Age + FT,data = players_data,subset = influence_rmvd)
#shapiro.test(resid(non_inflnc_model))
ols_plot_resid_qq(non_inflnc_model)
ols_plot_resid_fit(non_inflnc_model)
transformed<-
```

```
  lm(log(Salary) ~ FG + G + Age + FT, data=players_data,subset = influence_rmvd )
summary(transformed)
ols_plot_added_variable(transformed)
library(dplyr)

players_preds=select(players_data, G ,Age,FG,FT)

library(corrplot)
# NOTE: we pass the output of cor() to corrplot()
corrplot(cor(players_preds),
         method = 'color', order = 'hclust', diag = FALSE,
         addCoef.col = 'black', tl.pos= 'd',cl.pos='r')
players_data$FG_FT =  players_data$FG - players_data$FT
library(dplyr)

players_preds=select(players_data,G,Age,FT,FG_FT)

library(corrplot)
# NOTE: we pass the output of cor() to corrplot()
corrplot(cor(players_preds),
         method = 'color', order = 'hclust', diag = FALSE,
         addCoef.col = 'black', tl.pos= 'd',cl.pos='r')

library(olsrr)
featured_model =
  lm(Salary~FG_FT+ G + Age + FT ,data=players_data,subset=influence_rmvd)
#round(ols_eigen_cindex(featured_model)[,1:2],4)
library(faraway)
vif(featured_model)

summary(featured_model)
rmse(featured_model)

library(MASS)
bc = boxcox(featured_model, plotit = FALSE)

boxcox(featured_model,
       lambda = seq(0.1, 1, by = 0.1), plotit = TRUE)
sqrt_trans_mod<-lm(sqrt(Salary) ~ FG_FT + G + Age + FT,
                   data=players_data,subset=influence_rmvd )
summary(sqrt_trans_mod)
# sum of square totals

SST = sum((players_data$Salary - mean(players_data$Salary))^2)

1 - (sum((players_data$Salary - predict(sqrt_trans_mod)^2)^2) / SST)
bptest(sqrt_trans_mod)
# Check lengths
weights <- 1 / residuals(featured_model)^2
nrow(players_data)  # Length of the dataset
length(weights)  # Length of the weights vector

# Subset the data
```

```r
subset_data <- players_data[1:length(weights), ]

# Fit WLS model
wls_model <- lm(Salary ~ FG_FT + FT + Age + G,
                data = subset_data, weights = weights)
summary(wls_model)
#bptest(wls_model)
rmse(wls_model)

weights <- 1 / residuals(featured_model)^2
nrow(players_data)
length(weights)

subset_data <-
  players_data[1:length(weights), ]
wls_model <-
  lm(Salary ~ FG_FT + FT + Age + G, data = subset_data, weights = weights)
plot(fitted(wls_model), weighted.residuals(wls_model),
     pch = 20, xlab = 'Fitted Value', ylab = 'Weighted Residual')
abline(h=0, lwd=3, col='steelblue')
bptest(sqrt_trans_mod)
dim(players_data)
outlier_test_cutoff = function(model, alpha = 0.05) {
  n = length(resid(model))
  qt(alpha/(2 * n), df = df.residual(model) - 1, lower.tail = FALSE)
}

# vector of indices for observations deemed outliers.
cutoff =
  outlier_test_cutoff(sqrt_trans_mod, alpha = 0.05)

players_data_no_out=
  players_data[which(abs(rstudent(sqrt_trans_mod))<=cutoff),]
ols_plot_resid_fit(sqrt_trans_mod)

# Test Train Split
smp_size <- floor(0.75 * nrow(players_data))

## set the seed to make your partition reproducible
set.seed(123)
train_ind <- sample(seq_len(nrow(players_data)), size = smp_size)

train <- players_data[train_ind, ]
test <- players_data[-train_ind, ]



# Test Train Split

train_model <-
  lm(Salary ~ FG_FT + FT + Age + G, data = train)
weights <- 1 / residuals(train_model)^2
nrow(train)
```

```r
length(weights)
subset_data <- train[1:length(weights), ]
wls_model_train <-
  lm(Salary ~ FG_FT + FT + Age + G, data = subset_data, weights = weights)



#RMSE Function
rmse_wls =
  function(model){
  sqrt(mean((test$Salary - predict(model,test))^2))
}
#rmse_wls(wls_model_train)
###Lasso

library(glmnet)

x_train=model.matrix(Salary ~ .-PTS-TRB, data = train)[, -1]

# the vector of responses
y_train = train$Salary
# set the random seed for reproducibility
set.seed(123)

# fit the lasso using 10-fold cross-validation to determine lambda
mod_lasso = cv.glmnet(x_train, y_train)

plot(mod_lasso)

grid = exp(seq(12, 14.5, length=100))

mod_lasso = cv.glmnet(x_train, y_train, lambda = grid)

plot(mod_lasso)

# lambda.min
mod_lasso$lambda.min

# lambda.1se
mod_lasso$lambda.1se


coef(mod_lasso, s = 'lambda.min')

coef(mod_lasso, s = 'lambda.1se')
lasso_model_1se=
  lm(Salary ~ Age+G+FG+`X2P`+FTA+TOV,data=train)
lasso_model_min=
  lm(Salary ~Age+G+GS+FG+AST+STL+BLK+TOV,data=train)
#summary(lasso_model_min)
#summary(lasso_model_1se)
rmse_wls(lasso_model_1se)
rmse_wls(lasso_model_min)
```

```
rmse_wls(wls_model_train)
```