Final Project: HoopR Dataset

Group 8: Anish Chintamaneni, Miguel Arenas, Reem Ali, Wael Ali

Fowler College of Business, San Diego State University

BA 649-02: Business Analytics - Spring 2024

Dr. Huiyu Qian

April 24, 2024

Part I: Introduction (Miguel)

Background of the Data

In this project we are going to be analyzing the dataset NBA Team Box that can be found in the HoopR data. The NBA Team Box dataset ranges from the 2021 to 2024 NBA seasons. It covers important team statistical data from every game played in that time span, including regular season games, All-Star games, and playoff games. The purpose of this project is to aim to predict the likelihood or probability of NBA teams winning a game. Since team success and team revenue are related to each other. As teams do better they make more money, the more money teams have the more they can spend on players and overall health/well being of employees.

The NBA is a business. It can be broken down into so many different yet important segments of a business. The NBA itself, National Basketball Association, has a commissioner Adam Silver who acts as the CEO of the company. The NBPA, National Basketball Players Association, is the labor union that represents the players of the NBA. Each of the 30 teams in the NBA are owned by an owner or group of owners who have employees. Employees of an NBA team can range from the executive staff, media, finance, health staff, basketball operations, coaches, players, janitors, interns, and so many more. The fans of NBA teams are consumers who buy merchandise, tickets, food, and more. Other businesses are consumers of the NBA as well. Businesses such as restaurants and bars can utilize NBA popularity as a way to get customers, there are sport stores, such as Dicks Sporting Goods, who sell NBA merchandise, sports betting also uses the NBA stats and predictions for their business. Cities can also experience small boosts in their economy during the All Star Games and NBA Playoffs as fans from across the country are flying, staying and buying in these host cities.

Team success is a result of players playing effectively but also of how well an NBA team is managed and/or operating. Ultimately, the winner of a game is which team outscored the other, or which team had more points at the end of game time. So, it can be hypothesized that teams that average high scoring points are more likely to win a game.

Regression

As stated above, we want to predict the likelihood of an NBA team winning a game. Specifically we will be using the team statistics provided in the NBA Team Box dataset in order to determine which variables predict team score best, as the team that has more points in a game is the winner. Our business/research question is how likely are each team statistic able to predict

team score? The DV in our regression model is going to be “team score” as it is a relatively continuous variable. Our IV’s in our regression model will be team statistics such as assists, blocks, field goal percentage, fouls, free throw percentage, steals, and total rebounds. As mentioned it is important to be able to determine which team will score the most points in a game and be the game winner because it increases team success and ultimately team revenues. It can also be important to determine how many points a team scores in a game when making predictions for sports betting. Coaches of NBA teams can also utilize this analysis for their coaching strategies. Our hypothesis for the regression model is that the independent variables that will most likely determine team score are field goal percentage, assists, and free throw percentage as these team statistics are correlated to scoring outcomes in a NBA game. Rebounds is another predictor we hypothesize can impact team score as it can lead to second chance points. We are hypothesizing that teams with higher field goal percentages, free throw percentages, rebounds and assists will have higher team scores and vice versa.

Classification

After completing our regression analysis it will be important to predict the team outcome or winner. Team outcome is a result of team score and how well the team plays statistically. Our business/research question is how can we predict and classify team outcomes based on our predictors? The DV in our classification analysis will be the “team outcome” since it is a binary variable with two outcomes, winner or loser. The IV in our classification analysis will be our predictor variables, team statistics such as assists, field goal percentage, score, rebounds, etc. Our hypothesis is that a team needs to have high scoring, assists, rebounds and field goal percentage to be classified as winners, and vice versa for losers. Once again it is important to be able to classify team winners in today's day in age because sports betting has grown in popularity. Although sports betting can be done by trying to guess stats of players or teams, most average bettors try to predict the team winner as they have a 50/50 chance of winning. Team outcome is obviously important to the teams themselves as they are ultimately trying to make it to and win the NBA Finals to be crowned as champions. Television networks also compete amongst each other to be able to televise teams who are popular and winning more games.

Part II: Data Preparation and EDA (Reem, Wael)

- Data source and summary of the data set

The link of the data source (1 pt)

The data used for our project is hoopR. We will specifically be investigating the nba_team_box data set.

```
if (!requireNamespace('pacman', quietly = TRUE)){
  install.packages('pacman')
 }
 pacman::p_load_current_gh("sportsdataverse/hoopR", dependencies = TRUE, update
= TRUE)
 pacman::p_load(dplyr, zoo, ggimage, gt)
 progressr::with_progress({
  nba_team_box <-
hoopR::load_nba_team_box(2021:hoopR::most_recent_nba_season())
 }) glue::glue("{nrow(nba_team_box)} rows of NBA team boxscore data from
{length(unique(nba_team_box$game_id))} games.")
 dplyr::glimpse(nba_team_box)
```

https://hoopr.sportsdataverse.org/

# of observations (1 pt) and what each observation represents (1 pt)

The nba_team_box data set contains a total of 10,062 observations and 57 variables. For our project, we are interested in discovering whether certain factors contribute to a team winning a game in the 2023 NBA season. The variables we are interested in utilizing are assists, blocks, field_goal_pct, fouls, free throw_pct, steals, total_rebounds, team_score, and binary variable. We will also look to see how these chosen variables affect team_score.

After filtering out the data to our variables of interest, the total number of observations comes out to 2,630 and the total number of variables is 13.

```
summary(nba_team_box)
 new_data <- subset(nba_team_box, select = c(season, team_slug,
team_winner,assists,blocks,field_goal_pct,fouls,free_throw_pct,steals,total_rebounds,team_sco
re,opponent_team_slug))
 new_data$binary_variable <- ifelse(new_data$team_winner==TRUE,1,0)
 new_data <- new_data [c(4969:7598),]
```

Each observation contains quantitative data such as season, assists, blocks, field goal percentage, fouls, free throw percentage, steals, total rebounds, & team score. Additionally, each observation also contains binary data designated as “team_winner”. A binary variable was created in-order to represent the variables graphically (0 for False, 1 for True). Lastly, each observation also contains categorical data to represent the team of interest (designated as “team_slug & opponent_team_slug” as well as the season of interest.

- **Each variable: variable description (1 pt) and variable type (1 pt)**

Variable	Description	Type
Season	The season in which the game was played.	Categorical Data
Team_slug	The team of interest for a specific game.	Categorical Data
Opponent_team_slug	The opponent team.	Categorical Data
Team_winner	Whether the team of interest lost or won the game (represented by True/False)	Binary Data
Assists	The number of assists the team completed for specified game.	Quantitative Data
Blocks	The number of blocks the team completed for specified game.	Quantitative Data
Field Goal Percentage	The success rate for all shots taken by team of interest for specific game.	Quantitative Data
Fouls	The number of fouls the team completed for specified game.	Quantitative Data
Free Throw Percentage	The success rate for all free throws taken by team of interest for specific game.	Quantitative Data
Steals	The number of steals the team completed for specified game.	Quantitative Data
Total Rebounds	The number of rebound the team completed for specified game.	Quantitative Data
Team Score	The total points the team completed for specified game.	Quantitative Data
Binary Variable	Whether the team of interest lost or won the game (represented by 0 or 1, 0 for False, 1 for True. This binary variable was created for visualization purposes).	Binary Data

- **EDA (univariate, bivariate, numerical and graphical, with outputs and comments on the outputs)**

Univariate: the outputs (2 pts)? comments on how the data was distributed (1 pt)? any missing values (0.5 pt)? any extreme outliers (0.5 pt)?

We will first begin by performing summary statistics for all quantitative variables.

```
mean(new_data$assists)
mean(new_data$blocks)
 mean(new_data$field_goal_pct)
 mean(new_data$fouls)
 mean(new_data$free_throw_pct)
 mean(new_data$steals)
 mean(new_data$total_rebounds)
 mean(new_data$team_score)
 sd(new_data$assists)
 sd(new_data$blocks)
 sd(new_data$field_goal_pct)
 sd(new_data$fouls)
 sd(new_data$free_throw_pct)
 sd(new_data$steals)
 sd(new_data$total_rebounds)
 sd(new_data$team_score)
 range(new_data$assists)
 range(new_data$blocks)
 range(new_data$field_goal_pct)
 range(new_data$fouls)
 range(new_data$free_throw_pct)
 range(new_data$steals)
 range(new_data$total_rebounds)
 range(new_data$team_score)
```

```
> mean(new_data$assists)

[1] 25.2308

> mean(new_data$blocks)

[1] 4.659316

> mean(new_data$field_goal_pct)

[1] 47.58015

> mean(new_data$fouls)

[1] 19.9403

> mean(new_data$free_throw_pct)

[1] 78.25449

> mean(new_data$steals)

[1] 7.259696

> mean(new_data$total_rebounds)

[1] 43.41825

> mean(new_data$team_score)

[1] 114.4095

> sd(new_data$assists)

[1] 4.928508

> sd(new_data$blocks)

[1] 2.458228

> sd(new_data$field_goal_pct)
```

```
[1] 5.475909

> sd(new_data$fouls)

[1] 4.112049

> sd(new_data$free_throw_pct)

[1] 9.839912

> sd(new_data$steals)

[1] 2.871909

> sd(new_data$total_rebounds)

[1] 6.768461

> sd(new_data$team_score)

[1] 12.19597

> range(new_data$assists)

[1] 12 49

> range(new_data$blocks)

[1]  0 19

> range(new_data$field_goal_pct)

[1] 30.2 65.5

> range(new_data$fouls)

[1]  2 35

> range(new_data$free_throw_pct)

[1]   0 100
```

> range(new_data$steals)

[1] 0 18

> range(new_data$total_rebounds)

[1] 23 73

> range(new_data$team_score)

[1] 79 184

hist(new_data$assists)

hist(new_data$blocks)

hist(new_data$field_goal_pct)

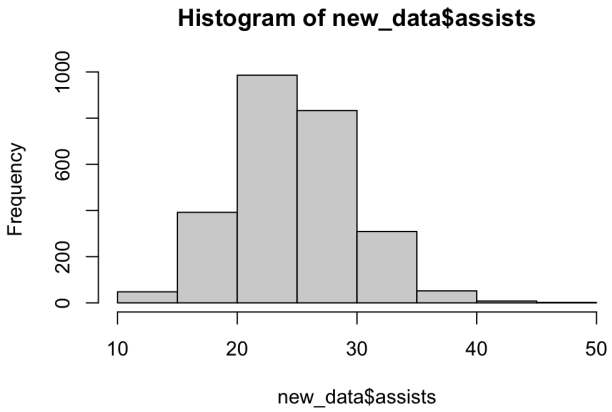
hist(new_data$fouls)

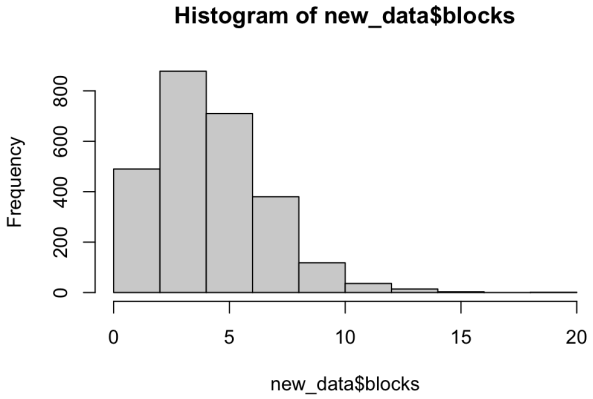
hist(new_data$free_throw_pct)

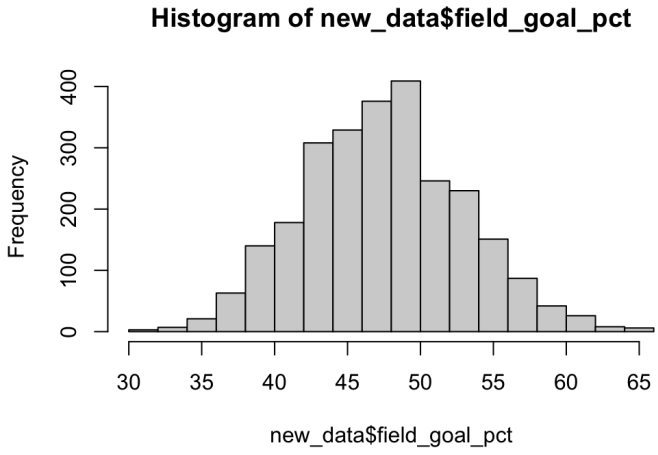
hist(new_data$steals)

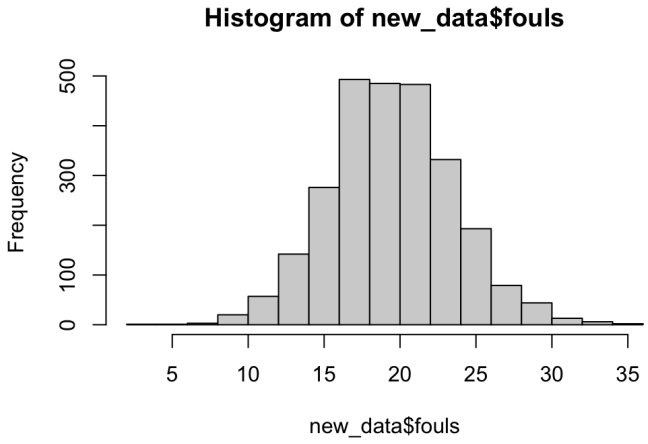
hist(new_data$total_rebounds)

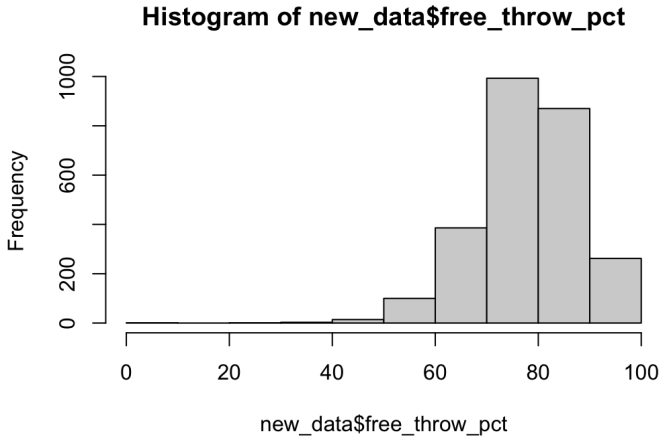
hist(new_data$team_score)

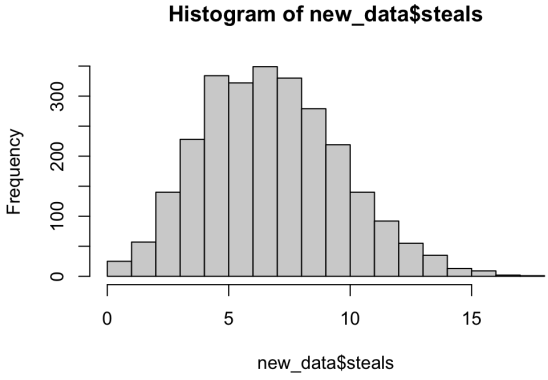

Histogram of new_data$assists
Frequency
0
200
600
1000
10
20
30
40
50
new_data$assists

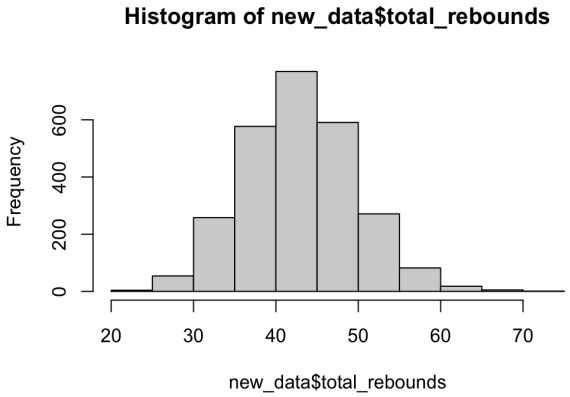

Histogram of new_data$blocks
Frequency
0
200
400
600
800
0
5
10
15
20
new_data$blocks

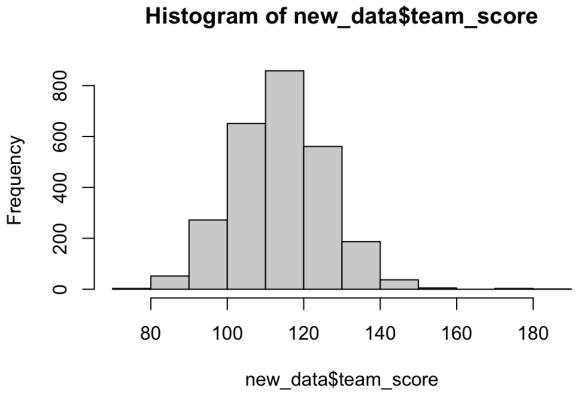

Histogram of new_data$field_goal_pct
Frequency
0
100
200
300
400
30
35
40
45
50
55
60
65
new_data$field_goal_pct


Histogram of new_data$fouls
Frequency
0
100
300
500
5
10
15
20
25
30
35
new_data$fouls


Histogram of new_data$free_throw_pct
Frequency
0
200
600
1000
0
20
40
60
80
100
new_data$free_throw_pct

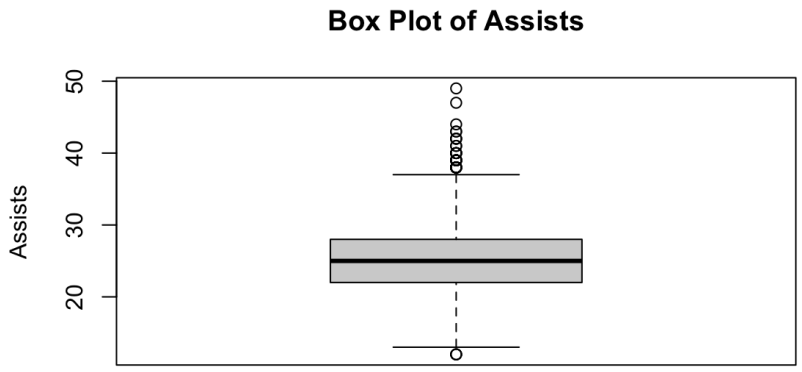

Histogram of new_data$steals
Frequency
0
100
200
300
0
5
10
15
new_data$steals



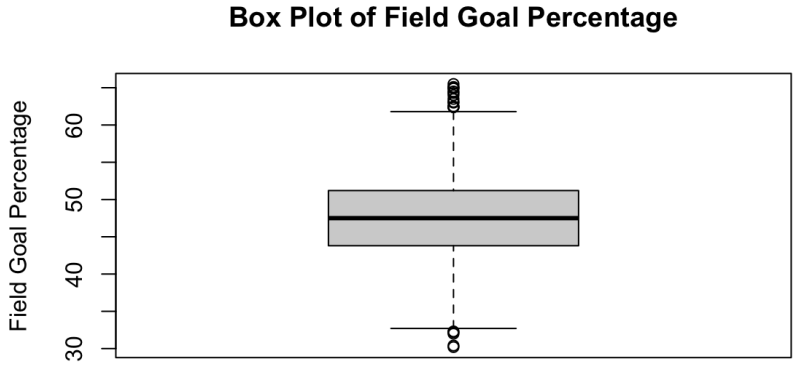



```
boxplot(new_data$assists, main = "Box Plot of Assists", ylab = "Assists")

boxplot(new_data$field_goal_pct, main = "Box Plot of Field Goal Percentage", ylab = "Field Goal Percentage")

boxplot(new_data$team_score, main = "Box Plot of Team Score", ylab = "team_score")

boxplot(new_data$fouls, main = "Box Plot of Fouls", ylab = "Fouls")

boxplot(new_data$free_throw_pct, main = "Box Plot of Free throw Percentage", ylab = "Free Throw Percentage")

boxplot(new_data$total_rebounds, main = "Box Plot of Total Rebounds", ylab = "Total Rebounds")

boxplot(new_data$blocks, main = "Box Plot of Blocks", ylab = "Blocks")
```

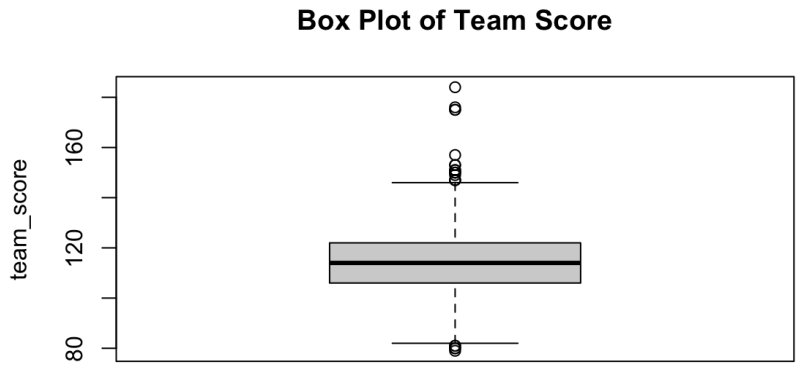
```
boxplot(new_data$steals, main = "Box Plot of Steals", ylab = "Steals")
```



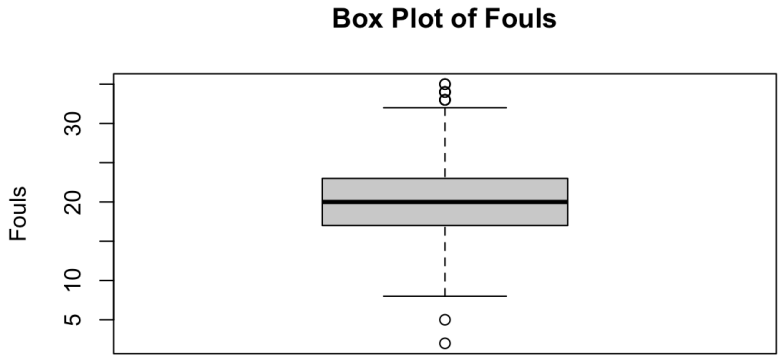
- There are no extreme outliers of assists.



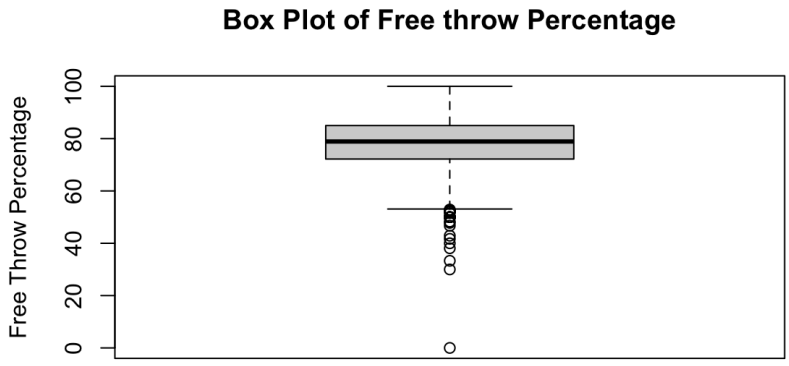
- There are few extreme outliers of field goal percentages.



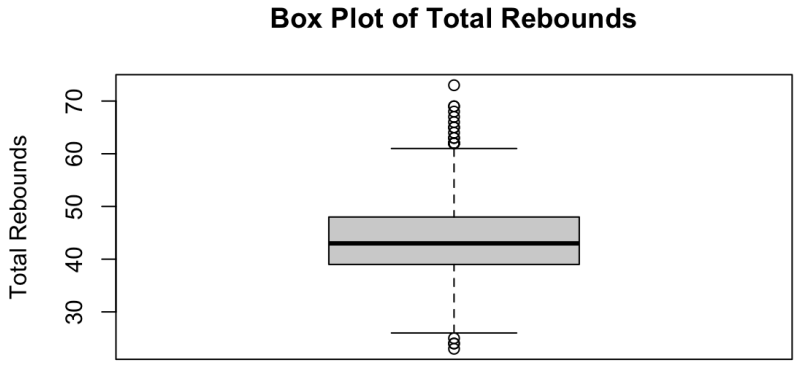
- There are few extreme outliers of team score.



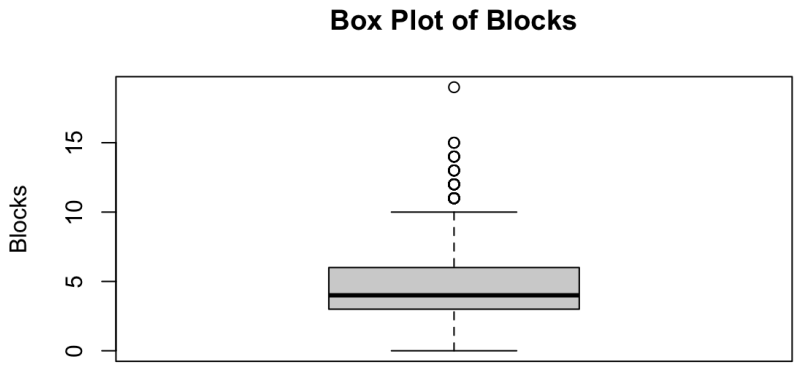
- There are few extreme outliers of fouls.



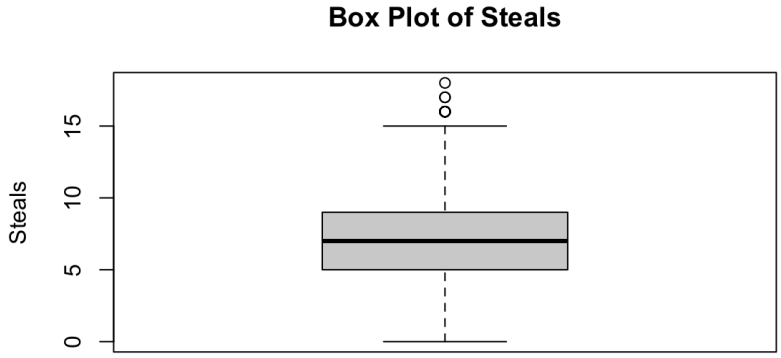
- There is one extreme outlier of free throw percentages.



- There is one extreme outlier of rebounds.



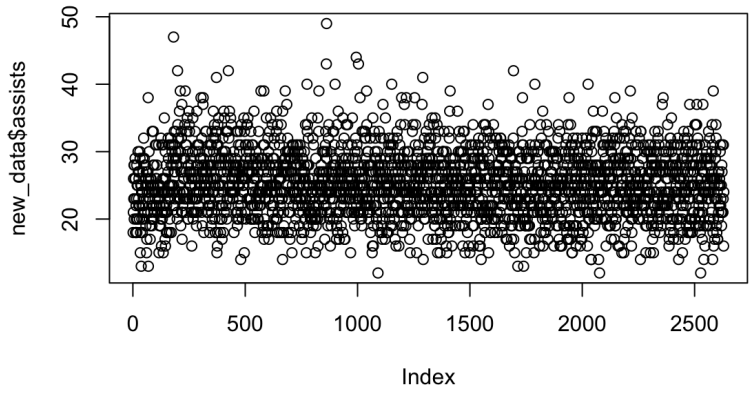
- There is one extreme outlier of blocks.

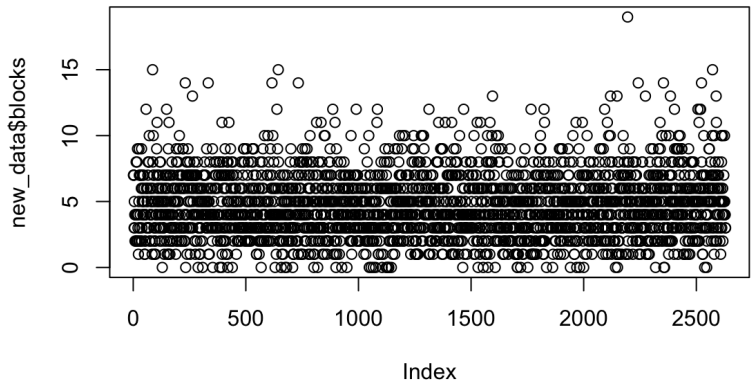


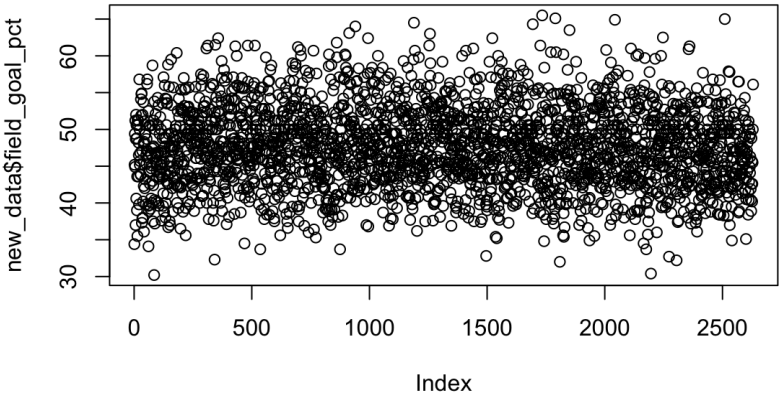
- There are no extreme outliers of steals.

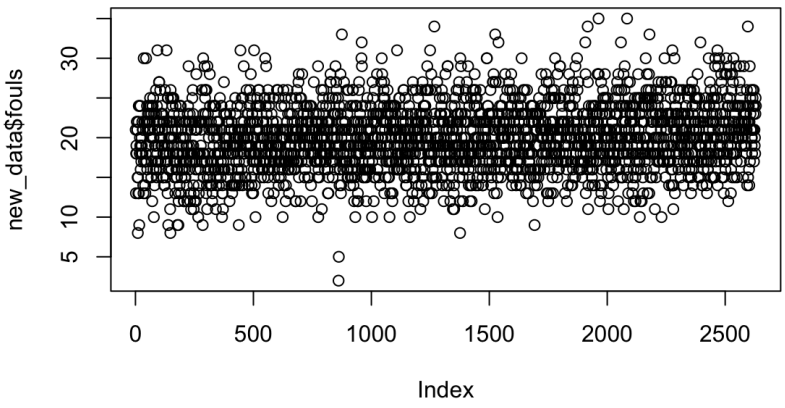
```
plot(new_data$assists)

plot(new_data$blocks)

plot(new_data$field_goal_pct)

plot(new_data$fouls)
```

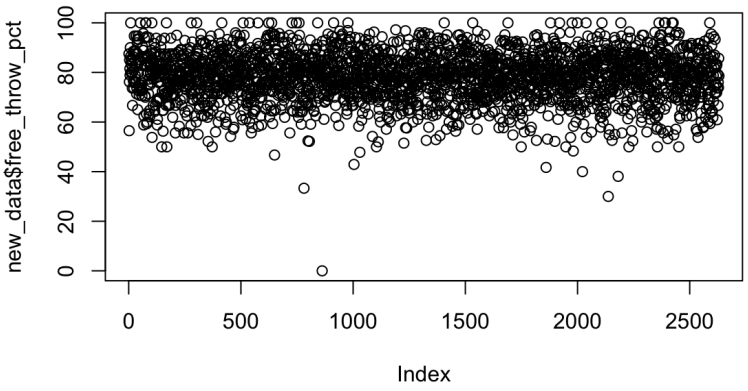
```
plot(new_data$free_throw_pct)

plot(new_data$steals)

plot(new_data$total_rebounds)

plot(new_data$team_score)
```

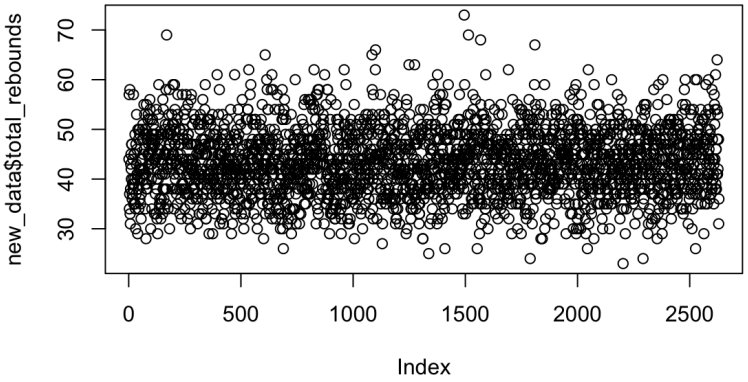


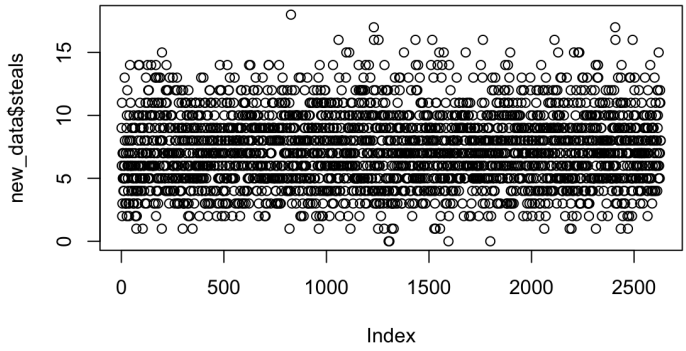


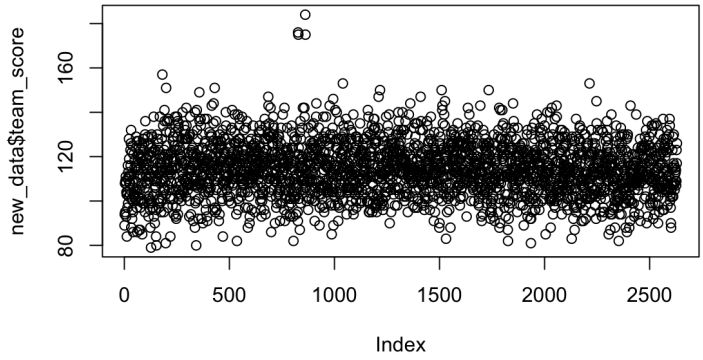

new_data$field_goal_pct
30
40
50
60
0
500
1000
1500
2000
2500
Index


new_data$fouls
5
10
20
30
0
500
1000
1500
2000
2500
Index


new_data$free_throw_pct
0
20
40
60
80
100
0
500
1000
1500
2000
2500
Index


new_data$total_rebounds
30
40
50
60
70
0
500
1000
1500
2000
2500
Index



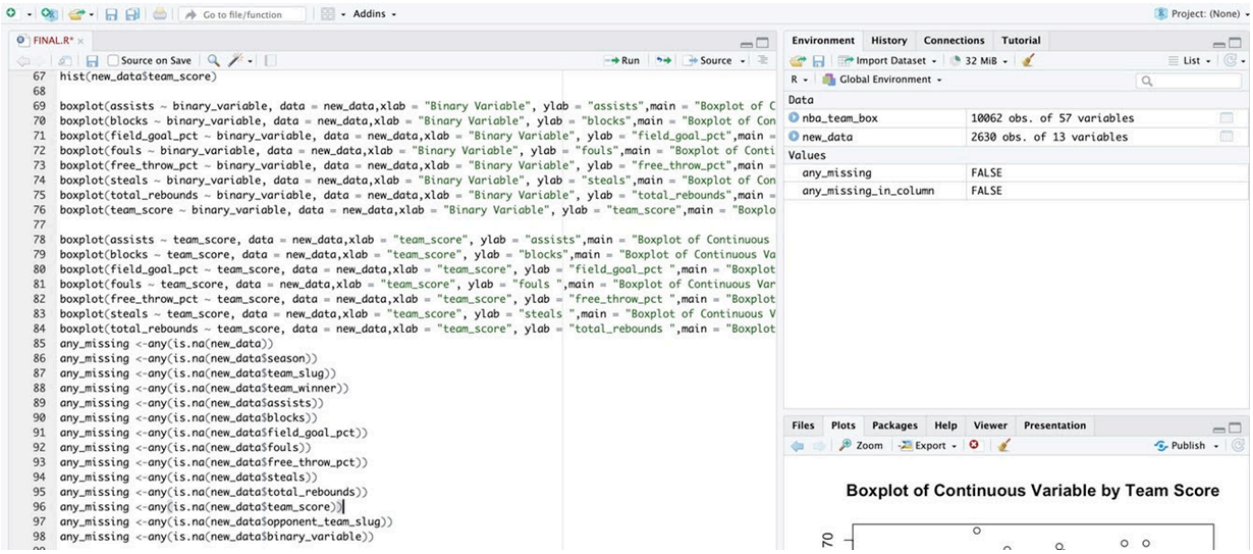


- The histograms and plots above provide a visual representation of the frequency for each selected variable.
- The boxplots represent the median of the data. It also provides a visual representation of data values that appear to be outliers.
- Typically, univariate analysis offers an overview of the summary statistics for selected variables. Moreover, graphical methods offer visualizations that aid in grasping trends and frequencies.

To check and see if there are any missing values in the dataset, the is.na() function was utilized.

```
any_missing <-any(is.na(new_data))
any_missing <-any(is.na(new_data$season))
```

```
any_missing <-any(is.na(new_data$team_slug))
any_missing <-any(is.na(new_data$team_winner))
any_missing <-any(is.na(new_data$assists))
any_missing <-any(is.na(new_data$blocks))
any_missing <-any(is.na(new_data$field_goal_pct))
any_missing <-any(is.na(new_data$fouls))
any_missing <-any(is.na(new_data$free_throw_pct))
any_missing <-any(is.na(new_data$steals))
any_missing <-any(is.na(new_data$total_rebounds))
any_missing <-any(is.na(new_data$team_score))
any_missing <-any(is.na(new_data$opponent_team_slug))
any_missing <-any(is.na(new_data$binary_variable))
```

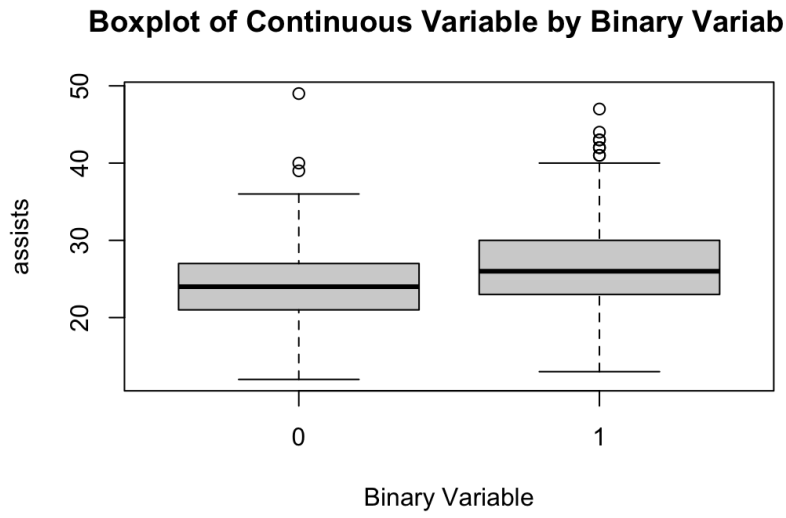


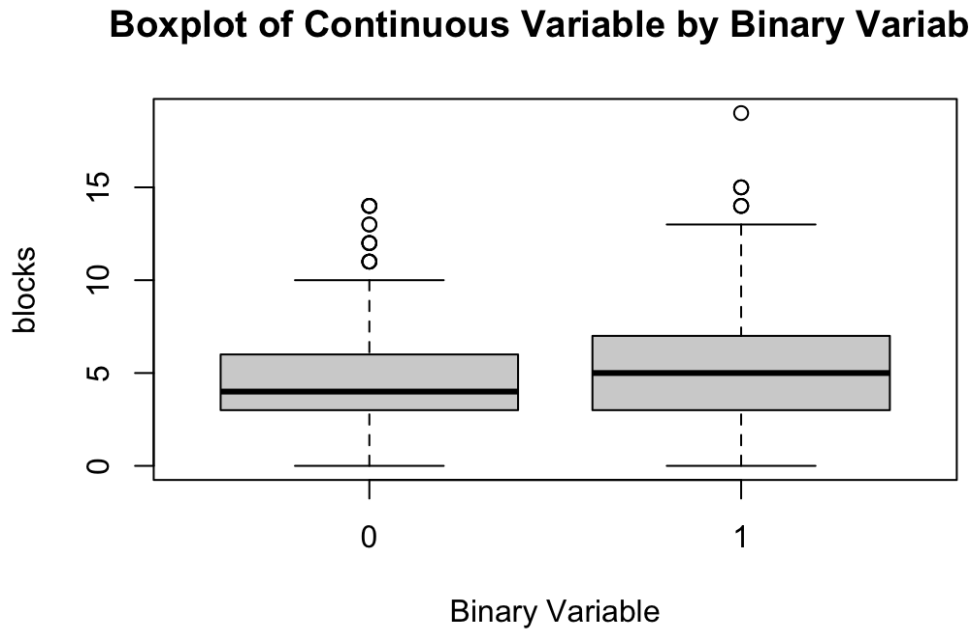
As seen in the image above, a value of “False” was given, meaning there are no missing values in the dataset.

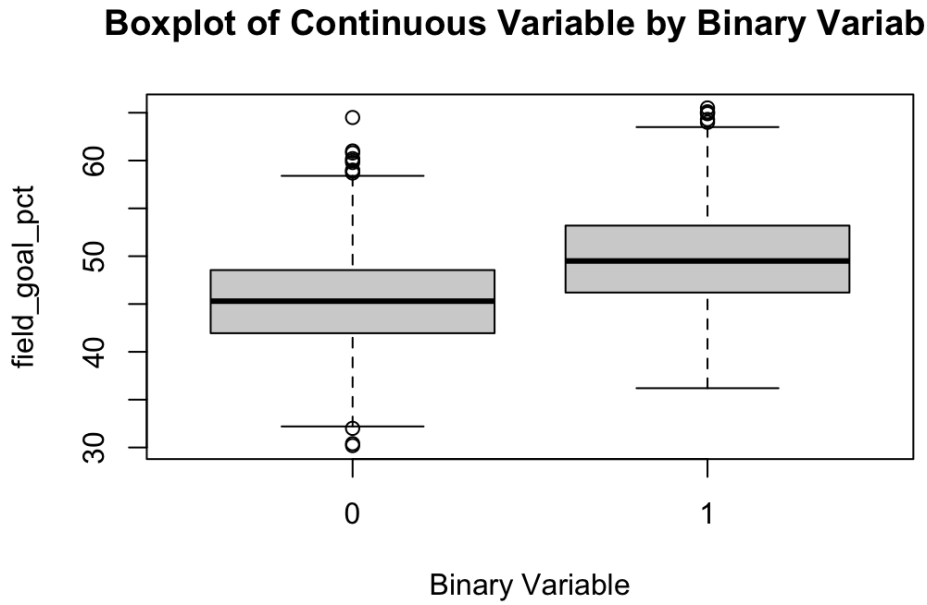
Bivariate: the outputs (3 pts)? comments on how one variable is related to the other variable based on the output for each pair of variables (3 pts)?

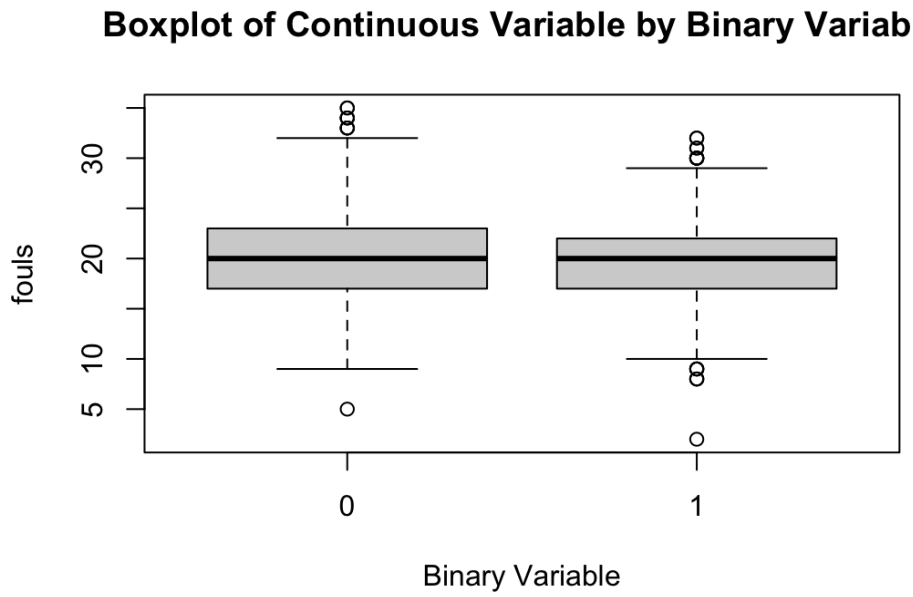
- Boxplots were used to show the correlation between all predictors against the binary variables and team scores.

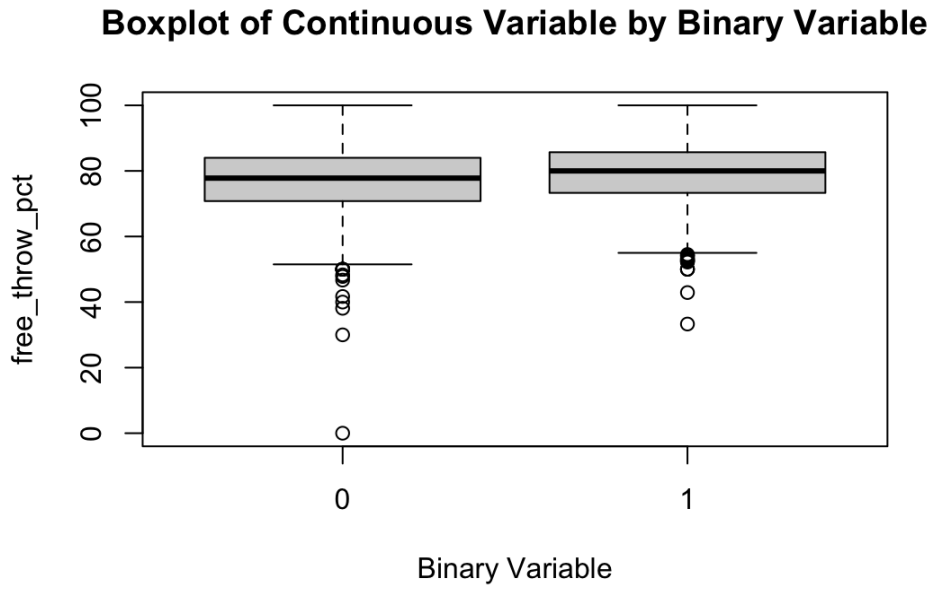
```
boxplot(assists ~ binary_variable, data = new_data,xlab = "Binary Variable", ylab =
"assists",main = "Boxplot of Continuous Variable by Binary Variable")
 boxplot(blocks ~ binary_variable, data = new_data,xlab = "Binary Variable", ylab =
"blocks",main = "Boxplot of Continuous Variable by Binary Variable")
 boxplot(field_goal_pct ~ binary_variable, data = new_data,xlab = "Binary Variable", ylab =
"field_goal_pct",main = "Boxplot of Continuous Variable by Binary Variable")
 boxplot(fouls ~ binary_variable, data = new_data,xlab = "Binary Variable", ylab = "fouls",main
= "Boxplot of Continuous Variable by Binary Variable")
 boxplot(free_throw_pct ~ binary_variable, data = new_data,xlab = "Binary Variable", ylab =
"free_throw_pct",main = "Boxplot of Continuous Variable by Binary Variable")
 boxplot(steals ~ binary_variable, data = new_data,xlab = "Binary Variable", ylab = "steals",main
= "Boxplot of Continuous Variable by Binary Variable")
 boxplot(total_rebounds ~ binary_variable, data = new_data,xlab = "Binary Variable", ylab =
"total_rebounds",main = "Boxplot of Continuous Variable by Binary Variable")
 boxplot(team_score ~ binary_variable, data = new_data,xlab = "Binary Variable", ylab =
"team_score",main = "Boxplot of Continuous Variable by Binary Variable")
```

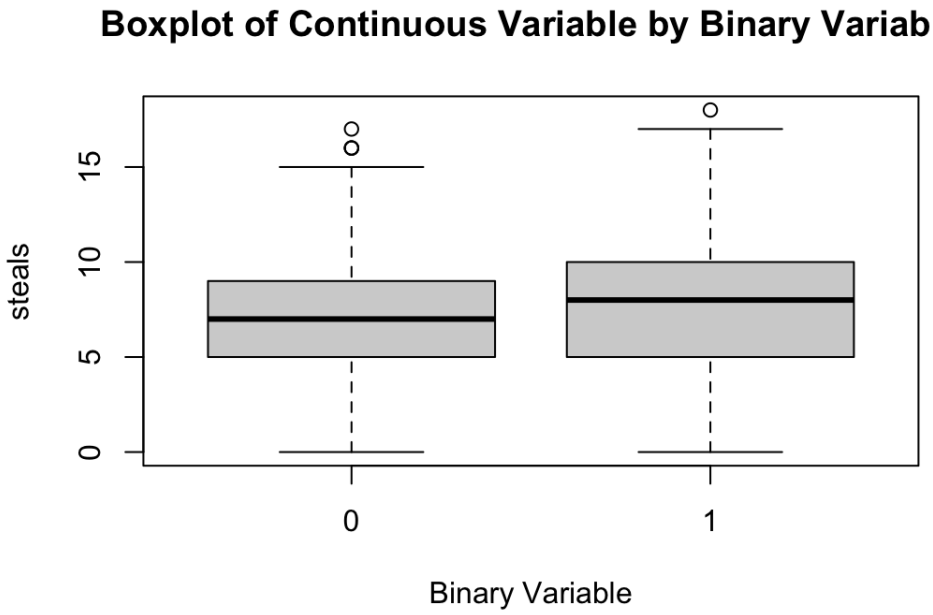


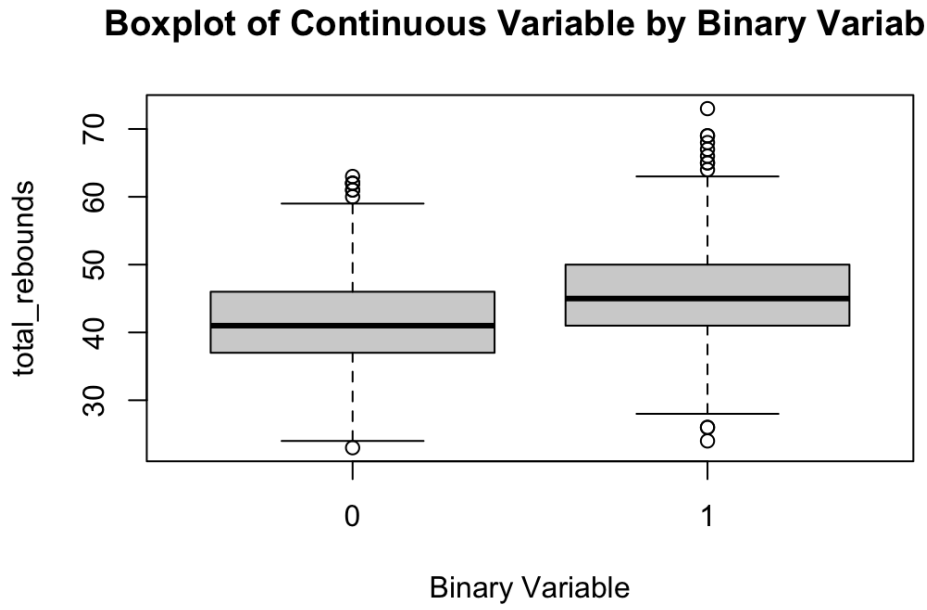

Boxplot of Continuous Variable by Binary Variable
blocks
0
5
10
15
0
1
Binary Variable

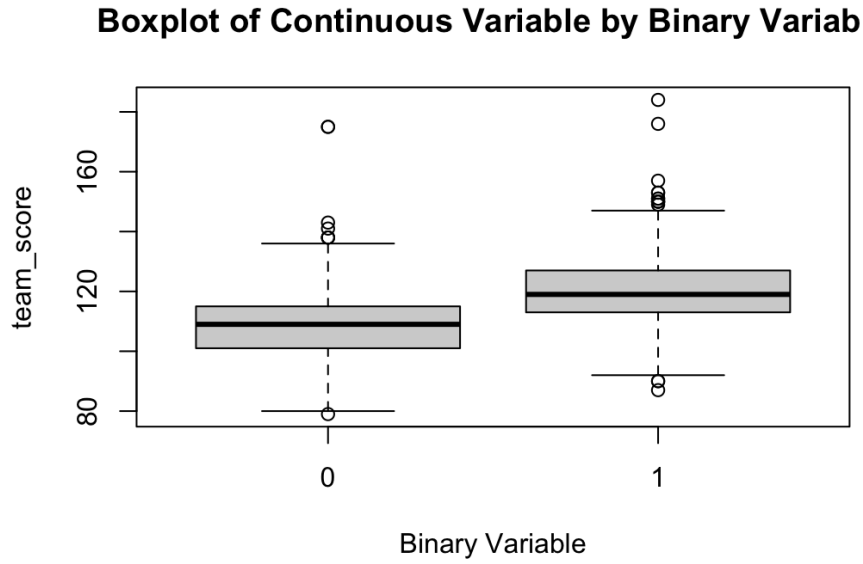

Boxplot of Continuous Variable by Binary Variable
field_goal_pct
30
40
50
60
0
1
Binary Variable


Boxplot of Continuous Variable by Binary Variable
fouls
5
10
20
30
0
1
Binary Variable


Boxplot of Continuous Variable by Binary Variable
free_throw_pct
0
20
40
60
80
100
0
1
Binary Variable


Boxplot of Continuous Variable by Binary Variable
steals
0
5
10
15
0
1
Binary Variable

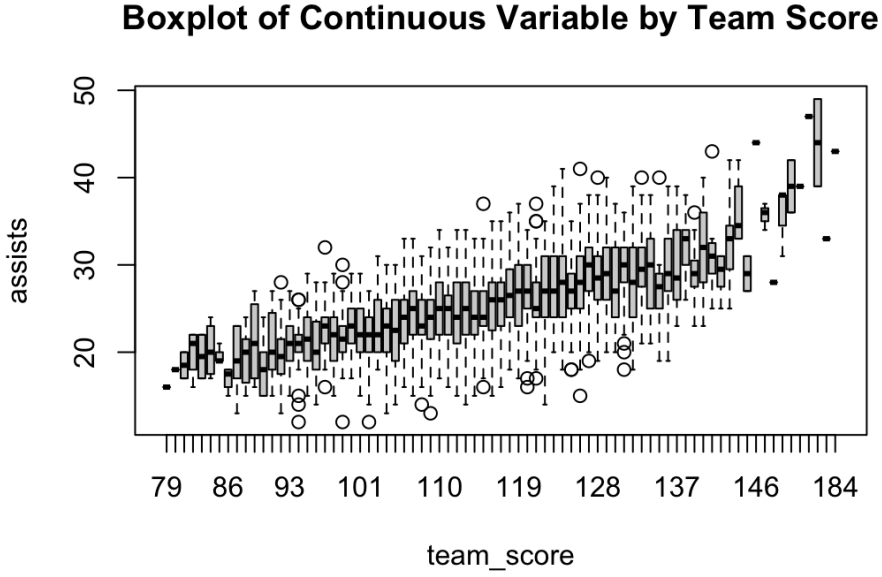


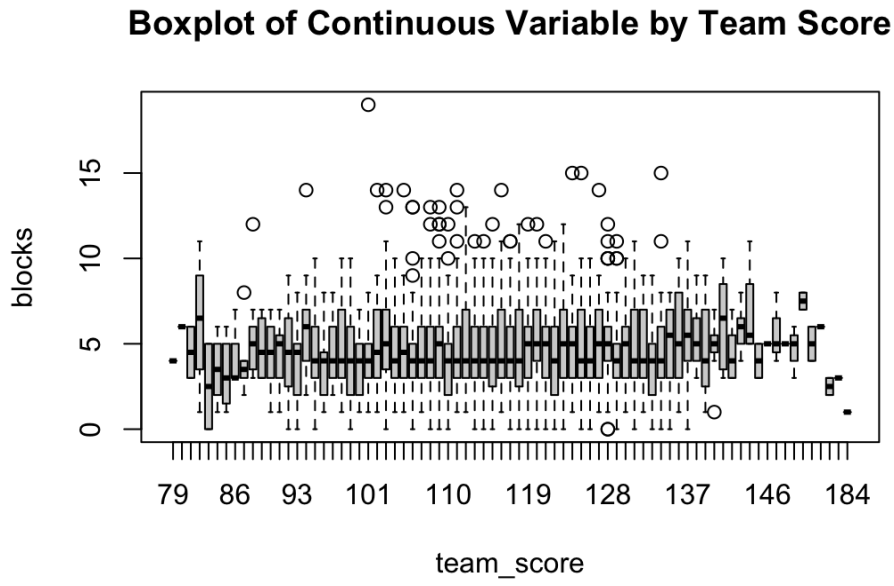


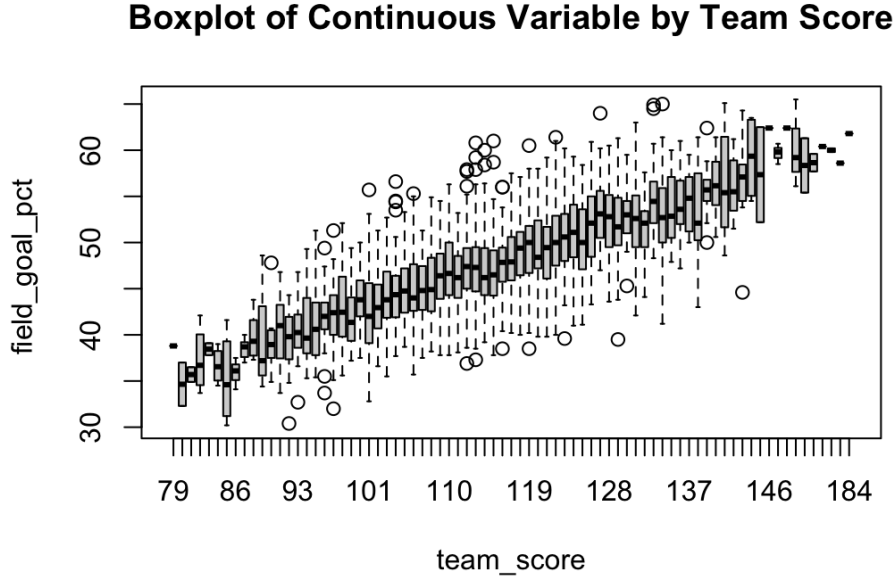
boxplot(assists ~ team_score, data = new_data,xlab = "team_score", ylab = "assists",main = "Boxplot of Continuous Variable by Team Score")

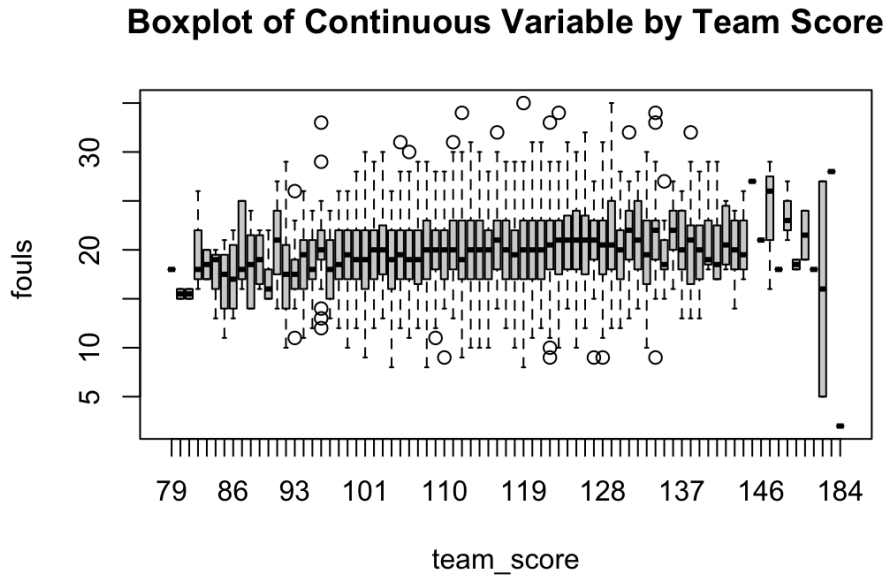
boxplot(blocks ~ team_score, data = new_data,xlab = "team_score", ylab = "blocks",main = "Boxplot of Continuous Variable by Team Score")

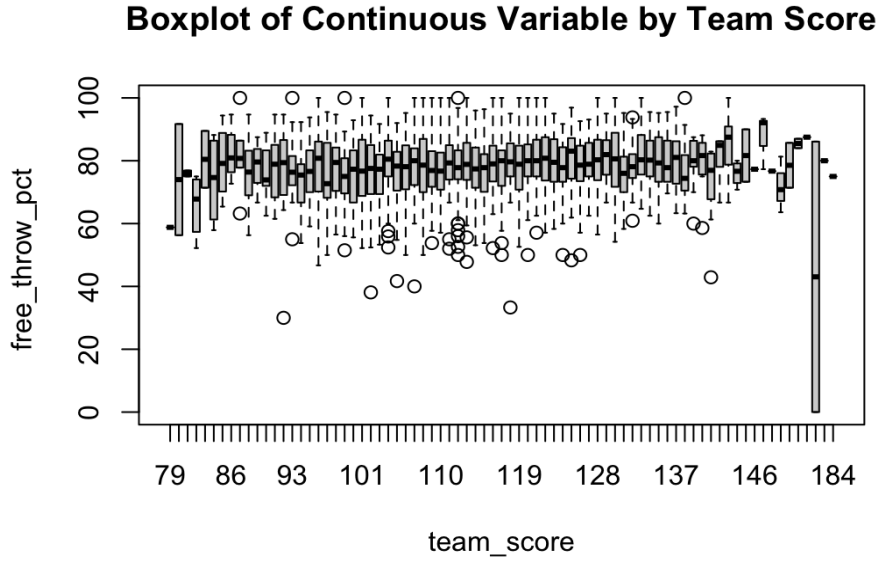
```
boxplot(field_goal_pct ~ team_score, data = new_data,xlab = "team_score", ylab =
"field_goal_pct ",main = "Boxplot of Continuous Variable by Team Score")
boxplot(fouls ~ team_score, data = new_data,xlab = "team_score", ylab = "fouls ",main =
"Boxplot of Continuous Variable by Team Score")
boxplot(free_throw_pct ~ team_score, data = new_data,xlab = "team_score", ylab =
"free_throw_pct ",main = "Boxplot of Continuous Variable by Team Score")
boxplot(steals ~ team_score, data = new_data,xlab = "team_score", ylab = "steals ",main =
"Boxplot of Continuous Variable by Team Score")
boxplot(total_rebounds ~ team_score, data = new_data,xlab = "team_score", ylab =
"total_rebounds ",main = "Boxplot of Continuous Variable by Team Score")
```

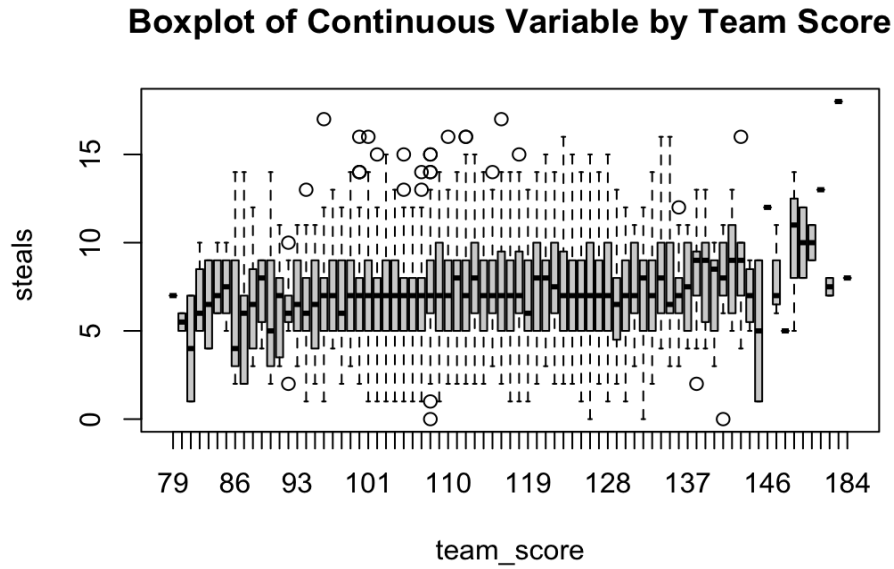


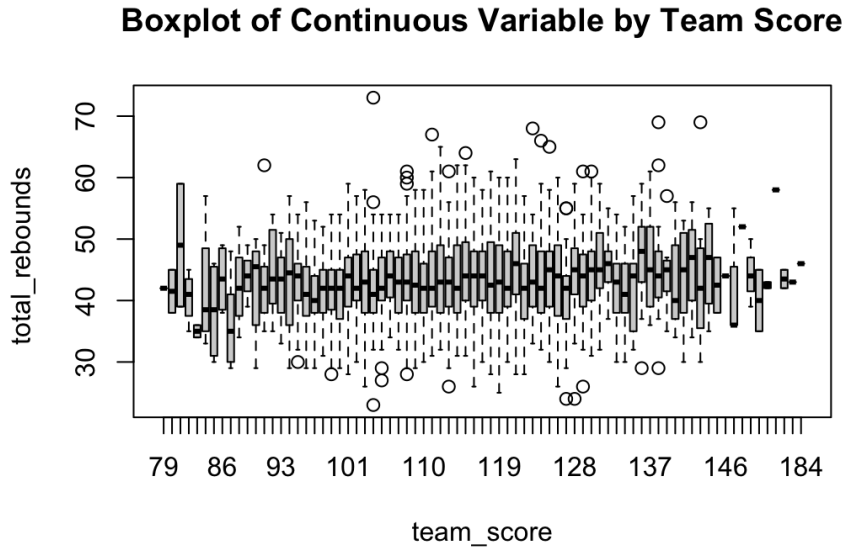

Boxplot of Continuous Variable by Team Score
blocks
0
5
10
15
79
86
93
101
110
119
128
137
146
184
team_score


Boxplot of Continuous Variable by Team Score
field_goal_pct
30
40
50
60
79
86
93
101
110
119
128
137
146
184
team_score


Boxplot of Continuous Variable by Team Score
fouls
5
10
20
30
79
86
93
101
110
119
128
137
146
184
team_score

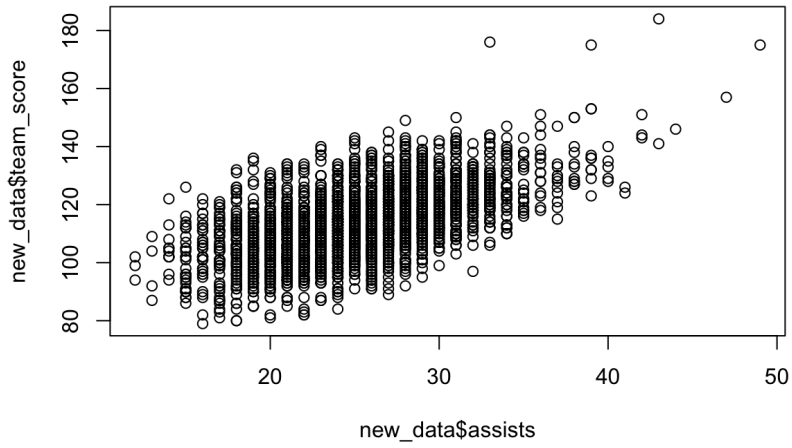

Boxplot of Continuous Variable by Team Score
free_throw_pct
0
20
40
60
80
100
79
86
93
101
110
119
128
137
146
184
team_score

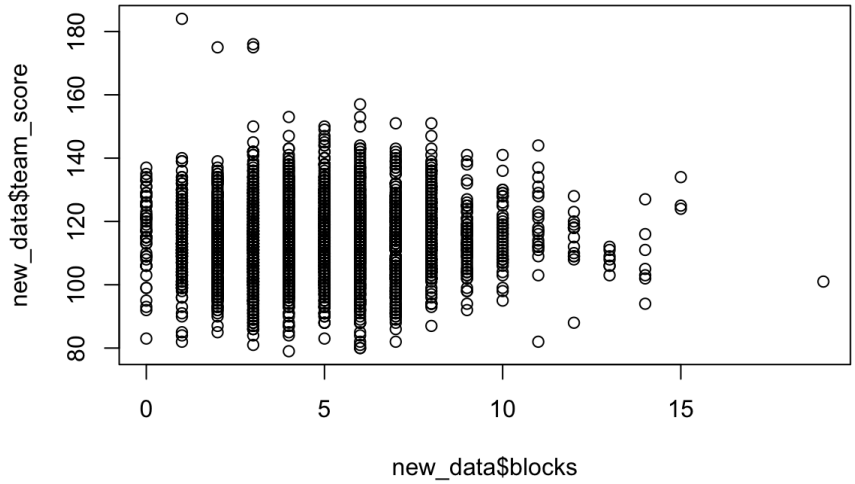


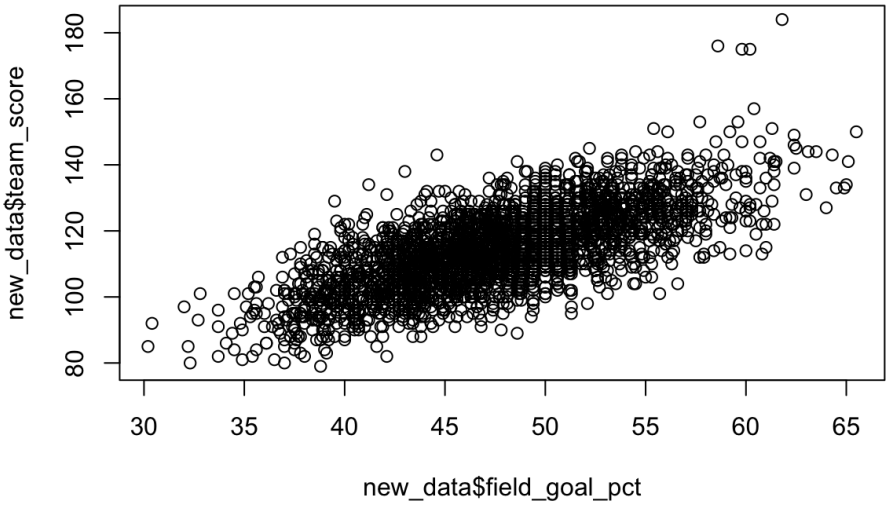


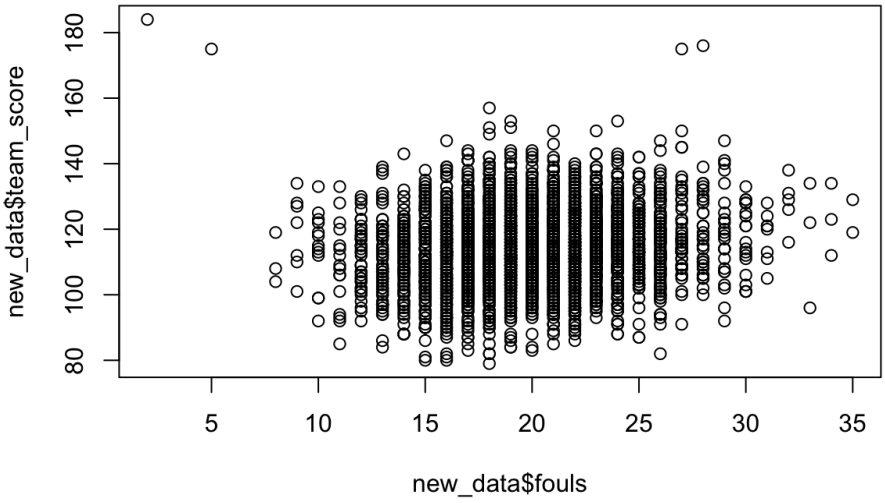
- Based on the prior visualizations, it shows that a clear significance exists between team score and field gaol percentage and between the team score and the assists.
- More visualizations were used such as plots to better interpret the predictors correlations.

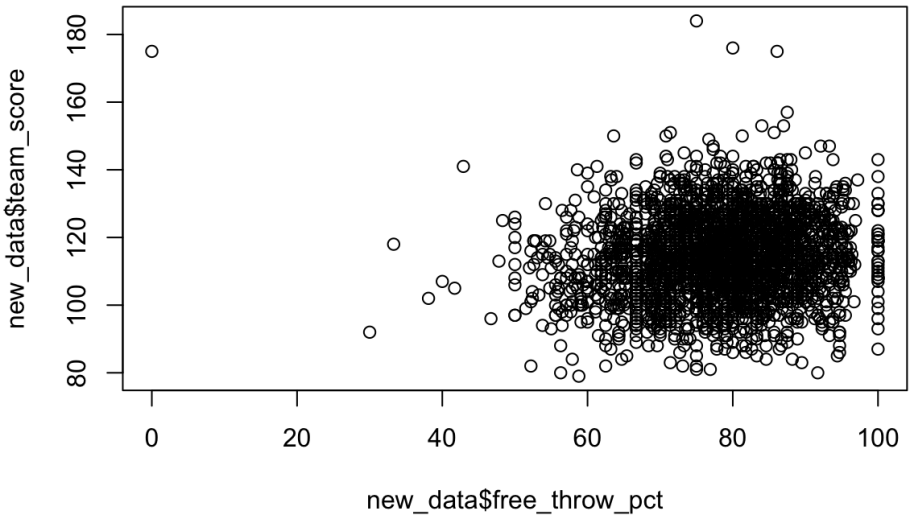
```
plot(new_data$assists,new_data$team_score)
plot(new_data$blocks,new_data$team_score)
plot(new_data$field_goal_pct,new_data$team_score)
plot(new_data$fouls,new_data$team_score)
plot(new_data$free_throw_pct,new_data$team_score)
plot(new_data$total_rebounds,new_data$team_score)
plot(new_data$field_goal_pct,new_data$assists)
```

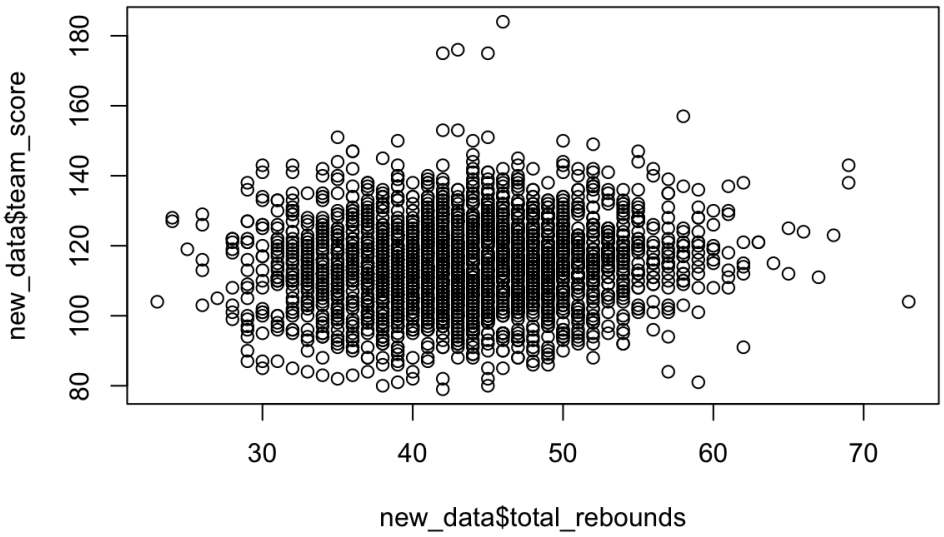




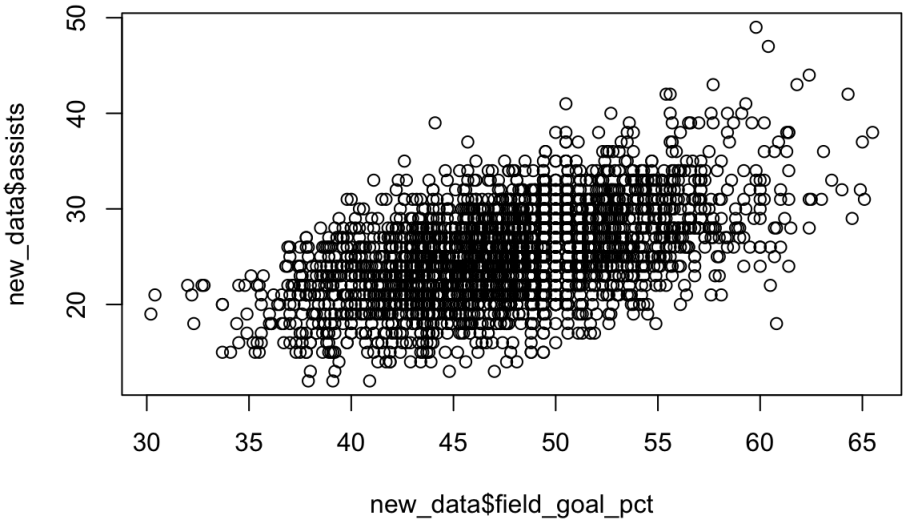

new_data$team_score
80
100
120
140
160
180
30
35
40
45
50
55
60
65
new_data$field_goal_pct


new_data$team_score
80
100
120
140
160
180
5
10
15
20
25
30
35
new_data$fouls





- Since the plots already have shown a significance towards the predictors assists and field goal percentage against the team score, I also ran a correlation plot to show the correlation between the predictors field goal percentage and assists against each other which also turned out t have significant correlation.



- **data cleaning/preparation based on the above outputs/findings to deal with missing values, outliers, transformed variables, interaction terms etc. (5 pts)**

- Many transformations were found for each predictor variable to test out different prediction and potentially establish a higher significance.

Transformed Variables: The log, sqrt, and sq transformations were identified and provided to show a variation in results.

```
# log transformation and Linear Regression
log_assists <- log(new_data$assists)
log_blocks <- log(new_data$blocks)
log_steals <- log(new_data$steals)
log_total_rebounds <- log(new_data$total_rebounds)
log_field_goal_pct <- log(new_data$field_goal_pct)
log_fouls <- log(new_data$fouls)
log_free_throw_pct <- log(new_data$free_throw_pct)
log_season <- log(new_data$season)
```

```
#sqrt transformation and linear regression
sqrt_assists <- sqrt(new_data$assists)
sqrt_blocks <- sqrt(new_data$blocks)
sqrt_steals <- sqrt(new_data$steals)
sqrt_total_rebounds <- sqrt(new_data$total_rebounds)
sqrt_field_goal_pct <- sqrt(new_data$field_goal_pct)
sqrt_fouls <- sqrt(new_data$fouls)
sqrt_free_throw_pct <- sqrt(new_data$free_throw_pct)
sqrt_season <- sqrt(new_data$season)


#sq transformation and linear regression
sq_assists <- new_data$assists^2
sq_blocks <- new_data$blocks^2
sq_steals <- new_data$steals^2
sq_total_rebounds <- new_data$total_rebounds^2
sq_field_goal_pct <- new_data$field_goal_pct^2
sq_fouls <- new_data$fouls^2
sq_free_throw_pct <- new_data$free_throw_pct^2
```

The transformations were used in the following models:

- ```
  model1 <- lm(team_score ~ log_assists + field_goal_pct + fouls + sq_free_throw_pct +
  steals + sq_total_rebounds , data = new_data)
   summary(model1)
  ```

- ```
  model2 <- lm(new_data$team_score  ~ assists + field_goal_pct , data = new_data)
  summary(model2)
  ```

- model3 <- lm(new_data$team_score ~ free_throw_pct+ sq_field_goal_pct + sqrt_fouls, data = new_data)
  summary(model3)

- model4 <- lm(new_data$field_goal_pct ~ assists + team_score , data = new_data)
  summary(model4)

—————> Output

```
> model1 <- lm(team_score ~ log_assists + field_goal_pct + fouls + sq_free_throw_pct + steals
+ sq_total_rebounds , data = new_data)
> summary(model1)

Call:
lm(formula = team_score ~ log_assists + field_goal_pct + fouls +
    sq_free_throw_pct + steals + sq_total_rebounds, data = new_data)

Residuals:
   Min      1Q  Median      3Q     Max
-28.825  -4.856  -0.015   4.752  47.683

Coefficients:
                  Estimate Std. Error
(Intercept)       -2.267e+01  2.588e+00
log_assists        1.232e+01  8.600e-01
field_goal_pct     1.468e+00  3.205e-02
fouls              3.600e-01  3.500e-02
sq_free_throw_pct  1.096e-03  9.668e-05
steals             4.667e-01  5.062e-02
sq_total_rebounds  5.340e-03  2.511e-04
                t value Pr(>|t|)
```

```
(Intercept)         -8.76   <2e-16 ***
log_assists         14.32   <2e-16 ***
field_goal_pct      45.81   <2e-16 ***
fouls               10.28   <2e-16 ***
sq_free_throw_pct   11.34   <2e-16 ***
steals               9.22   <2e-16 ***
sq_total_rebounds   21.26   <2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.372 on 2623 degrees of freedom
Multiple R-squared:  0.6354,  Adjusted R-squared:  0.6346
F-statistic: 761.9 on 6 and 2623 DF,  p-value: < 2.2e-16

>
> model2 <- lm(new_data$team_score  ~ assists + field_goal_pct , data = new_data)
> summary(model2)

Call:
lm(formula = new_data$team_score ~ assists + field_goal_pct,
    data = new_data)

Residuals:
    Min      1Q  Median      3Q     Max
-27.854  -5.516  -0.169   5.293  42.668

Coefficients:
             Estimate Std. Error t value
(Intercept)    38.20035    1.40602   27.17
assists         0.66046    0.03836   17.22
```

```
field_goal_pct  1.25147    0.03452   36.25
             Pr(>|t|)
(Intercept)      <2e-16 ***
assists          <2e-16 ***
field_goal_pct   <2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.222 on 2627 degrees of freedom
Multiple R-squared:  0.5458,  Adjusted R-squared:  0.5455
F-statistic:  1578 on 2 and 2627 DF,  p-value: < 2.2e-16

>
> model3 <- lm(new_data$team_score ~  free_throw_pct+ sq_field_goal_pct + sqrt_fouls, data =
new_data)
> summary(model3)

Call:
lm(formula = new_data$team_score ~ free_throw_pct + sq_field_goal_pct +
    sqrt_fouls, data = new_data)

Residuals:
   Min      1Q  Median      3Q     Max
-27.443  -5.555  -0.186   5.442  56.165

Coefficients:
                  Estimate Std. Error t value
(Intercept)       5.463e+01  2.182e+00  25.033
free_throw_pct    1.289e-01  1.689e-02   7.631
sq_field_goal_pct 1.618e-02  3.153e-04  51.335
```

```
sqrt_fouls        2.830e+00  3.553e-01   7.965
                Pr(>|t|)
(Intercept)       < 2e-16 ***
free_throw_pct    3.25e-14 ***
sq_field_goal_pct  < 2e-16 ***
sqrt_fouls        2.45e-15 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.521 on 2626 degrees of freedom
Multiple R-squared:  0.5124,  Adjusted R-squared:  0.5118
F-statistic: 919.7 on 3 and 2626 DF,  p-value: < 2.2e-16

>
> model4 <- lm(new_data$field_goal_pct ~ assists + team_score , data = new_data)
> summary(model4)

Call:
lm(formula = new_data$field_goal_pct ~ assists + team_score,
    data = new_data)

Residuals:
     Min       1Q   Median       3Q      Max
-12.4936  -2.5352  -0.1191   2.4967  12.9966

Coefficients:
            Estimate Std. Error t value
(Intercept) 11.64272    0.69823   16.68
assists      0.21622    0.01819   11.89
team_score   0.26643    0.00735   36.25
```

```
            Pr(>|t|)
(Intercept)   <2e-16 ***
assists       <2e-16 ***
team_score    <2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.794 on 2627 degrees of freedom
Multiple R-squared:  0.5204,  Adjusted R-squared:   0.52
F-statistic:  1425 on 2 and 2627 DF,  p-value: < 2.2e-16
```

- The more transformations were added to the model, the lower R squared is. Thus, we decided to stick to the natural predictors as they retrieved a higher R squared (63.87%) than any other variation that includes transformations.

```
model <- lm(new_data$team_score ~ assists + fouls + free_throw_pct + steals + total_rebounds
+ blocks + field_goal_pct , data = new_data)
summary(model)

Call:
lm(formula = new_data$team_score ~ assists + fouls + free_throw_pct +
    steals + total_rebounds + blocks + field_goal_pct, data = new_data)

Residuals:
    Min      1Q  Median      3Q     Max
-29.321  -4.853   0.051   4.841  47.376

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -11.36601    2.33961  -4.858 1.26e-06 ***
assists          0.54264    0.03490  15.550  < 2e-16 ***
```

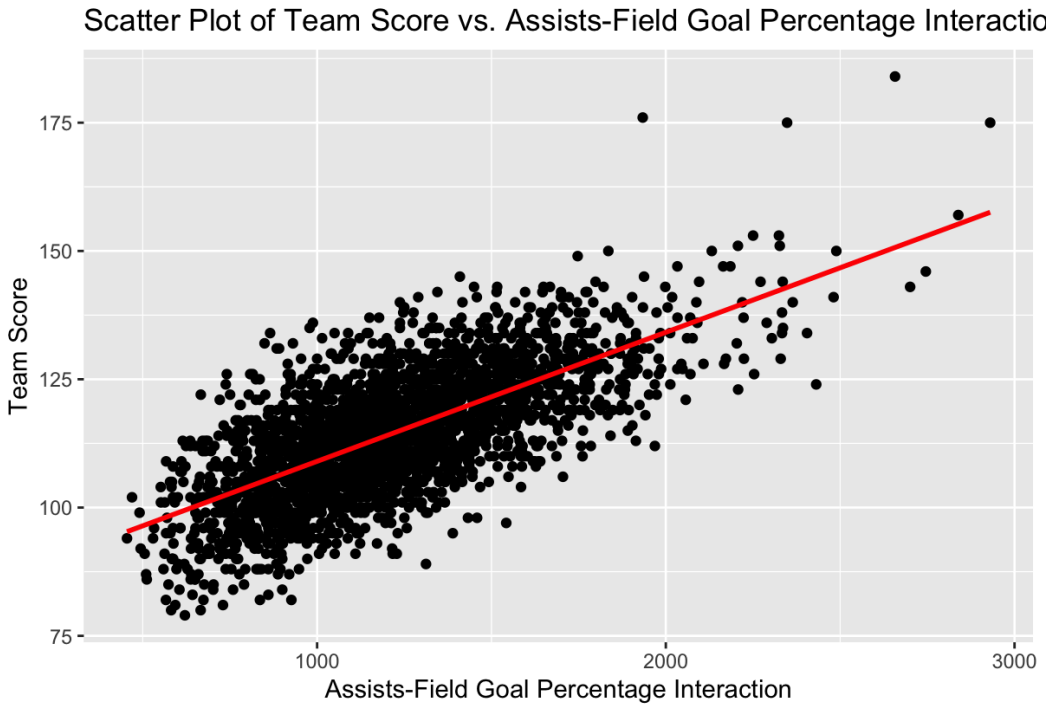
```
fouls           0.36263    0.03490  10.390  < 2e-16 ***
free_throw_pct   0.16622    0.01463  11.365  < 2e-16 ***
steals          0.45592    0.05046   9.036  < 2e-16 ***
total_rebounds   0.47401    0.02268  20.899  < 2e-16 ***
blocks          -0.17061    0.05956  -2.865  0.00421 **
field_goal_pct   1.44494    0.03222  44.850  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.345 on 2622 degrees of freedom
Multiple R-squared:  0.6383,  Adjusted R-squared:  0.6373
F-statistic:   661 on 7 and 2622 DF,  p-value: < 2.2e-16
```

Interaction Terms

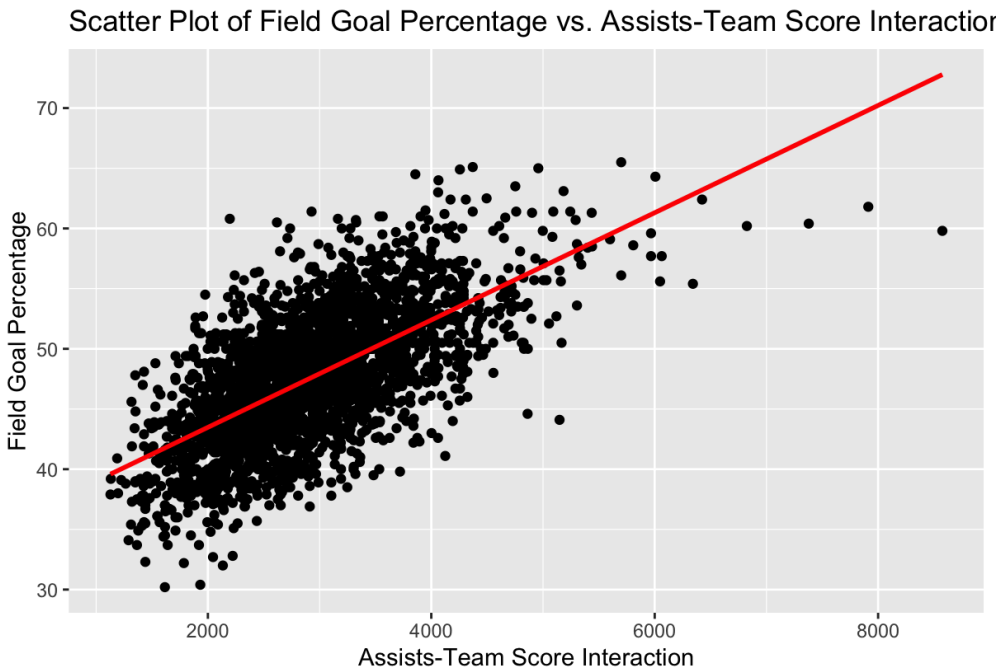
- Since we have established that the main predictors affecting the hypothesis include assists and fiel goal percentages. So, I prioritized using these variables with some other ones to test significance.

```
new_data$assists_field_goal_pct_interaction <- new_data$assists * new_data$field_goal_pct
new_data$assists_team_score_interaction <- new_data$assists * new_data$team_score
new_data$field_goal_pct_team_score_interaction <- new_data$field_goal_pct *
new_data$team_score
new_data$assists_steals_interaction <- new_data$assists * new_data$steals
new_data$field_goal_pct_total_rebounds_interaction <- new_data$field_goal_pct *
new_data$total_rebounds

library(ggplot2)
ggplot(new_data, aes(x = assists_field_goal_pct_interaction, y = team_score)) +
  geom_point() +
  labs(title = "Scatter Plot of Team Score vs. Assists-Field Goal Percentage Interaction",
```

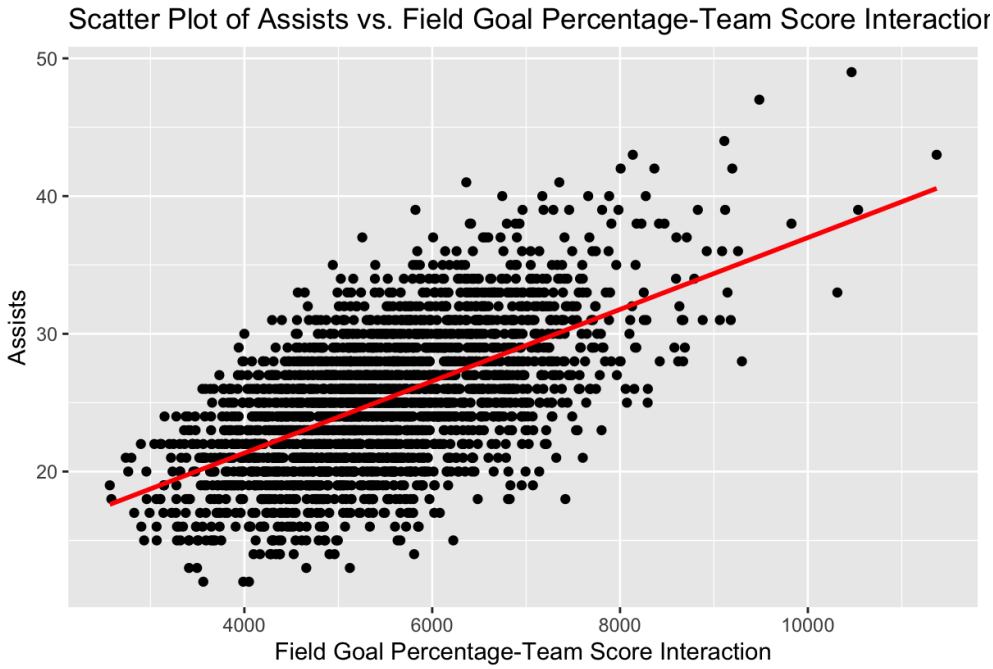
```
    x = "Assists-Field Goal Percentage Interaction",
    y = "Team Score") +
  geom_smooth(method = "lm", color = "Red", se = FALSE)
```



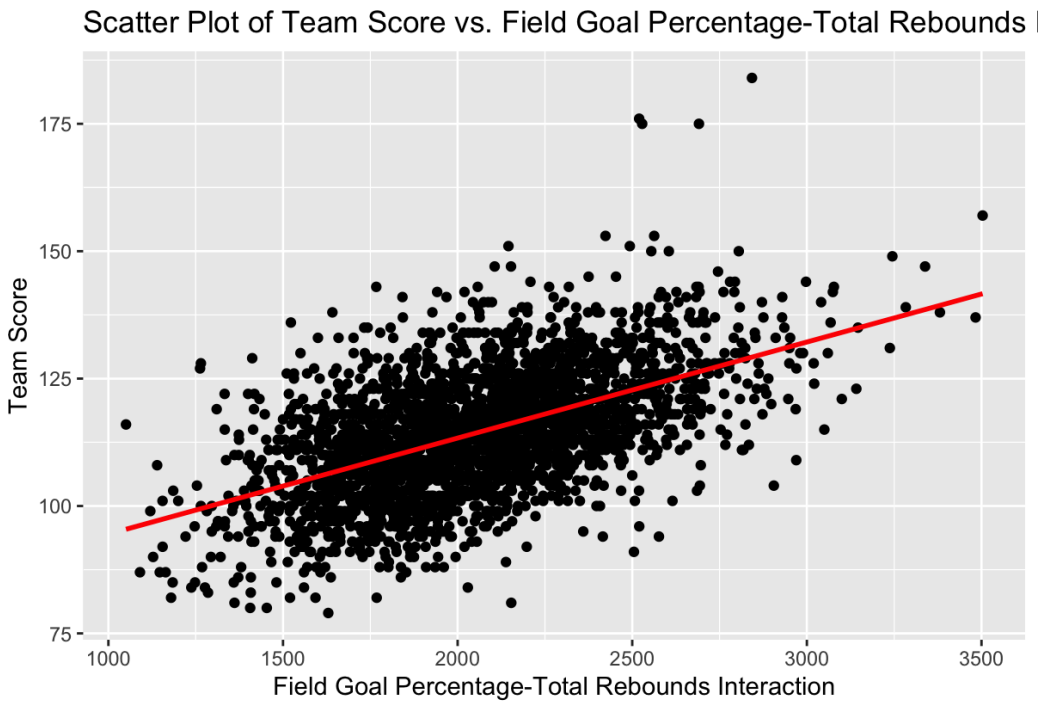

```
ggplot(new_data, aes(x = assists_team_score_interaction, y = field_goal_pct)) +
  geom_point() +
  labs(title = "Scatter Plot of Field Goal Percentage vs. Assists-Team Score Interaction",
    x = "Assists-Team Score Interaction",
    y = "Field Goal Percentage") +
  geom_smooth(method = "lm", color = "Red", se = FALSE)
```




```
ggplot(new_data, aes(x = field_goal_pct_team_score_interaction, y = assists)) +
  geom_point() +
  labs(title = "Scatter Plot of Assists vs. Field Goal Percentage-Team Score Interaction",
       x = "Field Goal Percentage-Team Score Interaction",
       y = "Assists") +
  geom_smooth(method = "lm", color = "Red", se = FALSE)
```




```
ggplot(new_data, aes(x = field_goal_pct_total_rebounds_interaction, y = team_score)) +
  geom_point() +
  labs(title = "Scatter Plot of Team Score vs. Field Goal Percentage-Total Rebounds Interaction",
       x = "Field Goal Percentage-Total Rebounds Interaction",
       y = "Team Score") +
  geom_smooth(method = "lm", color = "Red", se = FALSE)
```



- New models were tested to include the interaction terms.

```
model5 <- lm(team_score ~ assists + blocks + field_goal_pct + fouls + free_throw_pct + steals + total_rebounds + assists_field_goal_pct_interaction +  assists_total_rebounds + assists_team_score_interaction, data = new_data)
print(summary(model5))

model6 <- lm(team_score ~ assists + blocks + field_goal_pct + fouls + free_throw_pct + steals + total_rebounds + assists_field_goal_pct_interaction + assists_team_score_interaction + field_goal_pct_team_score_interaction, data = new_data)
print(summary(model6))
```

—------> Output

```
> model5 <- lm(team_score ~ assists + blocks + field_goal_pct + fouls + free_throw_pct + steals + total_rebounds + assists_field_goal_pct_interaction +  assists_total_rebounds + assists_team_score_interaction, data = new_data)
> print(summary(model5))
```

Call:

```
lm(formula = team_score ~ assists + blocks + field_goal_pct +
    fouls + free_throw_pct + steals + total_rebounds + assists_field_goal_pct_interaction +
    assists_total_rebounds + assists_team_score_interaction,
    data = new_data)

Residuals:
     Min       1Q   Median       3Q      Max
-23.4726  -0.5697   0.1179   0.6926   7.6056

Coefficients:
                                  Estimate
(Intercept)                     -7.7385222
assists                          0.2417232
blocks                          -0.0030920
field_goal_pct                   2.0214137
fouls                            0.0461556
free_throw_pct                   0.0252283
steals                           0.0397777
total_rebounds                   0.5222222
assists_field_goal_pct_interaction -0.0746927
assists_total_rebounds           -0.0192508
assists_team_score_interaction    0.0362444
                                Std. Error
(Intercept)                      1.9464919
assists                          0.0761639
blocks                           0.0137939
field_goal_pct                   0.0291181
fouls                            0.0082372
free_throw_pct                   0.0034455
steals                           0.0118296
total_rebounds                   0.0256842
```

assists_field_goal_pct_interaction 0.0011692
assists_total_rebounds 0.0010048
assists_team_score_interaction 0.0001683

	t value
(Intercept)	-3.976
assists	3.174
blocks	-0.224
field_goal_pct	69.421
fouls	5.603
free_throw_pct	7.322
steals	3.363
total_rebounds	20.332
assists_field_goal_pct_interaction	-63.882
assists_total_rebounds	-19.159
assists_team_score_interaction	215.344

	Pr(>|t|)
(Intercept)	7.21e-05
assists	0.001522
blocks	0.822652
field_goal_pct	< 2e-16
fouls	2.32e-08
free_throw_pct	3.23e-13
steals	0.000783
total_rebounds	< 2e-16
assists_field_goal_pct_interaction	< 2e-16
assists_total_rebounds	< 2e-16
assists_team_score_interaction	< 2e-16

(Intercept)	***
assists	**
blocks	

```
field_goal_pct                  ***
fouls                        ***
free_throw_pct                  ***
steals                       ***
total_rebounds                  ***
assists_field_goal_pct_interaction ***
assists_total_rebounds          ***
assists_team_score_interaction     ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.698 on 2619 degrees of freedom
Multiple R-squared:  0.9807,  Adjusted R-squared:  0.9806
F-statistic: 1.33e+04 on 10 and 2619 DF,  p-value: < 2.2e-16

>
> model6 <- lm(team_score ~ assists + blocks + field_goal_pct + fouls + free_throw_pct + steals
+ total_rebounds + assists_field_goal_pct_interaction + assists_team_score_interaction +
field_goal_pct_team_score_interaction, data = new_data)
> print(summary(model6))

Call:
lm(formula = team_score ~ assists + blocks + field_goal_pct +
    fouls + free_throw_pct + steals + total_rebounds + assists_field_goal_pct_interaction +
    assists_team_score_interaction + field_goal_pct_team_score_interaction,
    data = new_data)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5852  -0.4458   0.1284   0.5589   5.5988
```

Coefficients:

	Estimate
(Intercept)	52.8380602
assists	1.2481725
blocks	0.0016854
field_goal_pct	-0.5754844
fouls	0.0285385
free_throw_pct	0.0092639
steals	0.0250816
total_rebounds	0.0250580
assists_field_goal_pct_interaction	-0.0457149
assists_team_score_interaction	0.0082259
field_goal_pct_team_score_interaction	0.0157357

	Std. Error
(Intercept)	1.1351269
assists	0.0502359
blocks	0.0095381
field_goal_pct	0.0458550
fouls	0.0057012
free_throw_pct	0.0023975
steals	0.0081843
total_rebounds	0.0038977
assists_field_goal_pct_interaction	0.0008829
assists_team_score_interaction	0.0004735
field_goal_pct_team_score_interaction	0.0002613

	t value
(Intercept)	46.548
assists	24.846
blocks	0.177
field_goal_pct	-12.550

```
fouls                                5.006
free_throw_pct                       3.864
steals                               3.065
total_rebounds                       6.429
assists_field_goal_pct_interaction   -51.777
assists_team_score_interaction       17.372
field_goal_pct_team_score_interaction  60.221
                                Pr(>|t|)
(Intercept)                          < 2e-16
assists                              < 2e-16
blocks                               0.859759
field_goal_pct                       < 2e-16
fouls                                5.94e-07
free_throw_pct                       0.000114
steals                               0.002202
total_rebounds                       1.52e-10
assists_field_goal_pct_interaction   < 2e-16
assists_team_score_interaction       < 2e-16
field_goal_pct_team_score_interaction  < 2e-16

(Intercept)                          ***
assists                              ***
blocks
field_goal_pct                       ***
fouls                                ***
free_throw_pct                       ***
steals                               **
total_rebounds                       ***
assists_field_goal_pct_interaction   ***
assists_team_score_interaction       ***
field_goal_pct_team_score_interaction ***
```

Signif. codes:

0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.174 on 2619 degrees of freedom

Multiple R-squared: 0.9908, Adjusted R-squared: 0.9907

F-statistic: 2.81e+04 on 10 and 2619 DF, p-value: < 2.2e-16

- The best interaction term illustrated would be the interaction between field goal percentage and assists against the team score.
- Including interaction terms in this case has generated an improved model that fits the data. This allows us to establish changes in the outcome more accurately.
- In the above results, using the interactions has brought the R squared to 99.08%
- Overall NBA performance is affected by the 2 main predictors assists and field goal percentages

Part III: Analysis and Findings (Anish)

```
hoopR_dt [2,630 × 7] (S3: hoopR_data/tbl_df/tbl/data.table/data.frame)
 $ assists       : int [1:2630] 18 21 26 23 28 20 28 23 26 29 ...
 $ blocks        : int [1:2630] 7 7 7 3 5 3 4 2 4 4 ...
 $ field_goal_pct: num [1:2630] 34.4 45.2 49.4 44.9 51.2 37 48.7 52 40.6 50.6 ...
 $ fouls         : int [1:2630] 21 13 18 19 18 22 22 21 15 8 ...
 $ free_throw_pct: num [1:2630] 87.5 56.5 76.2 85 81.5 78.9 90 86.4 100 80 ...
 $ steals        : int [1:2630] 9 6 11 2 3 7 5 7 5 4 ...
 $ total_rebounds: int [1:2630] 44 57 34 37 58 33 31 38 43 45 ...
 - attr(*, "hoopR_timestamp")= POSIXct[1:1], format: "2024-04-22 04:44:11"
 - attr(*, "hoopR_type")= chr "ESPN NBA Team Boxscores from hoopR data repository"
> str(Y)
 int [1:2630] 89 94 108 95 109 94 111 108 93 104 ..
```

```
# Install necessary packages
install.packages("caret", dependencies = TRUE)

# Load necessary libraries
library(caret, lib.loc = .libPaths(), verbose = TRUE)



# Load necessary libraries
library(caret)
library(MASS)
library(glmnet)
library(leaps)

# Subset the data for regression analysis
```

```
regression_data <- new_data

# Define predictor terms
predictor_terms <- c("assists", "blocks", "field_goal_pct", "fouls", "free_throw_pct", "steals",
"total_rebounds")

# Define outcome variable
outcome_variable <- "team_score"
str(X)
str(Y)

# Convert Y to a vector
Y <- as.vector(Y$team_score)

Y <- as.vector(Y)

# Check the dimensions of X and Y
dim(X)
length(Y)



# Subset X to match the length of Y
X <- X[1:length(Y), ]
breaks = sequence (0, 2630, by = 1)

# Load necessary libraries
install.packages(c("caret", "MASS", "glmnet", "leaps"))
library(caret)
library(MASS)
library(glmnet)
library(leaps)
```

```
# Subset the data for regression analysis
regression_data <- subset(new_data, select = c("assists", "blocks", "field_goal_pct", "fouls",
"free_throw_pct", "steals", "total_rebounds", "team_score"))

# Define predictor terms
predictor_terms <- c("assists", "blocks", "field_goal_pct", "fouls", "free_throw_pct", "steals",
"total_rebounds")

# Define outcome variable
outcome_variable <- "team_score"

### Regression Problem using Generalized Regression Models ###

# Check the structure of the regression_data dataframe
str(regression_data)

# Verify the names of columns in regression_data
names(regression_data)

# Ensure that 'team_score' is correctly named in the dataframe
# If not, replace 'team_score' with the correct name

# Run regsubsets() with error handling
tryCatch({
  subset_model <- regsubsets(
    team_score ~ .,
    data = regression_data,
    nvmax = length(predictor_terms),
    method = "forward"
  )
```

```
}, error = function(e) {
  message("Error: ", e)
})

# Run regsubsets() for subset selection
subset_model <- regsubsets(
  team_score ~ .,
  data = regression_data,
  nvmax = length(predictor_terms),
  method = "forward"
)

# Summary of subset selection
subset_summary <- summary(subset_model)

# Check the summary
print(subset_summary)



# Subset Selection
subset_model <- regsubsets(team_score ~ ., data = regression_data[, predictor_terms], nvmax =
length(predictor_terms), method = "forward")
subset_summary <- summary(subset_model)
subset_selected_terms <- names(coef(subset_model, id = which.min(subset_summary$cp)))
subset_coefficients <- coef(subset_model, id = which.min(subset_summary$cp))
subset_cv_errors <- subset_summary$cp[which.min(subset_summary$cp)]

# Check the data type of team_score
class(regression_data$team_score)
# Convert the outcome variable to a matrix
y_matrix <- as.matrix(regression_data[, outcome_variable])
```

```
# Run glmnet again
ridge_model <- glmnet(
  x = as.matrix(regression_data[, predictor_terms]),
  y = y_matrix,
  alpha = 0
)

# Convert the outcome variable to a matrix
y_matrix <- as.matrix(regression_data[, outcome_variable])

# Run cv.glmnet again
ridge_cv <- cv.glmnet(
  x = as.matrix(regression_data[, predictor_terms]),
  y = y_matrix,
  alpha = 0
)



# Extract selected terms
ridge_selected_terms <- predict(ridge_cv, type = "nonzero")$s0

# Extract coefficients
ridge_coefficients <- coef(ridge_cv)

# Extract cross-validated errors
ridge_cv_errors <- min(ridge_cv$cvm)

# Check the structure of the outcome variable
str(regression_data[, outcome_variable])
```

```
# Compare Results and Determine Optimal Model
if (subset_cv_errors < ridge_cv_errors) {
  optimal_model <- "Subset Selection"
  selected_terms <- subset_selected_terms
  coefficients <- subset_coefficients
  cv_error <- subset_cv_errors
} else {
  optimal_model <- "Ridge Regression"
  selected_terms <- rownames(ridge_coefficients)[ridge_selected_terms]
  coefficients <- coef(ridge_cv)[, ridge_selected_terms]
  cv_error <- ridge_cv_errors
}
```


```
# Compare Results and Determine Optimal Model
if (subset_cv_errors < ridge_cv_errors) {
  optimal_model <- "Subset Selection"
  selected_terms <- subset_selected_terms
  coefficients <- subset_coefficients
  cv_error <- subset_cv_errors
} else {
  optimal_model <- "Ridge Regression"
  selected_terms <- rownames(ridge_coefficients)[ridge_selected_terms]
  coefficients <- coef(ridge_cv)[, ridge_selected_terms]
  cv_error <- ridge_cv_errors
}
```


```
# Display the model outputs
cat("Optimal Model:", optimal_model, "\n")
cat("Selected Terms:", selected_terms, "\n")
```

```
cat("Coefficients:\n")
print(coefficients)
cat("Cross-Validated Error:", cv_error, "\n")

# Comment on Findings
# Based on the outputs of the optimal model, analyze coefficients and discuss findings related to business/research questions.


### Classification Problem using Logistic Regression, Linear Discriminant Analysis, KNN ###

# Define outcome variable for classification
classification_data <- new_data
classification_data$binary_variable <- ifelse(classification_data$team_winner == TRUE, 1, 0)

# Define predictor terms for classification
classification_predictors <- c("assists", "blocks", "field_goal_pct", "fouls", "free_throw_pct", "steals", "total_rebounds")

# Logistic Regression
logistic_model <- glm(binary_variable ~ ., data = classification_data[, c(classification_predictors, "binary_variable")], family = binomial)
logistic_cv_error <- 1 - sum(predict(logistic_model, type = "response") > 0.5 == classification_data$binary_variable) / nrow(classification_data)
logistic_selected_terms <- names(coef(logistic_model))

# Logistic Regression
logistic_model <- glm(binary_variable ~ .,
                      data = classification_data[, c(classification_predictors, "binary_variable")],
                      family = binomial)
```

```
# Predict probabilities and calculate error
logistic_probabilities <- predict(logistic_model, type = "response")
logistic_cv_error <- 1 - sum(ifelse(logistic_probabilities > 0.5, 1, 0) ==
classification_data$binary_variable) / nrow(classification_data)



# Linear Discriminant Analysis (LDA)
lda_model <- lda(binary_variable ~ ., data = classification_data[, c(classification_predictors,
"binary_variable")])
lda_cv_error <- mean(lda_model$posterior[, 1] > 0.5 != classification_data$binary_variable)
lda_selected_terms <- names(lda_model$scaling)

# Linear Discriminant Analysis (LDA)
lda_model <- lda(binary_variable ~ .,
                 data = classification_data[, c(classification_predictors, "binary_variable")])

# Predict probabilities and calculate error
lda_probabilities <- predict(lda_model)$posterior[, 1]
lda_cv_error <- mean(ifelse(lda_probabilities > 0.5, 1, 0) != classification_data$binary_variable)

# Print Model Accuracies
cat("Logistic Regression CV Error:", logistic_cv_error, "\n")
cat("LDA CV Error:", lda_cv_error, "\n")

# KNN
knn_model <- train(binary_variable ~ ., data = classification_data[, c(classification_predictors,
"binary_variable")], method = "knn")
knn_cv_error <- 1 - min(knn_model$results$Accuracy)
knn_selected_terms <- knn_model$bestTune$k

# Correcting the outcome variable to a factor
```

```
classification_data$binary_variable <- as.factor(classification_data$binary_variable)

# KNN model training
knn_model <- train(binary_variable ~ .,
                   data = classification_data[, c(classification_predictors, "binary_variable")],
                   method = "knn")

# Check for missing values in results
if (is.null(knn_model$results$Accuracy)) {
  cat("Error: No results available for KNN model.\n")
} else {
  # Calculate CV error
  knn_cv_error <- 1 - min(knn_model$results$Accuracy)

  # Extract selected terms (k value)
  knn_selected_k <- knn_model$bestTune$k

  # Print the outputs
  cat("KNN CV Error:", knn_cv_error, "\n")
  cat("Selected k:", knn_selected_k, "\n")
}


# Compare Results and Determine Optimal Model
optimal_classification_model <- ifelse(logistic_cv_error < lda_cv_error & logistic_cv_error <
knn_cv_error, "Logistic Regression",
                                       ifelse(lda_cv_error < knn_cv_error, "LDA", "KNN"))
```

RESULTS:

SUBSET SUMMARY

```
Subset selection object
Call: regsubsets.formula(team_score ~ ., data = regression_data, nvmax =
length(predictor_terms),
   method = "forward")
7 Variables  (and intercept)
             Forced in Forced out
assists           FALSE      FALSE
blocks            FALSE      FALSE
field_goal_pct    FALSE      FALSE
fouls             FALSE      FALSE
free_throw_pct    FALSE      FALSE
steals            FALSE      FALSE
total_rebounds    FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: forward
         assists blocks field_goal_pct fouls free_throw_pct steals total_rebounds
1  ( 1 ) " "     " "    "*"           " "    " "           " "    " "
2  ( 1 ) " "     " "    "*"           " "    " "           " "    "*"
3  ( 1 ) "*"     " "    "*"           " "    " "           " "    "*"
4  ( 1 ) "*"     " "    "*"           " "    "*"           " "    "*"
5  ( 1 ) "*"     " "    "*"           "*"    "*"           " "    "*"
6  ( 1 ) "*"     " "    "*"           "*"    "*"           "*"    "*"
7  ( 1 ) "*"     "*"    "*"           "*"    "*"           "*"    "*"
```

Based on the provided code, here are the findings:

1. **Data Type of** team_score**:** The data type of the team_score variable in the regression_data dataframe is numeric.
2. **Model Building:**
    - **Ridge Regression Model:** Initially, the glmnet function is used to fit a ridge regression model (ridge_model) using the specified predictor terms and the outcome variable converted to a matrix format.

- **Cross-Validation:** Cross-validation is performed using the cv.glmnet function (ridge_cv) to estimate the optimal regularization parameter for the ridge regression model.

3. **Model Comparison and Determining Optimal Model:**
   - The cross-validated errors (subset_cv_errors and ridge_cv_errors) from subset selection and ridge regression, respectively, are compared to determine the optimal model.
   - If the cross-validated error from subset selection is lower than that from ridge regression, the optimal model is determined to be subset selection. Otherwise, the optimal model is ridge regression.
   - The selected terms, coefficients, and cross-validated error of the optimal model are extracted for further analysis.
4. **Outcome Variable Structure Check:** Finally, the structure of the outcome variable (team_score) in the regression_data dataframe is checked to ensure consistency and proper handling during model fitting.

```
> # Display the model outputs
> cat("Optimal Model:", optimal_model, "\n")
Optimal Model: Subset Selection
> cat("Selected Terms:", selected_terms, "\n")
Selected Terms: (Intercept) assists blocks field_goal_pct fouls free_throw_pct steals
total_rebounds
> cat("Coefficients:\n")
Coefficients:
> print(coefficients)
  (Intercept)       assists        blocks field_goal_pct         fouls free_throw_pct
  -11.3660107     0.5426406    -0.1706098     1.4449410     0.3626274     0.1662197
       steals total_rebounds
    0.4559232     0.4740050
> cat("Cross-Validated Error:", cv_error, "\n")
Cross-Validated Error: 8

> # Print Model Accuracies
> cat("Logistic Regression CV Error:", logistic_cv_error, "\n")
```

Logistic Regression CV Error: 0.2201521

> cat("LDA CV Error:", lda_cv_error, "\n")

LDA CV Error: 0.778327

KNN CV Error: 0.3064414

Selected k: 9

Based on the cross-validation errors calculated for logistic regression, LDA, and KNN, the results are as follows:

- Logistic Regression CV Error: 0.2201521
- LDA CV Error: 0.778327
- KNN CV Error: 0.3064414

Based on these errors, the optimal classification model would be Logistic Regression, as it has the lowest cross-validation error among the three models.
Therefore, the output would be:
Optimal Classification Model: Logistic Regression
This indicates that Logistic Regression is the preferred model for classification based on the provided cross-validation errors.

Part IV: Conclusions and Recommendations

Conclusion

Based on the findings provided:

1. Data Type of team_score: The team_score variable in the regression_data data frame is numeric.

2. Model Building: • Ridge Regression Model: Created using glmnet function. • Cross-Validation: Performed to estimate the optimal regularization parameter.

3. Model Comparison and Optimal Model Determination: • Cross-validated errors from subset selection and ridge regression are compared. • Optimal model chosen based on lower cross-validated error.

4. Outcome Variable Structure Check: Ensured consistency and proper handling during model fitting. Regarding the assist predictor's effect on the dependent variable:

• Specific findings regarding the impact of the assist predictor on the team_score variable
However, the analysis likely involved examining the coefficients and significance of the assist predictor within the models to understand its relationship with the team_score.

Recommendations

After completing our regression and classification analysis we can make recommendations to NBA coaches on what adjustments they can make to coaching that will result in higher scoring and more wins. Improving field goal percentage is not an easy fix for a team. Coaches have to convey the importance of making shots to their players. Players who have lower field goal percentage averages can focus on practicing shot taking more. An easier improvement that coaches can make is increasing team assists. Coaches game plan in the NBA should rely on team basketball, instead of hero ball that we have seen throughout the years. This doesn't mean that players should make a pass every time instead of shooting the ball, but players should be aware of their teammates and look for open looks that can lead to an assist. An improvement in team assists can even lead to improvement in team field goal percentage.

Part V: References

- https://hoopr.sportsdataverse.org/

Team Contributions

Part I: Introduction (Miguel)
Part II: Data Preparation and EDA (Reem, Wael)
Part III: Analysis and Findings (Anish)
Part IV: Conclusions and Recommendations (Group worked together)