DATA621 Homework #1 - Appendix A

William Aiken, Donald Butler, Michael Ippolito, Bharani Nittala, Leticia Salazar, Santiago Torres

9/25/2022

1. DATA EXPLORATION

Due to the number of fields in this data, I broke the dataset into intuitive sections and explored each section individually.

```
# Load data
eval_df <- read.csv("https://raw.githubusercontent.com/catfoodlover/DATA621/main/HW1/moneyball-evaluation-data.csv")
train_df <- read.csv("https://raw.githubusercontent.com/catfoodlover/DATA621/main/HW1/moneyball-training-data.csv")
```

Base Hits by Batter

- TARGET_WINS - Number of wins
- TEAM_BATTING_H - Base Hits by batters (1B,2B,3B,HR)
- TEAM_BATTING_2B - Doubles by batters (2B)
- TEAM_BATTING_3B - Triples by batters (3B)
- TEAM_BATTING_HR - Homeruns by batters (4B)

The means and medians are very similar for the base hits variables implying little skew to the distributions.

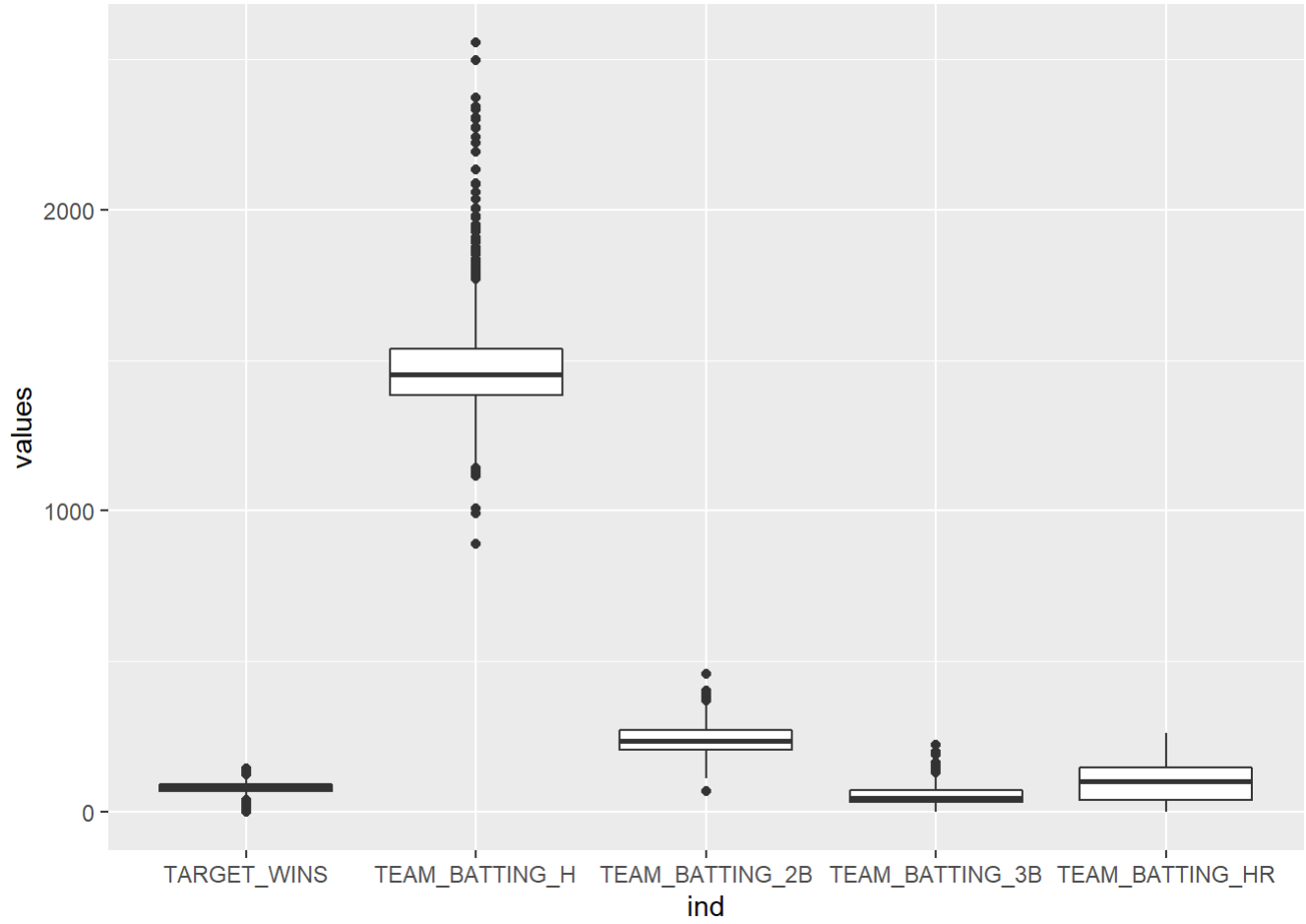
```
# Summary
train_df %>% select(c("TARGET_WINS", "TEAM_BATTING_H","TEAM_BATTING_2B","TEAM_BATTING_3B","TEAM_BATTING_HR")) %>% gtsummary::tbl_summary(statistic =list(c("TARGET_WINS", "TEAM_BATTING_H","TEAM_BATTING_2B","TEAM_BATTING_3B","TEAM_BATTING_HR") ~ "{mean} {median} {sd}"
))
```

Characteristic	**N = 2,276**[1]
TARGET_WINS	81 82 16
TEAM_BATTING_H	1,469 1,454 145
TEAM_BATTING_2B	241 238 47
TEAM_BATTING_3B	55 47 28
TEAM_BATTING_HR	100 102 61

[1] Mean Median SD

We see tight distributions except for all base hits by batters (TEAM_BATTING_H).

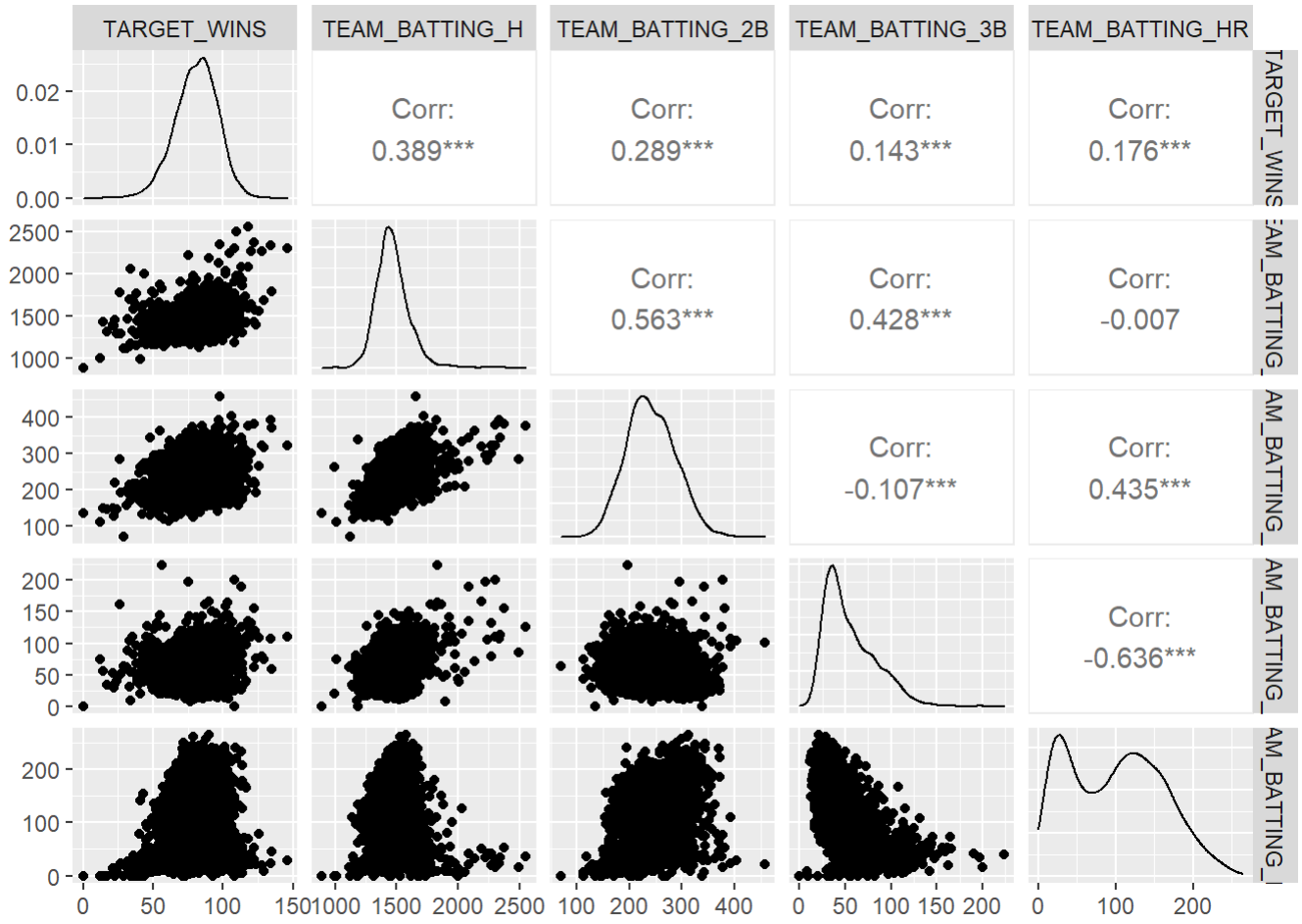
```
# Box plots
temp <- train_df %>% select(c("TARGET_WINS", "TEAM_BATTING_H","TEAM_BATTING_2B","TEAM_BATTING_3
B","TEAM_BATTING_HR"))
ggplot2::ggplot(stack(temp), aes(x = ind, y = values)) +
  geom_boxplot()
```



Unsurprisingly, all possible base hits (TEAM_BATTING_H) is correlated with winning. As you increase the number of bases achieved by an at bat, the correlation decreases.

Interestingly, doubles and triples are correlated with base hits while home runs are not.

```
# Correlation plot
train_df %>% select(c("TARGET_WINS", "TEAM_BATTING_H","TEAM_BATTING_2B","TEAM_BATTING_3B","TEAM_
BATTING_HR")) %>% GGally::ggpairs()
```



Batting

- TARGET_WINS - Number of wins
- TEAM_BATTING_BB - Walks by batters
- TEAM_BATTING_HBP - Batters hit by pitch (get a free base)
- TEAM_BATTING_SO - Strikeouts by batters
- TEAM_BASERUN_SB - Stolen bases
- TEAM_BASERUN_CS - Caught stealing

The measures of central tendency show us that most of these variable have slight skew to their distributions. Stolen bases has a large right skew to its distribution.

We are missing values for stikeouts, stolen bases and caught stealing.

```
# Summary
train_df %>% select(c("TEAM_BATTING_BB", "TEAM_BATTING_SO", "TEAM_BASERUN_SB", "TEAM_BASERUN_CS"
)) %>% gtsummary::tbl_summary(statistic =list(c("TEAM_BATTING_BB", "TEAM_BATTING_SO", "TEAM_BASE
RUN_SB", "TEAM_BASERUN_CS") ~ "{mean} {median} {sd}"
))
```

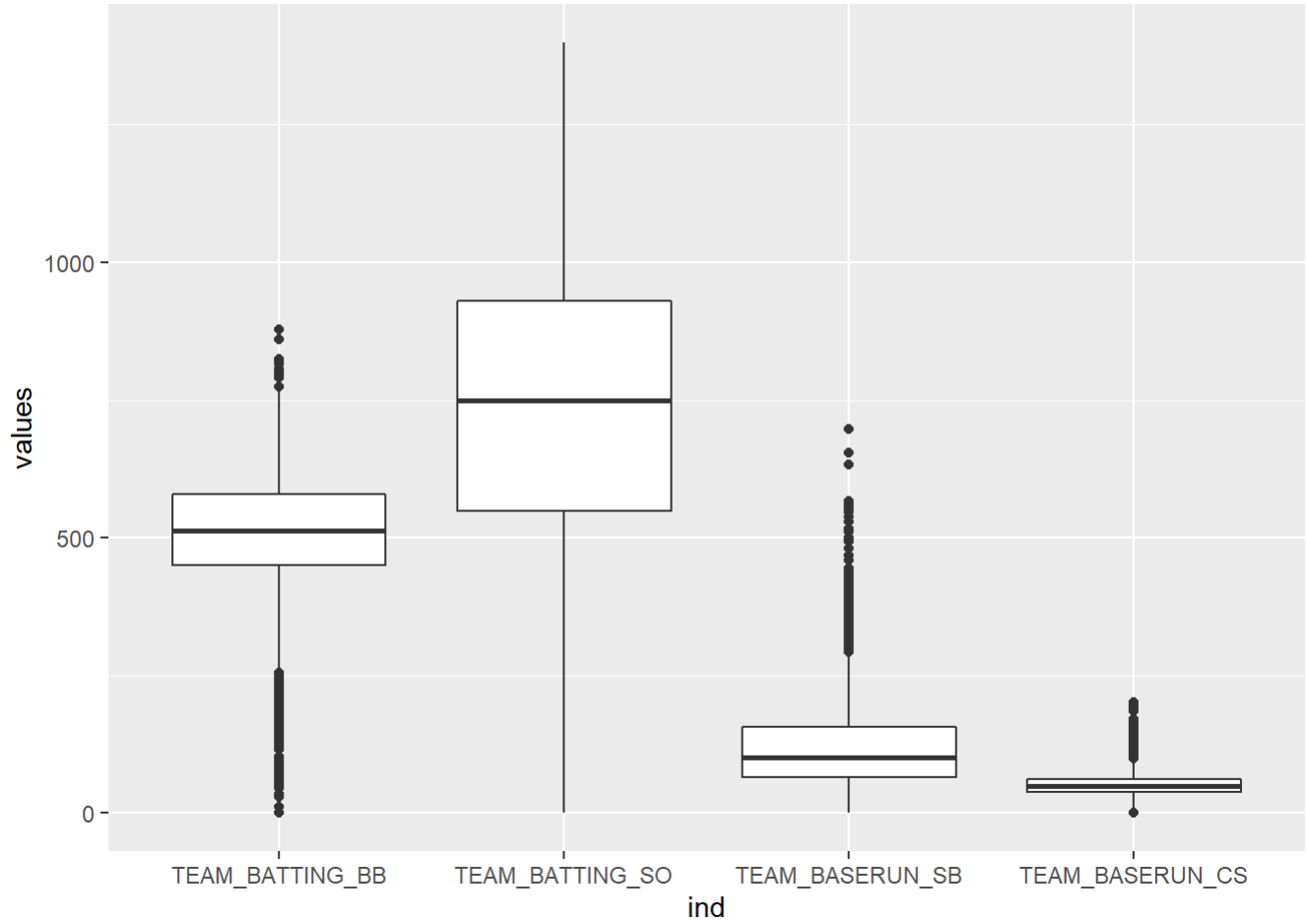
Characteristic	**N = 2,276**[1]
TEAM_BATTING_BB	502 512 123

[1] Mean Median SD

Characteristic	N = 2,276[1]
TEAM_BATTING_SO	736 750 249
Unknown	102
TEAM_BASERUN_SB	125 101 88
Unknown	131
TEAM_BASERUN_CS	53 49 23
Unknown	772

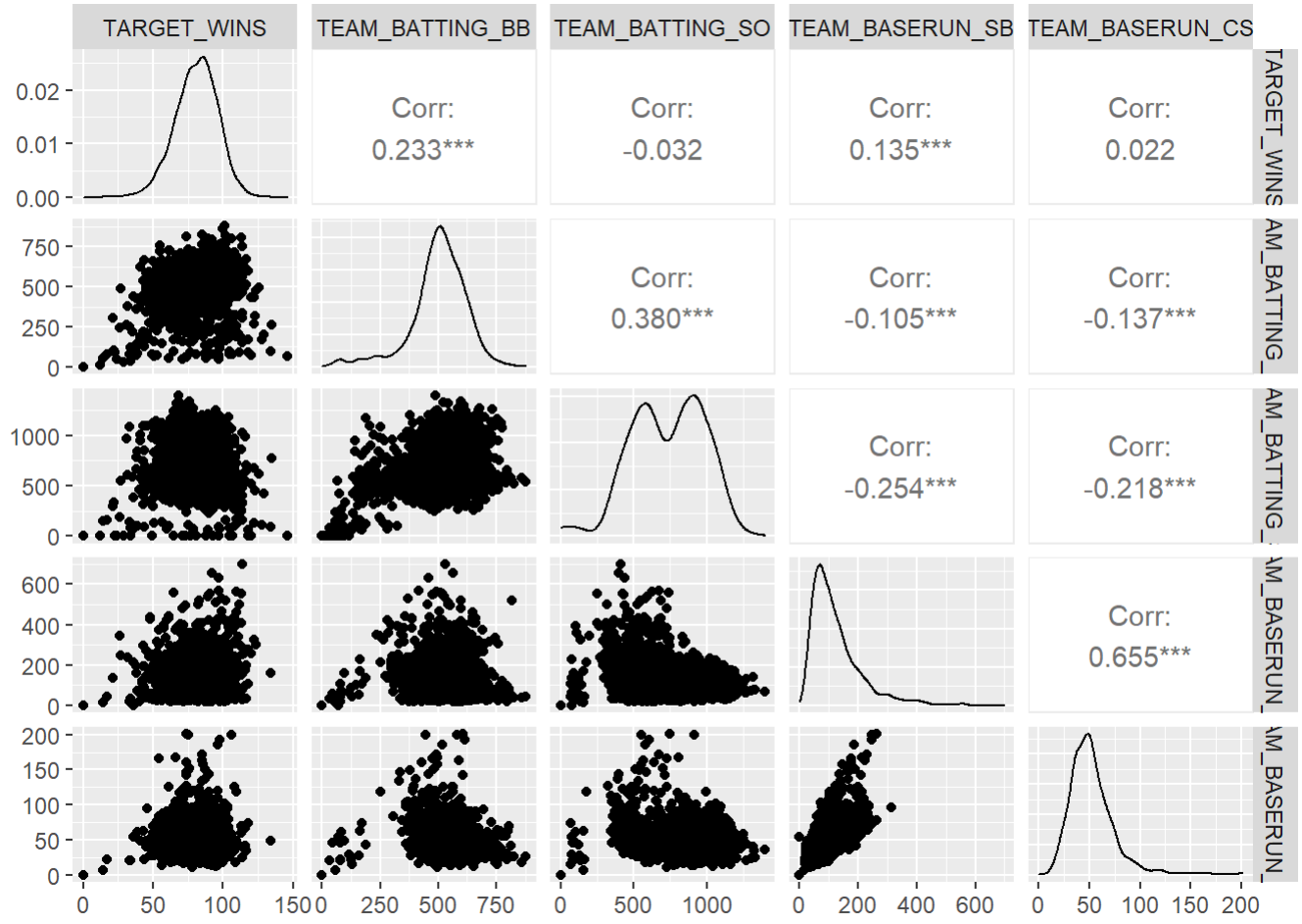
[1] Mean Median SD

```
# Box plots
temp <- train_df %>% select(c("TEAM_BATTING_BB", "TEAM_BATTING_SO", "TEAM_BASERUN_SB", "TEAM_BAS
ERUN_CS"))
ggplot2::ggplot(stack(temp), aes(x = ind, y = values)) +
  geom_boxplot()
```



Of all the batting variables, only walks by batter has a correlation to wins.

```
# Correlation plot
train_df %>% select(c("TARGET_WINS", "TEAM_BATTING_BB", "TEAM_BATTING_SO", "TEAM_BASERUN_SB", "T
EAM_BASERUN_CS")) %>% GGally::ggpairs()
```



Fielding

- TARGET_WINS - Number of wins
- TEAM_FIELDING_E - Errors
- TEAM_FIELDING_DP - Double Plays

The Errors variable(TEAM_FIELDING_E) has an incredibly right skewed distribution. We are missing some Double Plays values.

```
# Summary
train_df %>% select(c("TEAM_FIELDING_E", "TEAM_FIELDING_DP")) %>% gtsummary::tbl_summary(statist
ic =list(c("TEAM_FIELDING_E", "TEAM_FIELDING_DP") ~ "{mean} {median} {sd}"
))
```

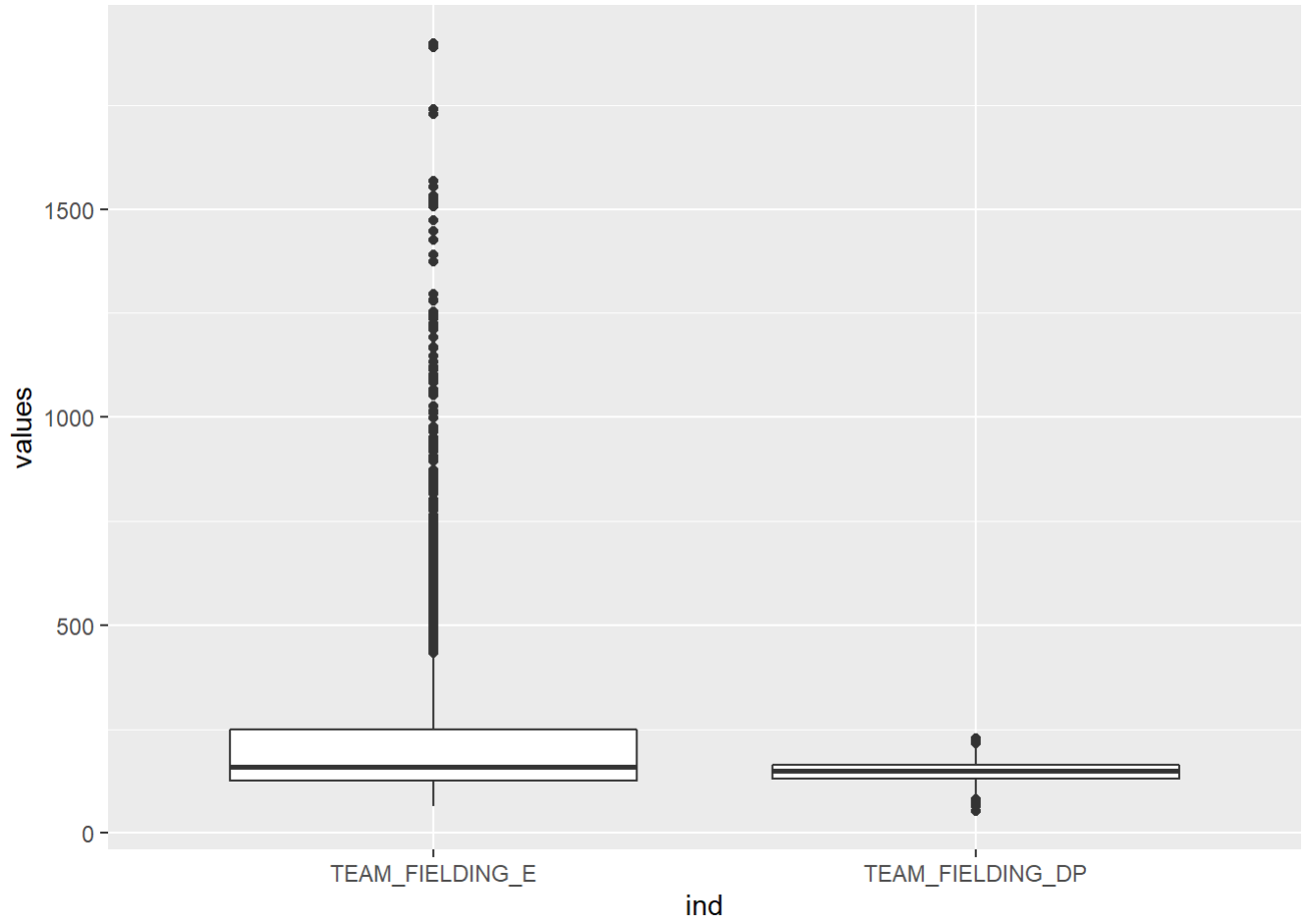
Characteristic	**N = 2,276**[1]
TEAM_FIELDING_E	246 159 228

[1] Mean Median SD

Characteristic	**N = 2,276**[1]
TEAM_FIELDING_DP	146 149 26
Unknown	286

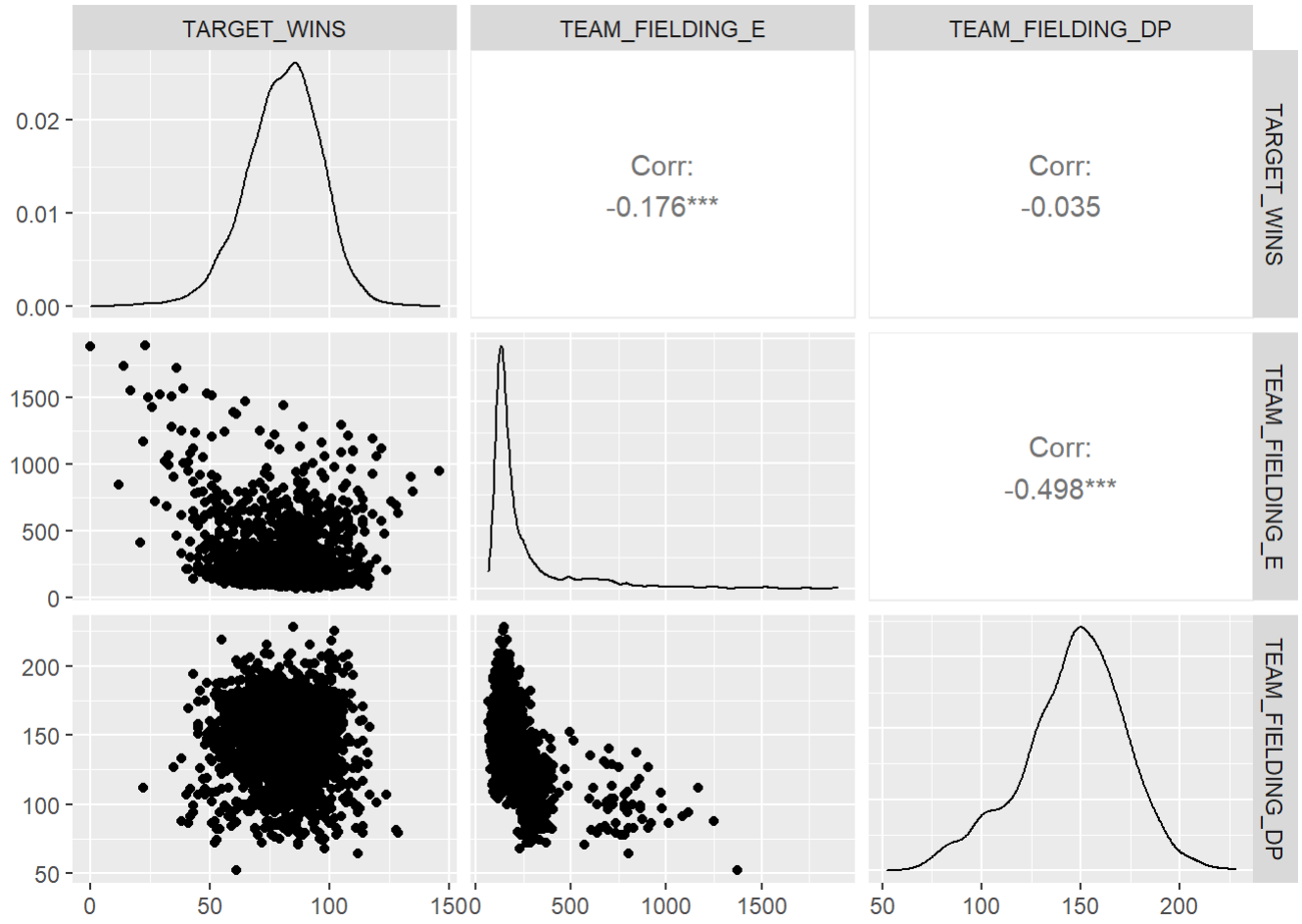
[1] Mean Median SD

```
# Box plots
temp <- train_df %>% select(c("TEAM_FIELDING_E", "TEAM_FIELDING_DP"))
ggplot2::ggplot(stack(temp), aes(x = ind, y = values)) +
  geom_boxplot()
```



Both the Fielding variables are negatively correlated with Wins.

```
# Correlation plot
train_df %>% select(c("TARGET_WINS", "TEAM_FIELDING_E", "TEAM_FIELDING_DP"))%>% GGally::ggpairs
()
```



Pitching

- TARGET_WINS - Number of wins
- TEAM_PITCHING_BB - Walks allowed
- TEAM_PITCHING_H - Hits allowed
- TEAM_PITCHING_HR - Homeruns allowed
- TEAM_PITCHING_SO - Strikeouts by pitchers

Hits allowed (TEAM_PITCHING_H) has a right skew and we are missing some Strikeouts by pitcher (TEAM_PITCHING_SO) values.

```
# Summary
train_df %>% select(c("TEAM_PITCHING_H", "TEAM_PITCHING_HR", "TEAM_PITCHING_BB", "TEAM_PITCHING_
SO")) %>% gtsummary::tbl_summary(statistic =list(c("TEAM_PITCHING_H", "TEAM_PITCHING_HR", "TEAM_
PITCHING_BB", "TEAM_PITCHING_SO") ~ "{mean} {median} {sd}"
))
```

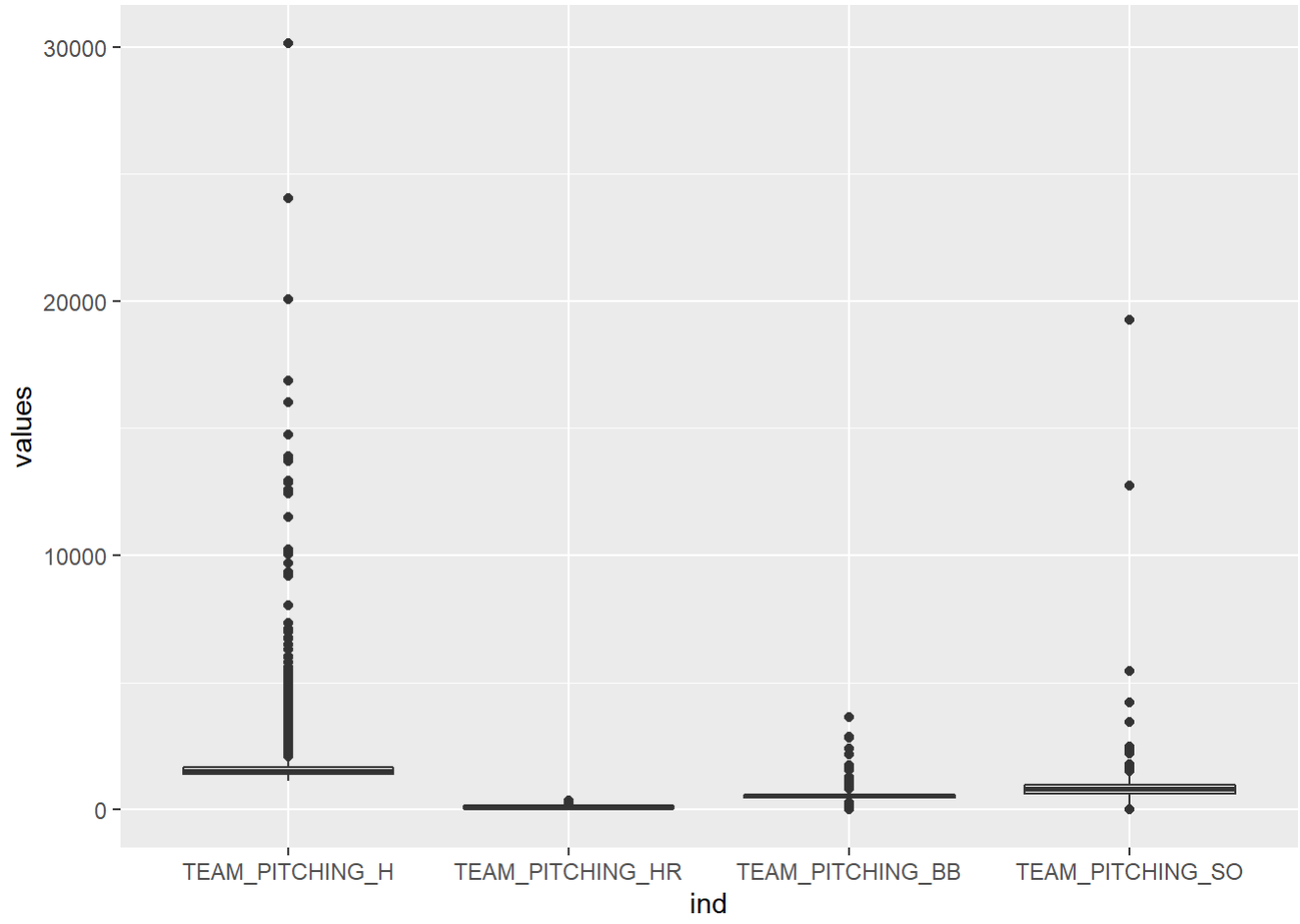
Characteristic	**N = 2,276**[1]
TEAM_PITCHING_H	1,779 1,518 1,407
TEAM_PITCHING_HR	106 107 61

[1] Mean Median SD

Characteristic	N = 2,276[1]
TEAM_PITCHING_BB	553 536 166
TEAM_PITCHING_SO	818 814 553
Unknown	102

[1] Mean Median SD

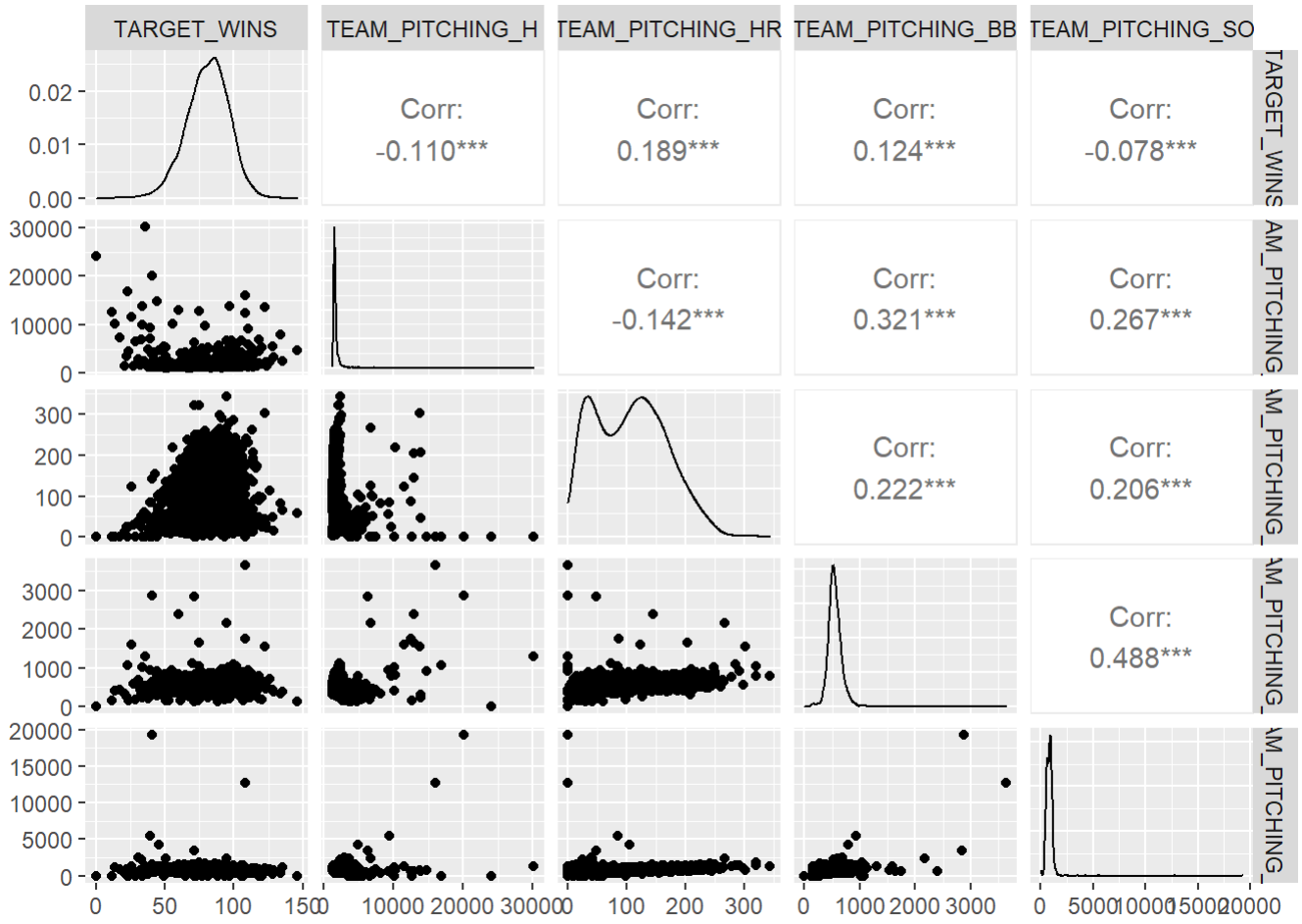
```
# Box plots
temp <- train_df %>% select(c("TEAM_PITCHING_H", "TEAM_PITCHING_HR", "TEAM_PITCHING_BB", "TEAM_P
ITCHING_SO"))
ggplot2::ggplot(stack(temp), aes(x = ind, y = values)) +
  geom_boxplot()
```



Hits allowed is negatively correlated with Winning.

Interestingly, Home runs allowed is positively correlated with Winning.

```
# Correlation plot
train_df %>% select(c("TARGET_WINS","TEAM_PITCHING_H", "TEAM_PITCHING_HR", "TEAM_PITCHING_BB",
"TEAM_PITCHING_SO"))%>% GGally::ggpairs()
```



Data Overview

The data set contains approximately 2276 records. Each record represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season.

Below is a short description of the variables:

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT
INDEX	Identification Variable (do not use)	None
TARGET_WINS	Number of wins	
TEAM_BATTING_H	Base Hits by batters (1B,2B,3B,HR)	Positive Impact on Wins
TEAM_BATTING_2B	Doubles by batters (2B)	Positive Impact on Wins
TEAM_BATTING_3B	Triples by batters (3B)	Positive Impact on Wins
TEAM_BATTING_HR	Homeruns by batters (4B)	Positive Impact on Wins
TEAM_BATTING_BB	Walks by batters	Positive Impact on Wins
TEAM_BATTING_HBP	Batters hit by pitch (get a free base)	Positive Impact on Wins
TEAM_BATTING_SO	Strikeouts by batters	Negative Impact on Wins
TEAM_BASERUN_SB	Stolen bases	Positive Impact on Wins
TEAM_BASERUN_CS	Caught stealing	Negative Impact on Wins
TEAM_FIELDING_E	Errors	Negative Impact on Wins
TEAM_FIELDING_DP	Double Plays	Positive Impact on Wins
TEAM_PITCHING_BB	Walks allowed	Negative Impact on Wins
TEAM_PITCHING_H	Hits allowed	Negative Impact on Wins
TEAM_PITCHING_HR	Homeruns allowed	Negative Impact on Wins
TEAM_PITCHING_SO	Strikeouts by pitchers	Positive Impact on Wins

https://raw.githubusercontent.com/letisalba/Data_621/master/Homework_1/Images/homework1_table.png (https://raw.githubusercontent.com/letisalba/Data_621/master/Homework_1/Images/homework1_table.png)

- INDEX - Identification Variable
- TARGET_WINS - Number of wins
- TEAM_BATTING_H - Base Hits by batters (1B,2B,3B,HR)
- TEAM_BATTING_2B - Doubles by batters (2B)
- TEAM_BATTING_3B - Triples by batters (3B)
- TEAM_BATTING_HR - Homeruns by batters (4B)
- TEAM_BATTING_BB - Walks by batters
- TEAM_BATTING_HBP - Batters hit by pitch (get a free base)
- TEAM_BATTING_SO - Strikeouts by batters
- TEAM_BASERUN_SB - Stolen bases
- TEAM_BASERUN_CS - Caught stealing
- TEAM_FIELDING_E - Errors
- TEAM_FIELDING_DP - Double Plays
- TEAM_PITCHING_BB - Walks allowed
- TEAM_PITCHING_H - Hits allowed
- TEAM_PITCHING_HR - Homeruns allowed
- TEAM_PITCHING_SO - Strikeouts by pitchers

Objective

To build a multiple linear regression model on the training data to predict *TARGET_WINS*, which is the number of wins for the team.

Data Exploration and Preparation

```
# Read data
baseball_df <- read.csv('https://raw.githubusercontent.com/letisalba/Data_621/master/Homework_1/csv/moneyball-training-data.csv')
baseball_eval <- read.csv('https://raw.githubusercontent.com/letisalba/Data_621/master/Homework_1/csv/moneyball-evaluation-data.csv')

# Data overview
head(baseball_df)
```

```
##   INDEX TARGET_WINS TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
## 1     1          39           1445              194               39
## 2     2          70           1339              219               22
## 3     3          86           1377              232               35
## 4     4          70           1387              209               38
## 5     5          82           1297              186               27
## 6     6          75           1279              200               36
##   TEAM_BATTING_HR TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB
## 1              13             143              842              NA
## 2             190             685             1075              37
## 3             137             602              917              46
## 4              96             451              922              43
## 5             102             472              920              49
## 6              92             443              973             107
##   TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
## 1              NA               NA            9364               84
## 2              28               NA            1347              191
## 3              27               NA            1377              137
## 4              30               NA            1396               97
## 5              39               NA            1297              102
## 6              59               NA            1279               92
##   TEAM_PITCHING_BB TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP
## 1              927             5456             1011               NA
## 2              689             1082              193              155
## 3              602              917              175              153
## 4              454              928              164              156
## 5              472              920              138              168
## 6              443              973              123              149
```

```
dim(baseball_df)
```

```
## [1] 2276   17
```

```
# Data summary
summary(baseball_df)
```

```
##      INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
##  Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
##  1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
##  Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0
##
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0
##  Median : 47.00   Median :102.00   Median :512.0   Median : 750.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 735.6
##  3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
##                                                    NA's   :102
##  TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H
##  Min.   :  0.0   Min.   :  0.0   Min.   :29.00    Min.   : 1137
##  1st Qu.: 66.0   1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419
##  Median :101.0   Median : 49.0   Median :58.00    Median : 1518
##  Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779
##  3rd Qu.:156.0   3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682
##  Max.   :697.0   Max.   :201.0   Max.   :95.00    Max.   :30132
##  NA's   :131     NA's   :772     NA's   :2085
##  TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##  Min.   :  0.0    Min.   :   0.0   Min.   :    0.0   Min.   :  65.0
##  1st Qu.: 50.0    1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0
##  Median :107.0    Median : 536.5   Median :  813.5   Median : 159.0
##  Mean   :105.7    Mean   : 553.0   Mean   :  817.7   Mean   : 246.5
##  3rd Qu.:150.0    3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2
##  Max.   :343.0    Max.   :3645.0   Max.   :19278.0   Max.   :1898.0
##                                    NA's   :102
##  TEAM_FIELDING_DP
##  Min.   : 52.0
##  1st Qu.:131.0
##  Median :149.0
##  Mean   :146.4
##  3rd Qu.:164.0
##  Max.   :228.0
##  NA's   :286
```

```
print(paste0('Number of observations: ', nrow(baseball_df)))
```

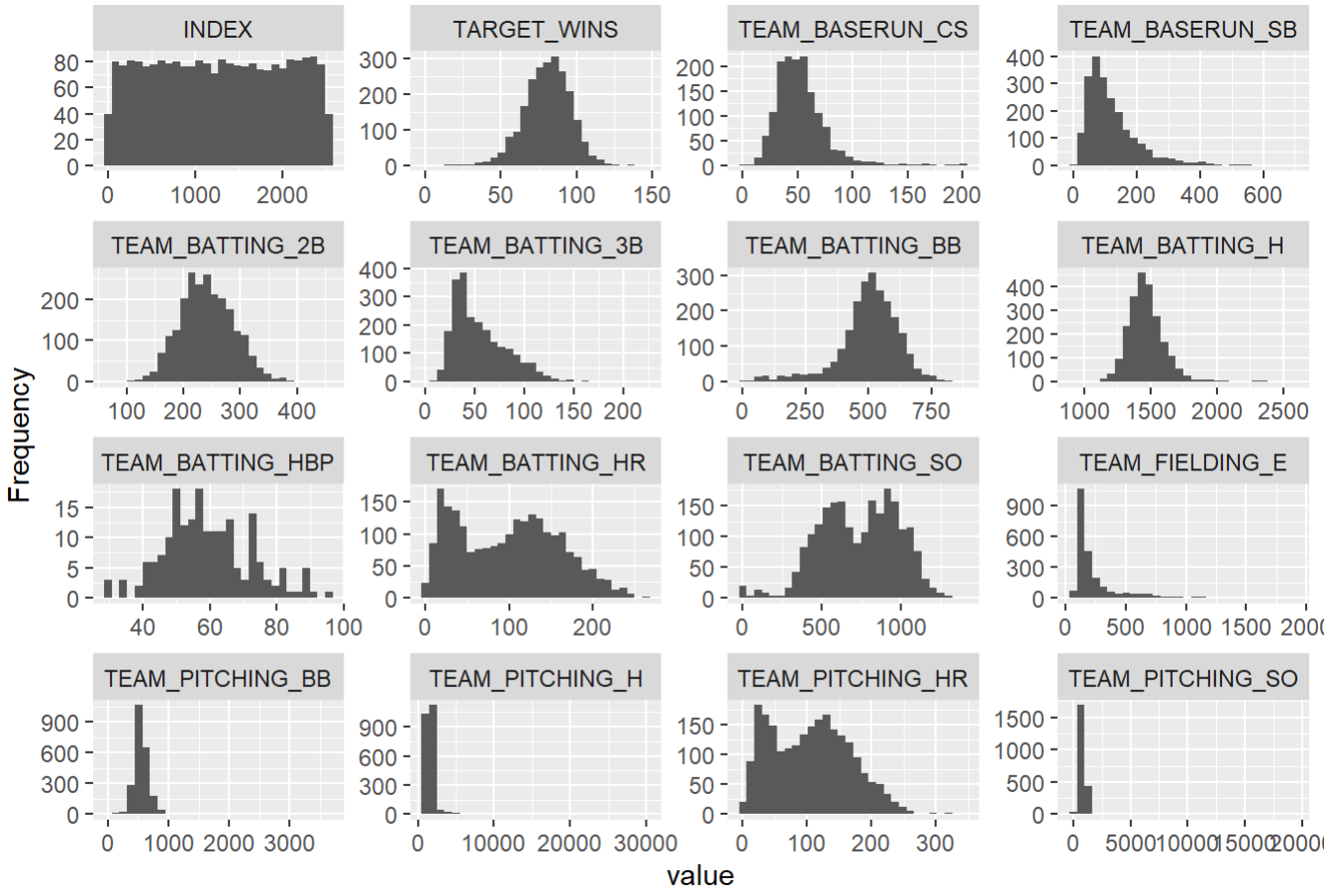
```
## [1] "Number of observations: 2276"
```

```
print(paste0('Observations per year, 1871 - 2006: ',round(nrow(baseball_df)/(2006-1871),2)))
```

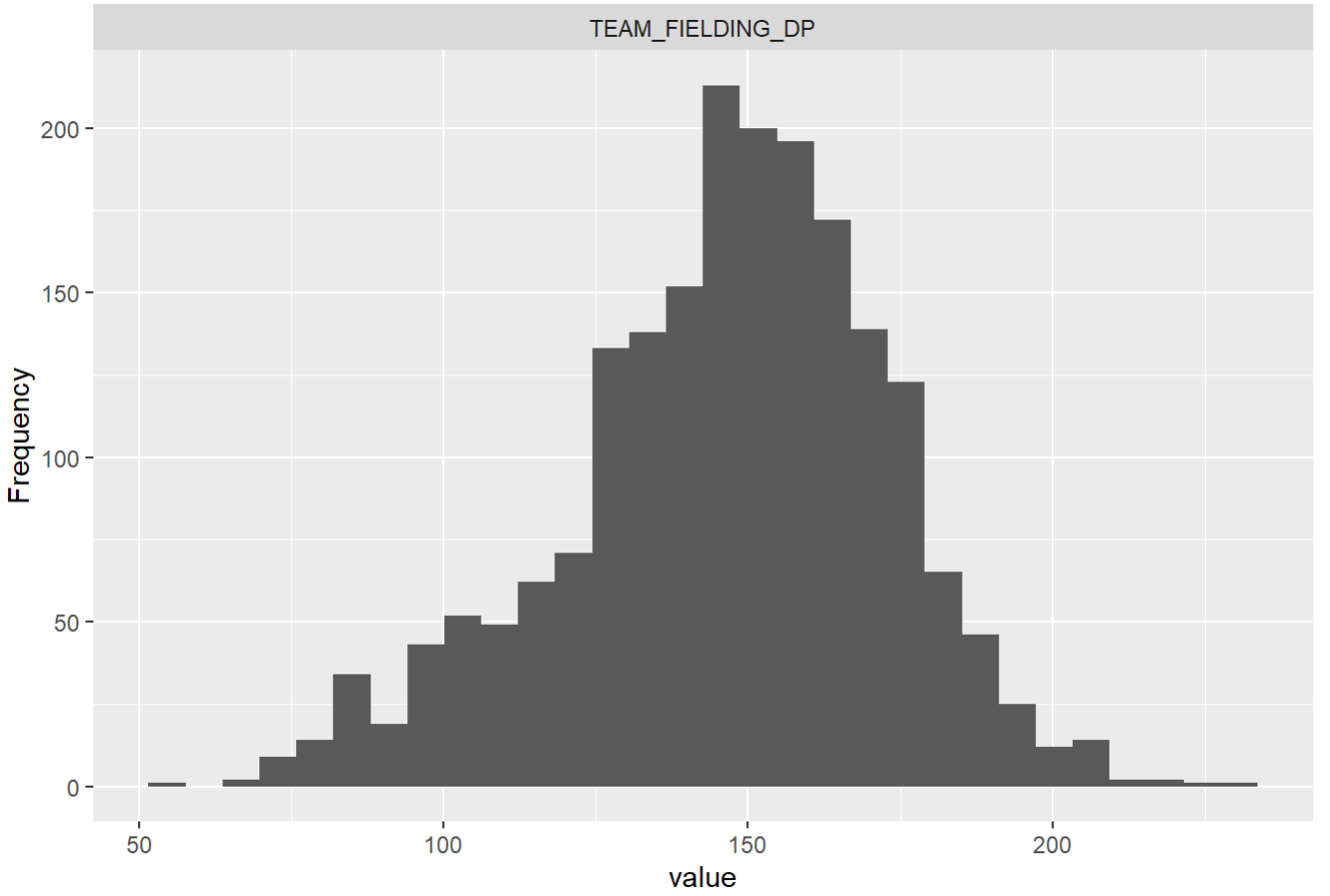
```
## [1] "Observations per year, 1871 - 2006: 16.86"
```

Some columns have maximum values that are clearly outliers, like TEAM_PITCHING_H AND TEAM_PITCHING_HR. The assignment mentions that some of the season records were adjusted to match the performance during a 162-game season. There are 2276 seasons in the training set. Observations span 128 years, with an average of 17 teams playing per year.

```
# Distribution
plot_histogram(baseball_df)
```


INDEX
TARGET_WINS
TEAM_BASERUN_CS
TEAM_BASERUN_SB
TEAM_BATTING_2B
TEAM_BATTING_3B
TEAM_BATTING_BB
TEAM_BATTING_H
TEAM_BATTING_HBP
TEAM_BATTING_HR
TEAM_BATTING_SO
TEAM_FIELDING_E
TEAM_PITCHING_BB
TEAM_PITCHING_H
TEAM_PITCHING_HR
TEAM_PITCHING_SO
Frequency
value

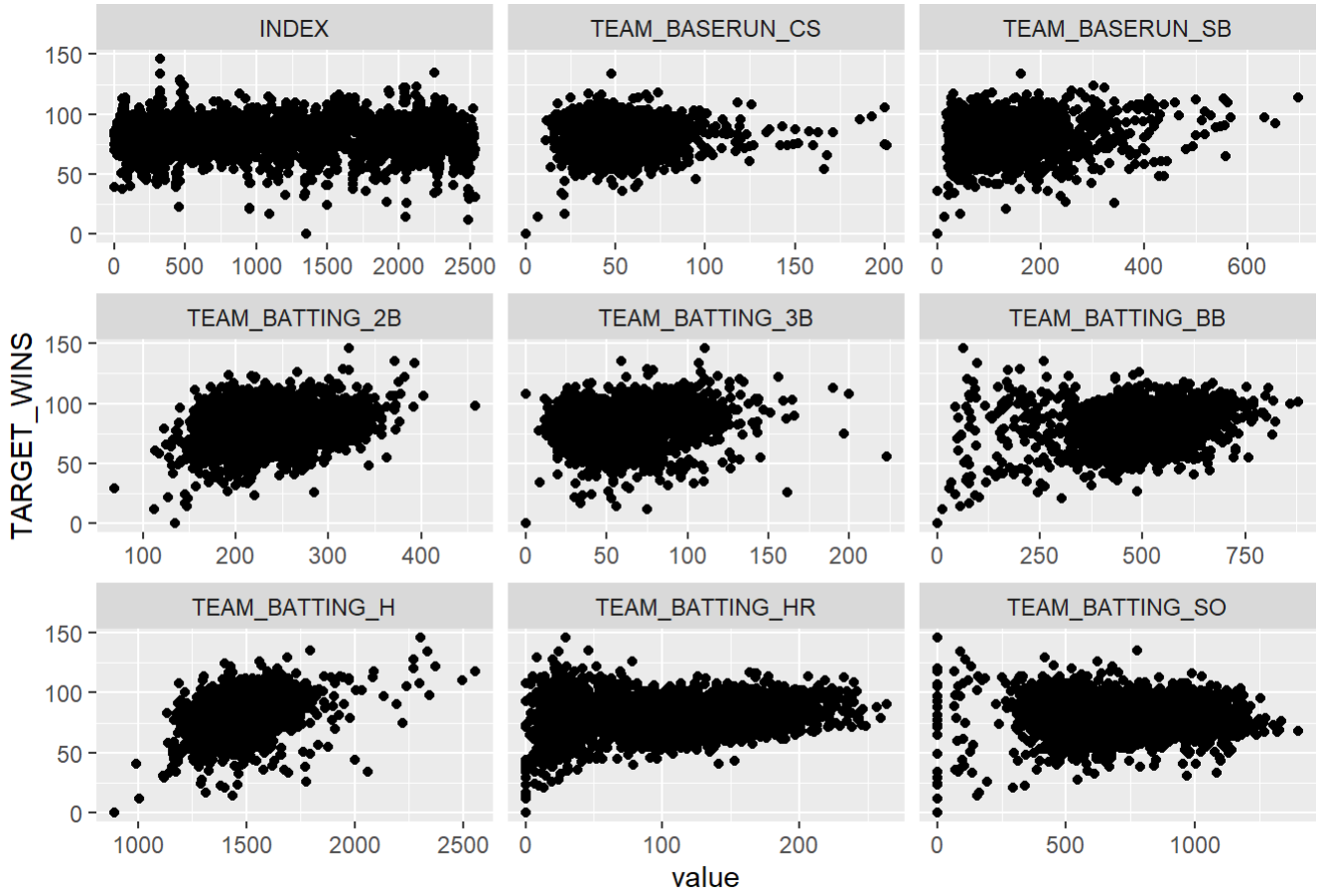
Page 1


TEAM_FIELDING_DP
Frequency
value

Page 2

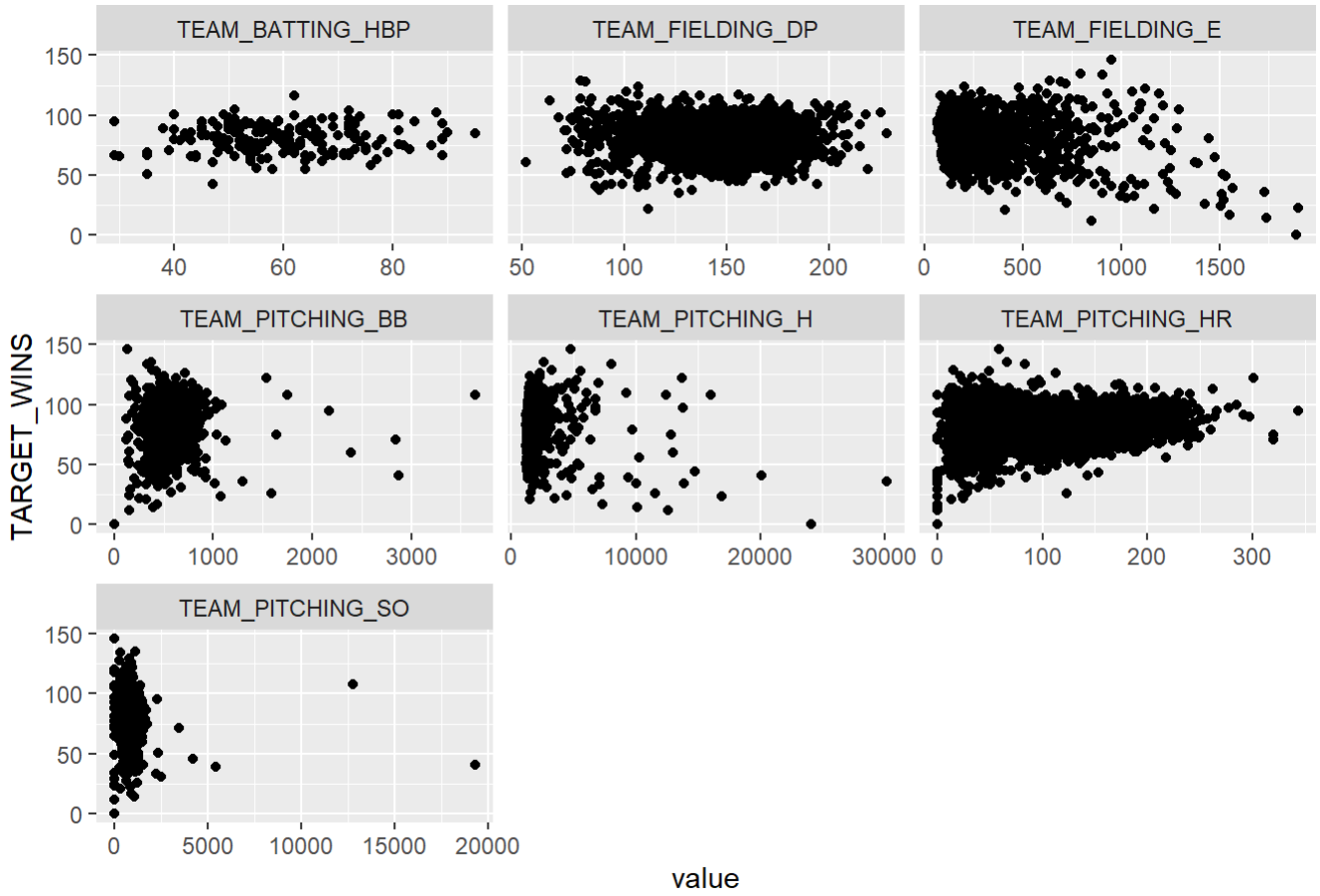
```
# Plot against the response variable
plot_scatterplot(baseball_df, by = "TARGET_WINS")
```

```
## Warning: Removed 1005 rows containing missing values (geom_point).
```



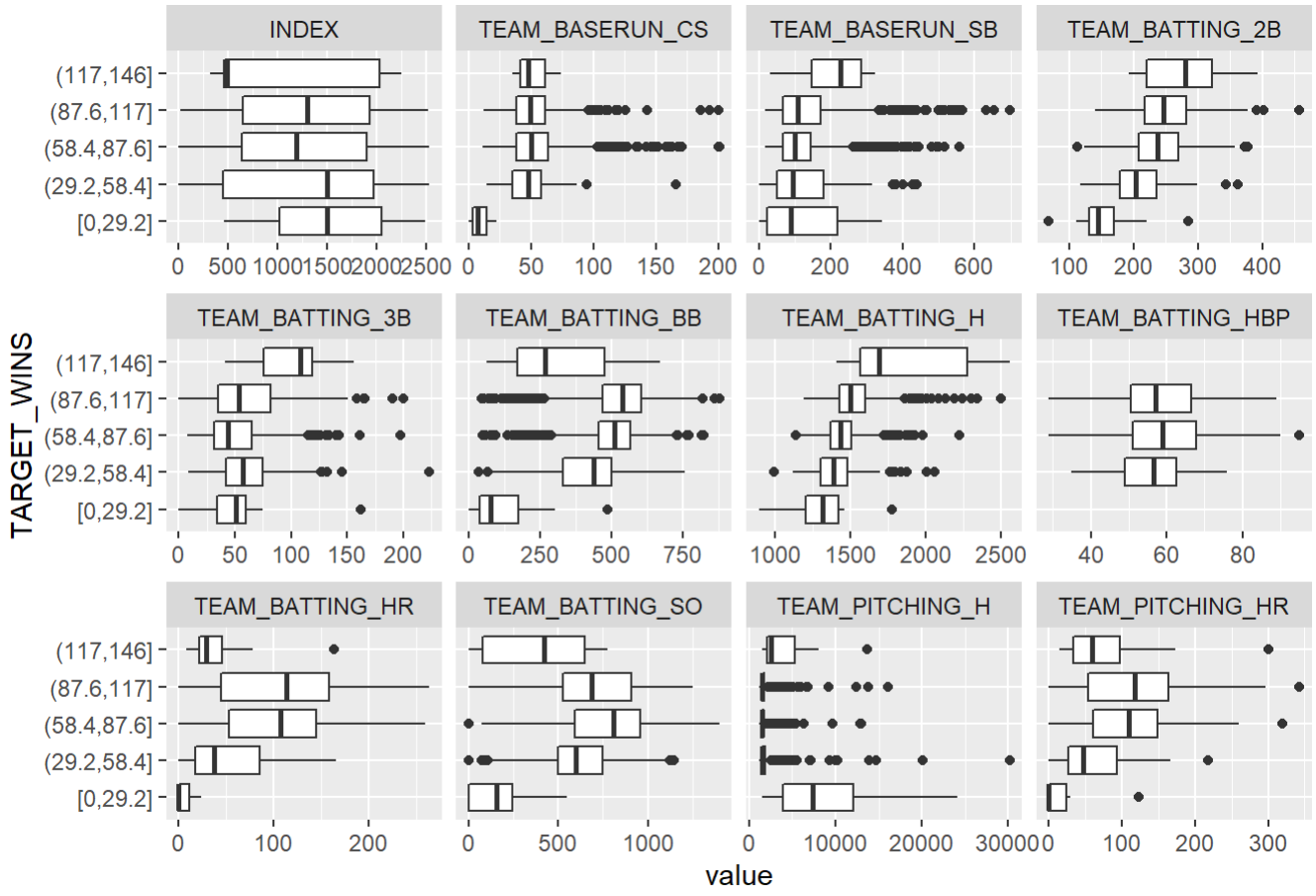
Page 1

```
## Warning: Removed 2473 rows containing missing values (geom_point).
```

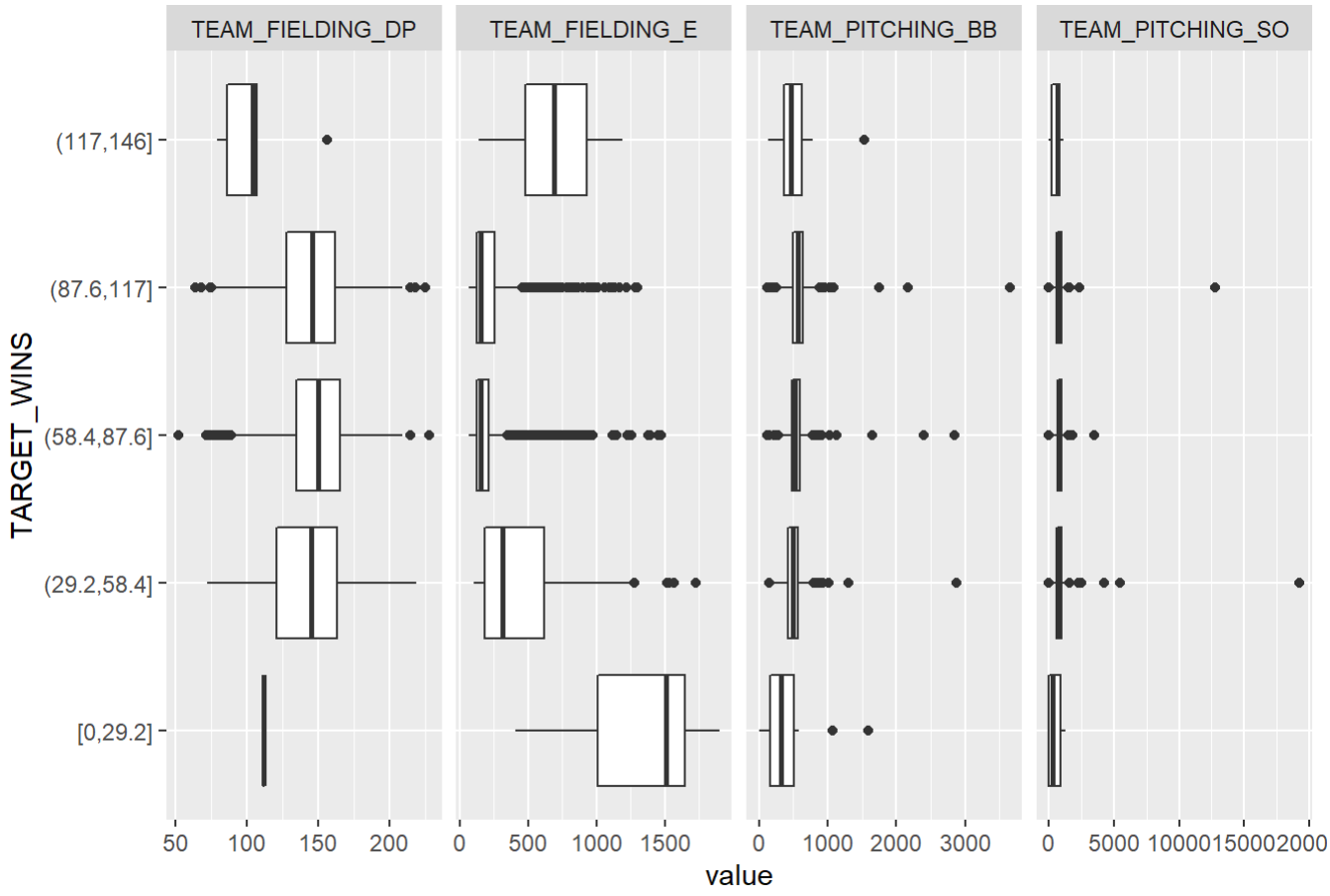



```
# Boxplot for train dataset
plot_boxplot(baseball_df, by = "TARGET_WINS")
```

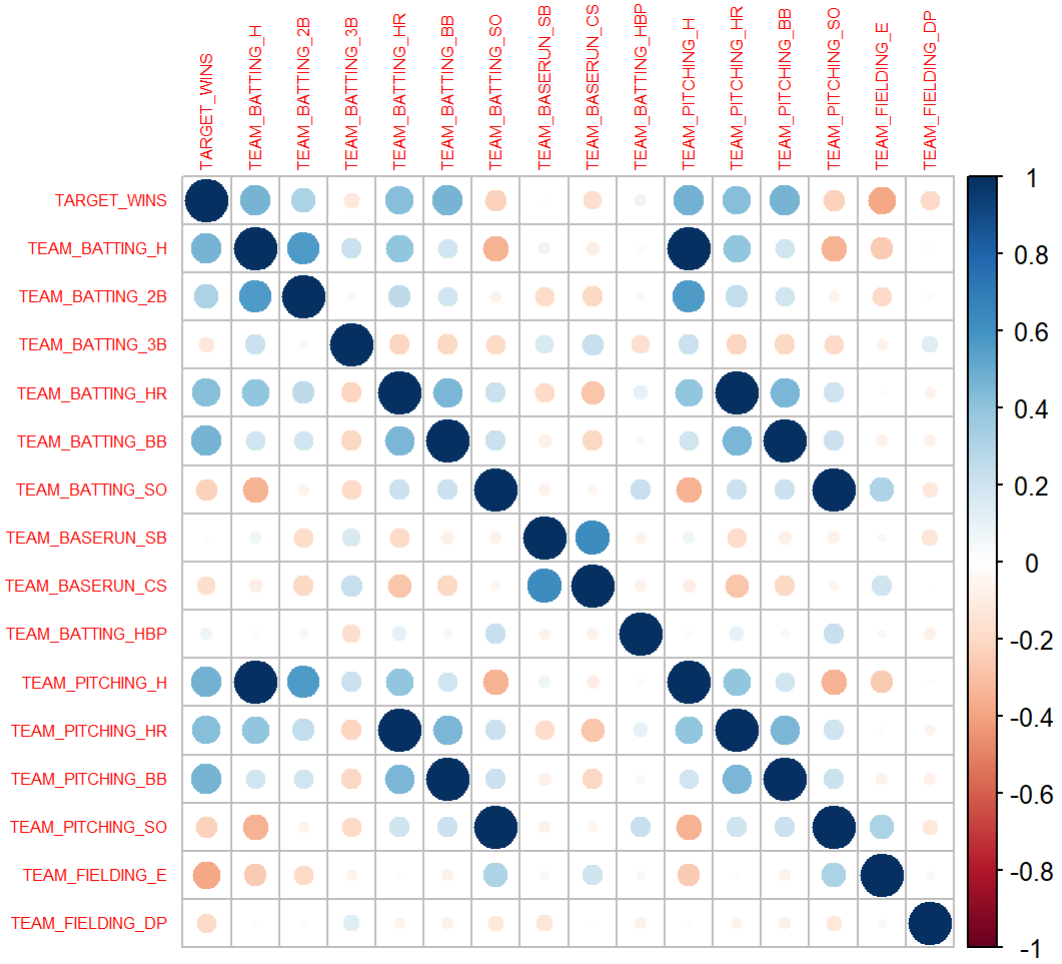
```
## Warning: Removed 3090 rows containing non-finite values (stat_boxplot).
```




```
## Warning: Removed 388 rows containing non-finite values (stat_boxplot).
```




```
# Correlation plot
corrplot(cor(baseball_df[,2:17], use = 'complete.obs'), tl.cex = 0.5)
```



Looking at the correlation plot, there appear to be several strong correlations between explanatory variables and the target. From an initial inspection, it appears the team should focus on getting players on base through hits or walks. Teams can still win if the pitchers allow homeruns, hits and walks to the other team.

Variables with Highest Positive Correlation with TARGET_WINS:

- TEAM_BATTING_H = 0.47
- TEAM_BATTING_HR = 0.42
- TEAM_BATTING_BB = 0.47
- TEAM_PITCHING_H = 0.47
- TEAM_PITCHING_HR = 0.42
- TEAM_PITCHING_BB = 0.47

To win more games it makes sense the team will need to make fewer errors.

Variables with Strongly Negative Correlation with TARGET_WINS:

- There were several batting variables which were related.

Positive Correlations between variables:

- TEAM_PITCHING_H and TEAM_BATTING_H = 0.99
- TEAM_PITCHING_HR and TEAM_BATTING_HR = 0.99
- TEAM_PITCHING_BB and TEAM_BATTING_BB = 0.99
- TEAM_PITCHING_SO and TEAM_BATTING_SO = 0.99

Missing values

```
# Missing values
round(100*colSums(is.na(baseball_df))/nrow(baseball_df),2) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"), font_size = 13)
```

	x
INDEX	0.00
TARGET_WINS	0.00
TEAM_BATTING_H	0.00
TEAM_BATTING_2B	0.00
TEAM_BATTING_3B	0.00
TEAM_BATTING_HR	0.00
TEAM_BATTING_BB	0.00
TEAM_BATTING_SO	4.48
TEAM_BASERUN_SB	5.76
TEAM_BASERUN_CS	33.92
TEAM_BATTING_HBP	91.61
TEAM_PITCHING_H	0.00
TEAM_PITCHING_HR	0.00
TEAM_PITCHING_BB	0.00
TEAM_PITCHING_SO	4.48
TEAM_FIELDING_E	0.00
TEAM_FIELDING_DP	12.57

In terms of missing values, there are two variables missing many observations. TEAM_BATTING_HBP is missing over 90% of its values, while TEAM_BASERUN_CS is missing just around 30%.

```
#New DF with Missing Removed
baseball_df_mv <- baseball_df[, !names(baseball_df) %in% c('TEAM_BATTING_HBP','TEAM_BASERUN_CS',
'TEAM_FIELDING_DP')]
summary(baseball_df_mv)
```

```
##      INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
##  Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
##  1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
##  Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0
##
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0
##  Median : 47.00   Median :102.00   Median :512.0   Median : 750.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 735.6
##  3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
##                                                    NA's   :102
##  TEAM_BASERUN_SB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##  Min.   :  0.0   Min.   : 1137   Min.   :  0.0    Min.   :   0.0
##  1st Qu.: 66.0   1st Qu.: 1419   1st Qu.: 50.0    1st Qu.: 476.0
##  Median :101.0   Median : 1518   Median :107.0    Median : 536.5
##  Mean   :124.8   Mean   : 1779   Mean   :105.7    Mean   : 553.0
##  3rd Qu.:156.0   3rd Qu.: 1682   3rd Qu.:150.0    3rd Qu.: 611.0
##  Max.   :697.0   Max.   :30132   Max.   :343.0    Max.   :3645.0
##  NA's   :131
##  TEAM_PITCHING_SO  TEAM_FIELDING_E
##  Min.   :    0.0   Min.   :  65.0
##  1st Qu.:  615.0   1st Qu.: 127.0
##  Median :  813.5   Median : 159.0
##  Mean   :  817.7   Mean   : 246.5
##  3rd Qu.:  968.0   3rd Qu.: 249.2
##  Max.   :19278.0   Max.   :1898.0
##  NA's   :102
```

```
#Impute NAs with Median
baseball_df_imputed <- mice(baseball_df_mv, m=5, maxit = 5, method = 'pmm')
```

```
##
##  iter imp variable
##   1   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
```

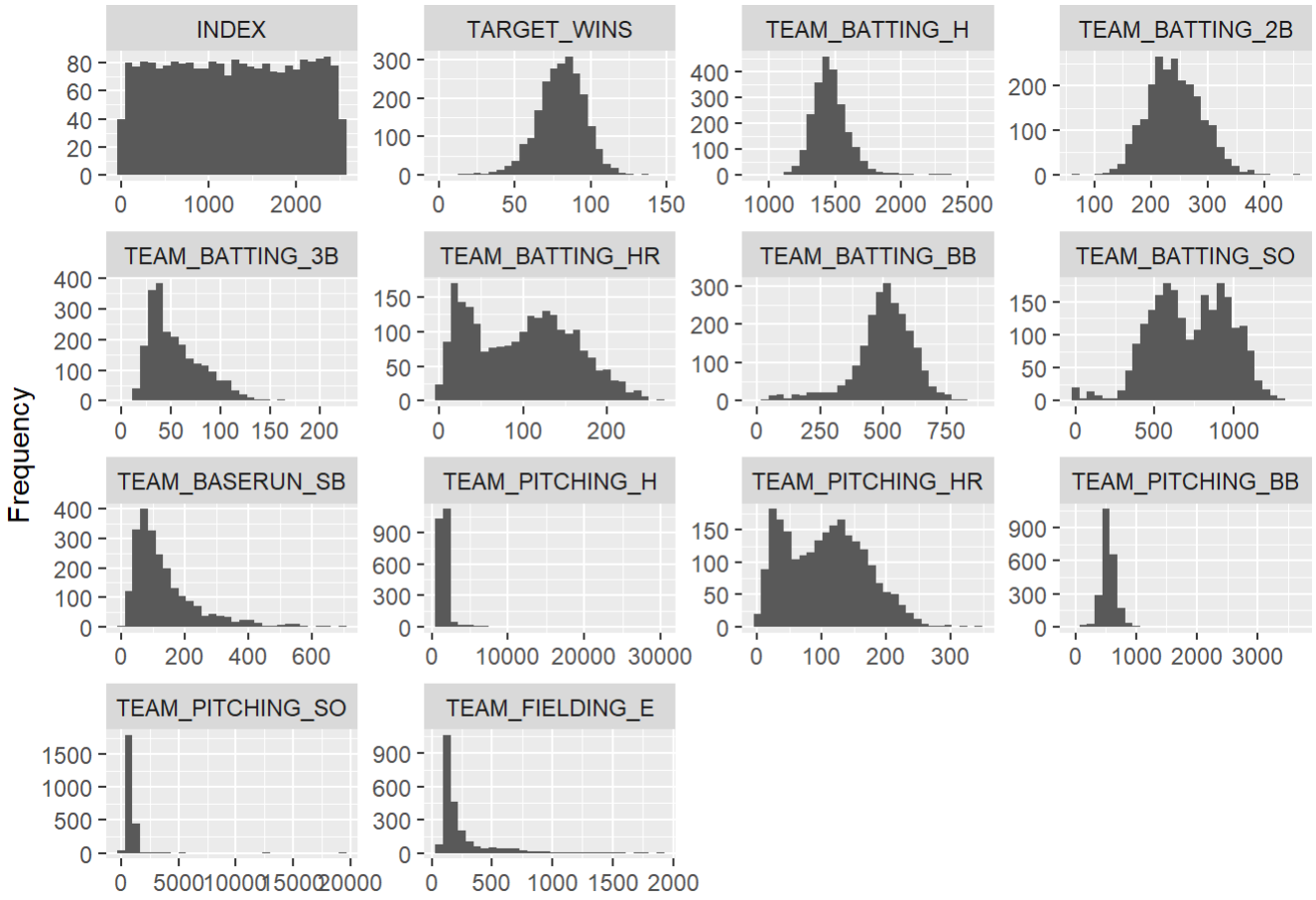
```
baseball_df_final <- complete(baseball_df_imputed)
summary(baseball_df_final)
```

```
##      INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
##  Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
##  1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
##  Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 542.0
##  Median : 47.00   Median :102.00   Median :512.0   Median : 733.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 727.5
##  3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 925.0
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
##  TEAM_BASERUN_SB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##  Min.   :  0.0   Min.   : 1137   Min.   :  0.0    Min.   :   0.0
##  1st Qu.: 67.0   1st Qu.: 1419   1st Qu.: 50.0    1st Qu.: 476.0
##  Median :105.0   Median : 1518   Median :107.0    Median : 536.5
##  Mean   :136.4   Mean   : 1779   Mean   :105.7    Mean   : 553.0
##  3rd Qu.:170.0   3rd Qu.: 1682   3rd Qu.:150.0    3rd Qu.: 611.0
##  Max.   :697.0   Max.   :30132   Max.   :343.0    Max.   :3645.0
##  TEAM_PITCHING_SO  TEAM_FIELDING_E
##  Min.   :    0.0   Min.   :  65.0
##  1st Qu.:  609.0   1st Qu.: 127.0
##  Median :  802.0   Median : 159.0
##  Mean   :  809.9   Mean   : 246.5
##  3rd Qu.:  958.0   3rd Qu.: 249.2
##  Max.   :19278.0   Max.   :1898.0
```

```
ggplot(melt(baseball_df_final), aes(x=value)) + geom_histogram() + facet_wrap(~variable, scale=
'free') + labs(x='', y='Frequency')
```

```
## Using  as id variables
```

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




```
#Replace Error Maxs
baseball_df_final$TEAM_PITCHING_H[baseball_df_final$TEAM_PITCHING_H > 3*sd(baseball_df_final$TEA
M_PITCHING_H)] <- median(baseball_df_final$TEAM_PITCHING_H)
baseball_df_final$TEAM_PITCHING_BB[baseball_df_final$TEAM_PITCHING_BB > 3*sd(baseball_df_final$T
EAM_PITCHING_BB)] <- median(baseball_df_final$TEAM_PITCHING_BB)
baseball_df_final$TEAM_PITCHING_SO[baseball_df_final$TEAM_PITCHING_SO > 3*sd(baseball_df_final$T
EAM_PITCHING_SO)] <- median(baseball_df_final$TEAM_PITCHING_SO)
baseball_df_final$TEAM_FIELDING_E[baseball_df_final$TEAM_FIELDING_E > 3*sd(baseball_df_final$TEA
M_FIELDING_E)] <- median(baseball_df_final$TEAM_FIELDING_E)
summary(baseball_df_final)
```

```
##      INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
##  Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
##  1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
##  Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 542.0
##  Median : 47.00   Median :102.00   Median :512.0   Median : 733.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 727.5
##  3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 925.0
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
##  TEAM_BASERUN_SB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##  Min.   :  0.0   Min.   :1137    Min.   :  0.0    Min.   :  0.0
##  1st Qu.: 67.0   1st Qu.:1419    1st Qu.: 50.0    1st Qu.:476.0
##  Median :105.0   Median :1518    Median :107.0    Median :536.5
##  Mean   :136.4   Mean   :1605    Mean   :105.7    Mean   :500.4
##  3rd Qu.:170.0   3rd Qu.:1660    3rd Qu.:150.0    3rd Qu.:536.5
##  Max.   :697.0   Max.   :4134    Max.   :343.0    Max.   :536.5
##  TEAM_PITCHING_SO TEAM_FIELDING_E
##  Min.   :   0.0   Min.   : 65.0
##  1st Qu.: 609.0   1st Qu.:127.0
##  Median : 802.0   Median :159.0
##  Mean   : 787.9   Mean   :198.9
##  3rd Qu.: 954.2   3rd Qu.:215.0
##  Max.   :1600.0   Max.   :681.0
```

Model Building

Model 1 - Full Model

By testing all variables in this first model we are able to see how significant are the variables in our dataset. We will then be able to use this model to base our other models.

```
# Model 1
m1 <- lm(TARGET_WINS ~., data = baseball_df_final, na.action = na.omit)
summ(m1)
```

Observations	2276
Dependent variable	TARGET_WINS
Type	OLS linear regression

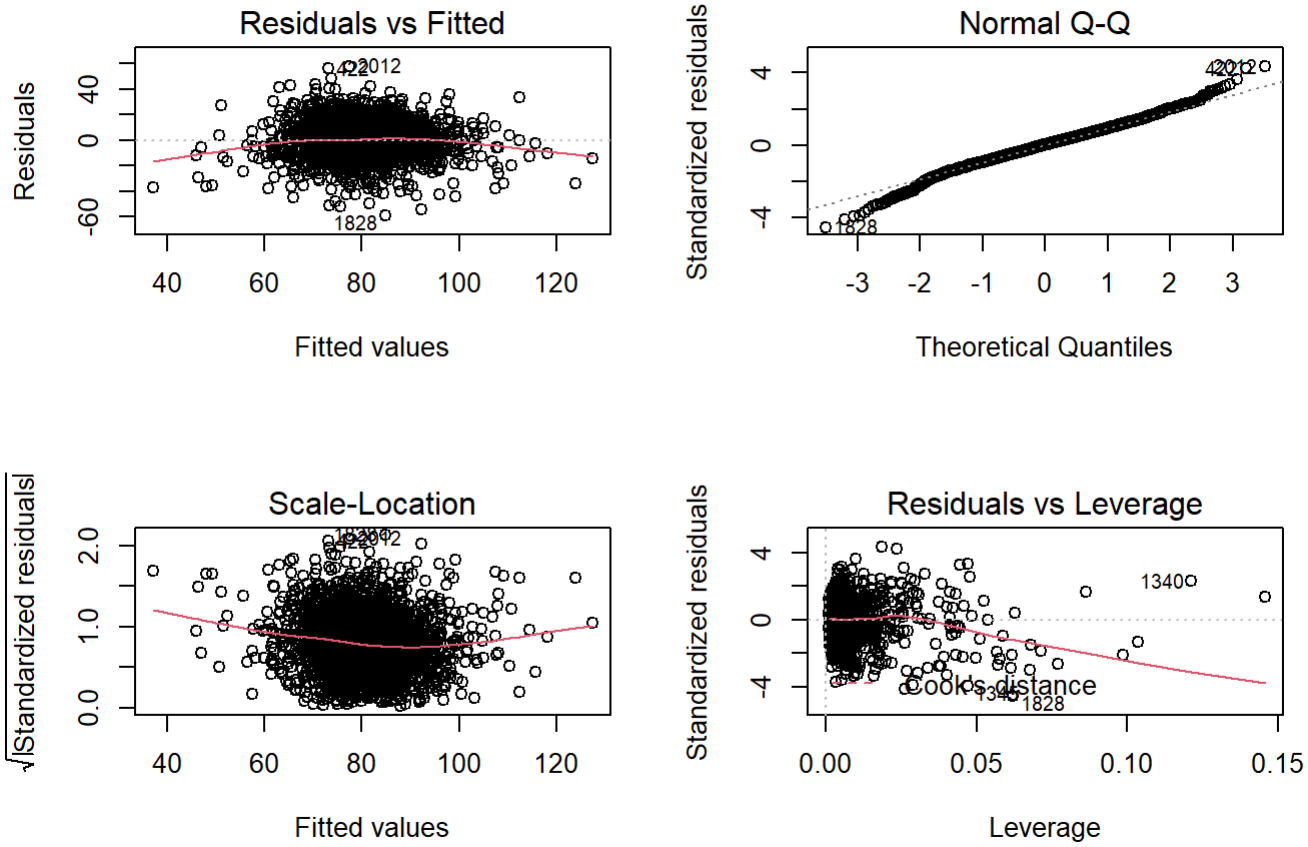
F(13,2262)	68.74

R^2	0.28
Adj. R^2	0.28

	Est.	S.E.	t val.	p
(Intercept)	7.84	5.39	1.45	0.15
INDEX	-0.00	0.00	-1.50	0.13
TEAM_BATTING_H	0.04	0.00	9.65	0.00
TEAM_BATTING_2B	-0.01	0.01	-1.30	0.19
TEAM_BATTING_3B	0.09	0.02	5.27	0.00
TEAM_BATTING_HR	0.09	0.03	3.21	0.00
TEAM_BATTING_BB	0.04	0.00	9.21	0.00
TEAM_BATTING_SO	0.01	0.00	1.28	0.20
TEAM_BASERUN_SB	0.04	0.00	10.12	0.00
TEAM_PITCHING_H	0.00	0.00	2.13	0.03
TEAM_PITCHING_HR	-0.03	0.02	-1.34	0.18
TEAM_PITCHING_BB	-0.02	0.01	-2.43	0.02
TEAM_PITCHING_SO	-0.01	0.00	-2.28	0.02
TEAM_FIELDING_E	-0.02	0.00	-7.11	0.00

Standard errors: OLS

```
# Plot results
par(mfrow=c(2,2))
plot(m1)
```



Model 2: Log transformation

Use of log transformation method which distributes skewness into a more “normally” distributed shape. I applied log transformation for highly skewed variables (less than -1 or greater than 1).

Note: Model 2 was not a successful model compared to model 1. There weren’t any significant changes between the two models therefore discarding this model.

```
# Checking skewness of dataset
sapply(baseball_df_final, function(x) skewness(x)) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"), font_size = 13)
```

	x
INDEX	0.0042149
TARGET_WINS	-0.3987232
TEAM_BATTING_H	1.5713335
TEAM_BATTING_2B	0.2151018
TEAM_BATTING_3B	1.1094652
TEAM_BATTING_HR	0.1860421
TEAM_BATTING_BB	-1.0257599

	x
TEAM_BATTING_SO	-0.2258649
TEAM_BASERUN_SB	1.8607589
TEAM_PITCHING_H	3.3490180
TEAM_PITCHING_HR	0.2877877
TEAM_PITCHING_BB	-2.4931634
TEAM_PITCHING_SO	-0.0237076
TEAM_FIELDING_E	2.1084199

```
# Doing log transformations from model 1
baseball_df_final_log <- baseball_df_final

# Applying log transformation for highly skewed variables
baseball_df_final_log$TEAM_BATTING_H <- log10(baseball_df_final_log$TEAM_BATTING_H + 1)
baseball_df_final_log$TEAM_BATTING_2B <- log10(baseball_df_final_log$TEAM_BATTING_2B + 1)
baseball_df_final_log$TEAM_PITCHING_H <- log10(baseball_df_final_log$TEAM_PITCHING_H + 1)
baseball_df_final_log$TEAM_PITCHING_BB <- log10(baseball_df_final_log$TEAM_PITCHING_BB + 1)
baseball_df_final_log$TEAM_FIELDING_E <- log10(baseball_df_final_log$TEAM_FIELDING_E + 1)
baseball_df_final_log$TEAM_BASERUN_SB <- log10(baseball_df_final_log$TEAM_BASERUN_SB + 1)

# Checking skewness
sapply(baseball_df_final_log, function(x) skewness(x)) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"), font_size = 13)
```

	x
INDEX	0.0042149
TARGET_WINS	-0.3987232
TEAM_BATTING_H	0.7835017
TEAM_BATTING_2B	-0.4041236
TEAM_BATTING_3B	1.1094652
TEAM_BATTING_HR	0.1860421
TEAM_BATTING_BB	-1.0257599
TEAM_BATTING_SO	-0.2258649
TEAM_BASERUN_SB	-0.1466066
TEAM_PITCHING_H	2.1429864
TEAM_PITCHING_HR	0.2877877
TEAM_PITCHING_BB	-13.6001569

	x
TEAM_PITCHING_SO	-0.0237076
TEAM_FIELDING_E	1.0877139

```
# Model 2 log
m2 <- lm(TARGET_WINS ~., data = baseball_df_final_log, na.action = na.omit)

#Summary
summ(m2)
```

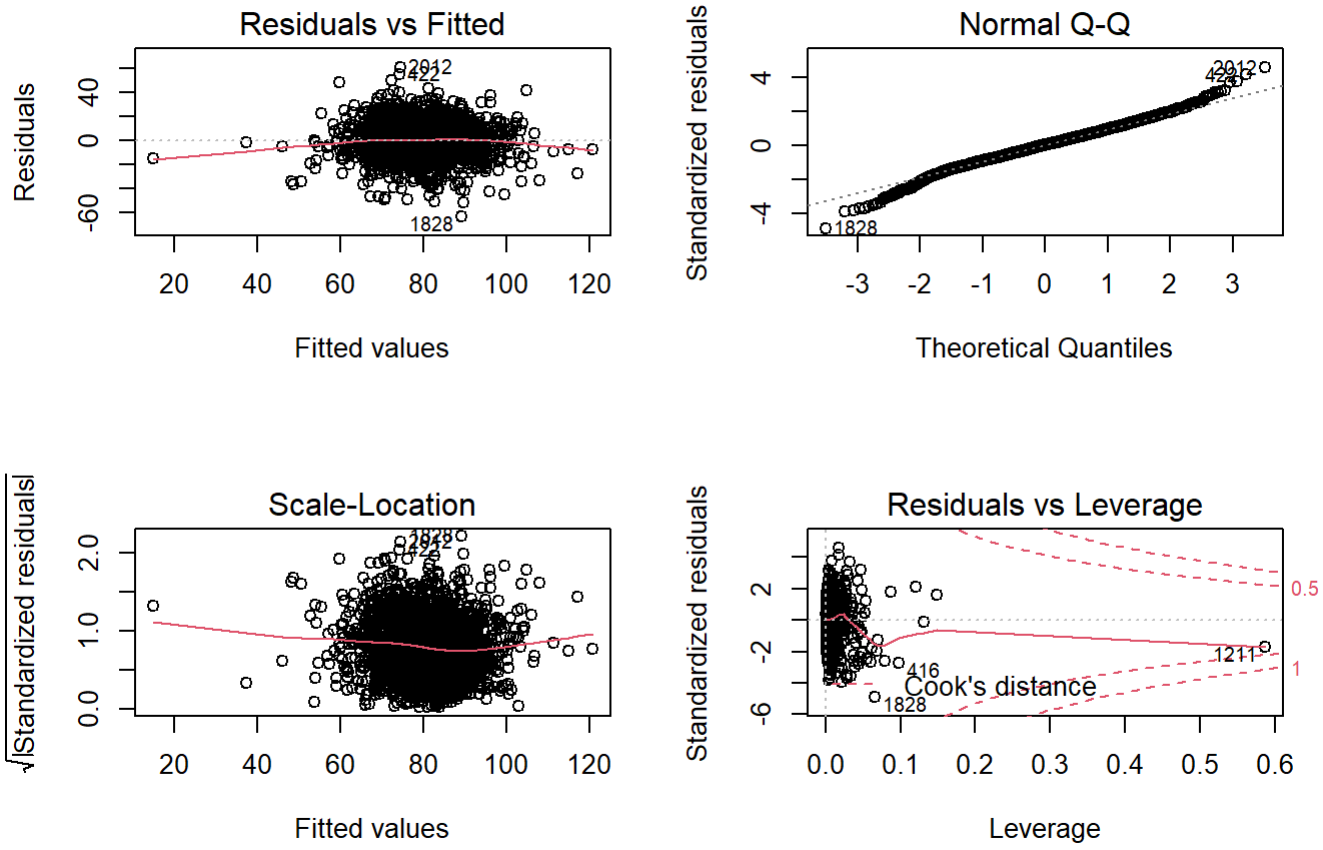
Observations	2276
Dependent variable	TARGET_WINS
Type	OLS linear regression

F(13,2262)	69.80
R^2	0.29
Adj. R^2	0.28

	Est.	S.E.	t val.	p
(Intercept)	-291.04	37.37	-7.79	0.00
INDEX	-0.00	0.00	-1.64	0.10
TEAM_BATTING_H	111.79	13.71	8.16	0.00
TEAM_BATTING_2B	-5.99	5.26	-1.14	0.25
TEAM_BATTING_3B	0.11	0.02	6.02	0.00
TEAM_BATTING_HR	0.09	0.03	3.41	0.00
TEAM_BATTING_BB	0.03	0.00	8.57	0.00
TEAM_BATTING_SO	-0.00	0.00	-0.15	0.88
TEAM_BASERUN_SB	13.72	1.25	10.94	0.00
TEAM_PITCHING_H	8.57	5.82	1.47	0.14
TEAM_PITCHING_HR	-0.03	0.02	-1.36	0.17
TEAM_PITCHING_BB	-2.12	4.27	-0.50	0.62
TEAM_PITCHING_SO	-0.01	0.00	-1.96	0.05
TEAM_FIELDING_E	-17.02	2.30	-7.40	0.00

Standard errors: OLS

```
# Plot results
par(mfrow=c(2,2))
plot(m2)
```



Model 3: Statistically significant

Focusing on statistically significant values chosen primarily from their R output.

```
# Model 3
m3 <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TE
AM_BASERUN_SB + TEAM_FIELDING_E, data = baseball_df_final)

# Summary
summ(m3)
```

Observations	2276
Dependent variable	TARGET_WINS
Type	OLS linear regression

F(6,2269)	141.91

R^2	0.27
Adj. R^2	0.27

	Est.	S.E.	t val.	p
(Intercept)	1.35	3.39	0.40	0.69
TEAM_BATTING_H	0.04	0.00	14.87	0.00
TEAM_BATTING_3B	0.09	0.02	5.29	0.00
TEAM_BATTING_HR	0.05	0.01	5.94	0.00
TEAM_BATTING_BB	0.03	0.00	12.26	0.00
TEAM_BASERUN_SB	0.04	0.00	10.56	0.00
TEAM_FIELDING_E	-0.02	0.00	-6.51	0.00

Standard errors: OLS

```
# par(mfrow=c(2,2))
# plot(m3)
```

Model 4: Backwards Elimination

Variables that are not statistically significant are removed to determine a best fit model.

```
# Model 4
m4 <- lm(TARGET_WINS ~ TEAM_BATTING_2B + TEAM_PITCHING_H +  TEAM_PITCHING_HR +  TEAM_PITCHING_BB
+ TEAM_PITCHING_SO, data = baseball_df_final)

# Summary
summ(m4)
```

Observations	2276
Dependent variable	TARGET_WINS
Type	OLS linear regression

F(5,2270)	82.61
R^2	0.15
Adj. R^2	0.15

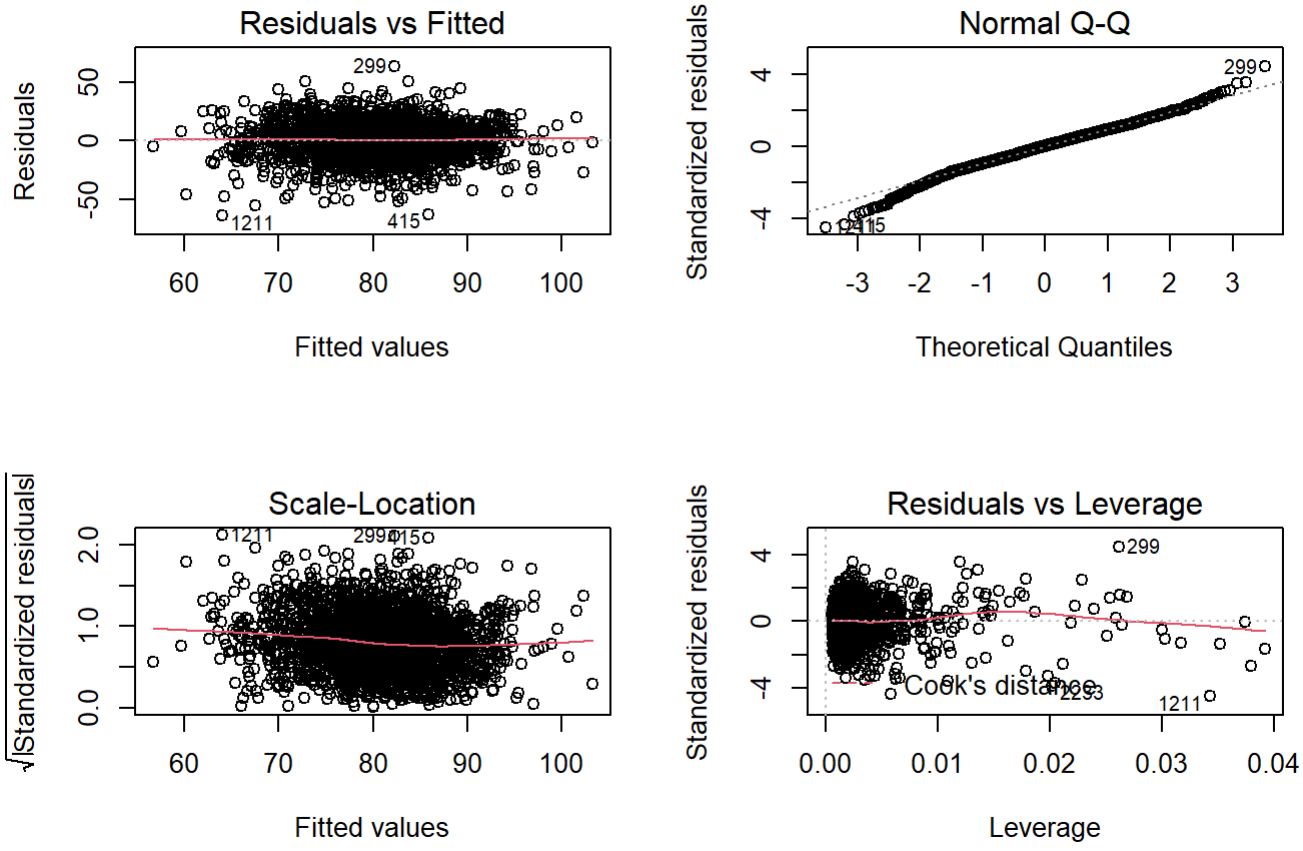
	Est.	S.E.	t val.	p
(Intercept)	44.79	3.44	13.01	0.00

Standard errors: OLS

	Est.	S.E.	t val.	p
TEAM_BATTING_2B	0.06	0.01	7.19	0.00
TEAM_PITCHING_H	0.01	0.00	8.09	0.00
TEAM_PITCHING_HR	0.06	0.01	8.65	0.00
TEAM_PITCHING_BB	0.03	0.01	6.09	0.00
TEAM_PITCHING_SO	-0.02	0.00	-9.97	0.00

Standard errors: OLS

```
# Plot results
par(mfrow=c(2,2))
plot(m4)
```



Model 5 - Power

Using a power model may be more effective considering each independent variable doesn't appear to have a truly linear relationship with wins. Here we create a model using a cubit for each independent variable.

```
# Model 5
m5 <- lm(TARGET_WINS ~ TEAM_BATTING_H + I(TEAM_BATTING_H^2) + I(TEAM_BATTING_H^3) +
           TEAM_BATTING_2B + I(TEAM_BATTING_2B^2) + I(TEAM_BATTING_2B^3) +
           TEAM_BATTING_3B + I(TEAM_BATTING_3B^2) + I(TEAM_BATTING_3B^3) +
           TEAM_BATTING_HR + I(TEAM_BATTING_HR^2) + I(TEAM_BATTING_HR^3) +
           TEAM_BATTING_BB + I(TEAM_BATTING_BB^2) + I(TEAM_BATTING_BB^3) +
           TEAM_BATTING_SO + I(TEAM_BATTING_SO^2) + I(TEAM_BATTING_SO^3) +
           TEAM_BASERUN_SB + I(TEAM_BASERUN_SB^2) + I(TEAM_BASERUN_SB^3) +
           TEAM_PITCHING_H + I(TEAM_PITCHING_H^2) + I(TEAM_PITCHING_H^3) +
           TEAM_PITCHING_HR + I(TEAM_PITCHING_HR^2) + I(TEAM_PITCHING_HR^3) +
           TEAM_PITCHING_BB + I(TEAM_PITCHING_BB^2) + I(TEAM_PITCHING_BB^3) +
           TEAM_PITCHING_SO + I(TEAM_PITCHING_SO^2) + I(TEAM_PITCHING_SO^3) +
           TEAM_FIELDING_E + I(TEAM_FIELDING_E^2) + I(TEAM_FIELDING_E^3), data = baseball_df_fin
al)

# Summary
summ(m5)
```

Observations	2276
Dependent variable	TARGET_WINS
Type	OLS linear regression

F(36,2239)	35.78
R²	0.37
Adj. R²	0.35

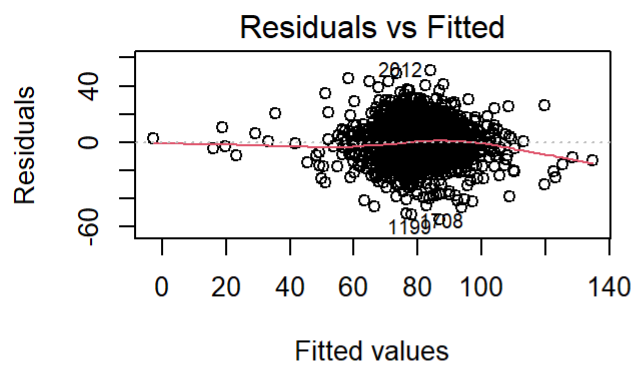
	Est.	S.E.	t val.	p
(Intercept)	-13.91	64.05	-0.22	0.83
TEAM_BATTING_H	0.07	0.13	0.53	0.59
I(TEAM_BATTING_H^2)	-0.00	0.00	-0.26	0.80
I(TEAM_BATTING_H^3)	0.00	0.00	0.41	0.68
TEAM_BATTING_2B	1.05	0.20	5.32	0.00
I(TEAM_BATTING_2B^2)	-0.00	0.00	-5.29	0.00
I(TEAM_BATTING_2B^3)	0.00	0.00	5.11	0.00
TEAM_BATTING_3B	-0.04	0.09	-0.44	0.66
I(TEAM_BATTING_3B^2)	0.00	0.00	2.76	0.01
I(TEAM_BATTING_3B^3)	-0.00	0.00	-3.98	0.00
TEAM_BATTING_HR	0.16	0.14	1.15	0.25

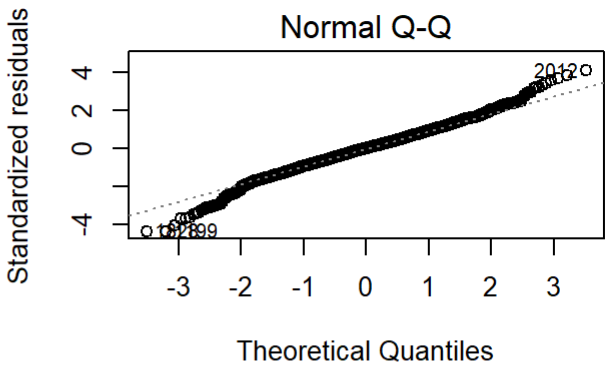
Standard errors: OLS

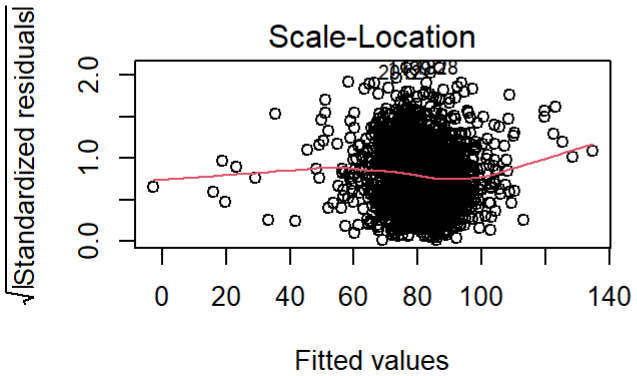
	Est.	S.E.	t val.	p
I(TEAM_BATTING_HR^2)	-0.00	0.00	-1.11	0.27
I(TEAM_BATTING_HR^3)	0.00	0.00	0.84	0.40
TEAM_BATTING_BB	0.21	0.05	4.15	0.00
I(TEAM_BATTING_BB^2)	-0.00	0.00	-3.45	0.00
I(TEAM_BATTING_BB^3)	0.00	0.00	3.26	0.00
TEAM_BATTING_SO	0.11	0.03	3.47	0.00
I(TEAM_BATTING_SO^2)	-0.00	0.00	-2.83	0.00
I(TEAM_BATTING_SO^3)	0.00	0.00	1.96	0.05
TEAM_BASERUN_SB	0.08	0.02	4.46	0.00
I(TEAM_BASERUN_SB^2)	-0.00	0.00	-1.21	0.23
I(TEAM_BASERUN_SB^3)	0.00	0.00	0.30	0.77
TEAM_PITCHING_H	-0.13	0.03	-4.36	0.00
I(TEAM_PITCHING_H^2)	0.00	0.00	3.89	0.00
I(TEAM_PITCHING_H^3)	-0.00	0.00	-3.30	0.00
TEAM_PITCHING_HR	-0.34	0.12	-2.75	0.01
I(TEAM_PITCHING_HR^2)	0.00	0.00	3.77	0.00
I(TEAM_PITCHING_HR^3)	-0.00	0.00	-3.87	0.00
TEAM_PITCHING_BB	0.10	0.11	0.92	0.36
I(TEAM_PITCHING_BB^2)	-0.00	0.00	-1.71	0.09
I(TEAM_PITCHING_BB^3)	0.00	0.00	2.01	0.04
TEAM_PITCHING_SO	-0.07	0.02	-2.88	0.00
I(TEAM_PITCHING_SO^2)	0.00	0.00	2.36	0.02
I(TEAM_PITCHING_SO^3)	-0.00	0.00	-2.22	0.03
TEAM_FIELDING_E	-0.14	0.04	-3.53	0.00
I(TEAM_FIELDING_E^2)	0.00	0.00	1.51	0.13
I(TEAM_FIELDING_E^3)	-0.00	0.00	-0.38	0.71

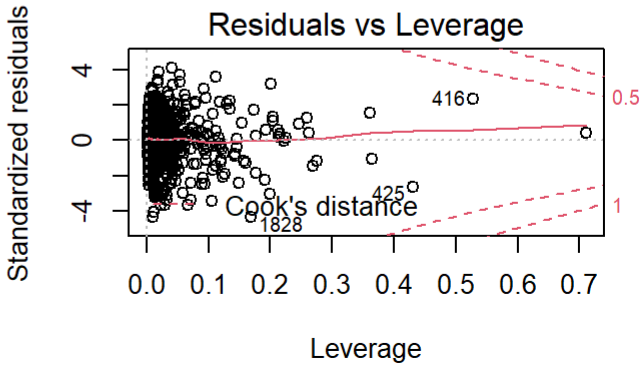
Standard errors: OLS

```
# Plot results
par(mfrow=c(2,2))
plot(m5)
```









Model 6 - Power with Reverse Elimination

Used reverse elimination on model 5 to remove variables with p-values higher than .05.

```
# Model 6
m6 <- lm(TARGET_WINS ~ TEAM_BATTING_H +
           TEAM_BATTING_2B + I(TEAM_BATTING_2B^2) + I(TEAM_BATTING_2B^3) +
           I(TEAM_BATTING_3B^2) + I(TEAM_BATTING_3B^3) +
           TEAM_BATTING_BB + I(TEAM_BATTING_BB^2) + I(TEAM_BATTING_BB^3) +
           TEAM_BATTING_SO + I(TEAM_BATTING_SO^2) +
           TEAM_BASERUN_SB + I(TEAM_BASERUN_SB^2) +
           TEAM_PITCHING_H + I(TEAM_PITCHING_H^2) + I(TEAM_PITCHING_H^3) +
           TEAM_PITCHING_HR + I(TEAM_PITCHING_HR^2) + I(TEAM_PITCHING_HR^3) +
           I(TEAM_PITCHING_BB^2) + I(TEAM_PITCHING_BB^3) +
           TEAM_PITCHING_SO + I(TEAM_PITCHING_SO^2) + I(TEAM_PITCHING_SO^3) +
           TEAM_FIELDING_E + I(TEAM_FIELDING_E^2), data = baseball_df_final)

# Summary
summ(m6)
```

Observations	2276
Dependent variable	TARGET_WINS
Type	OLS linear regression

F(26,2249)	49.21
R²	0.36
Adj. R²	0.36

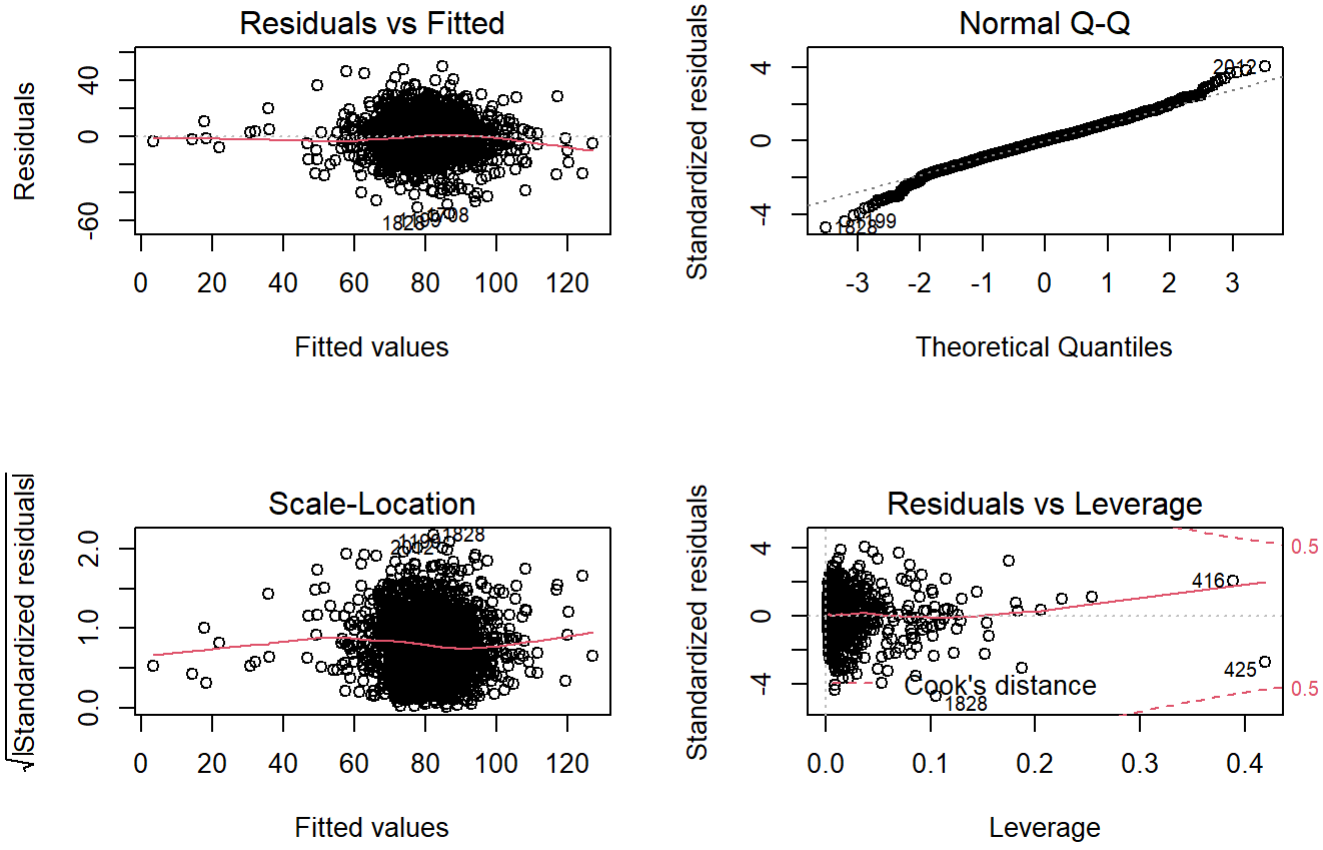
	Est.	S.E.	t val.	p
(Intercept)	6.69	23.75	0.28	0.78
TEAM_BATTING_H	0.05	0.00	13.00	0.00
TEAM_BATTING_2B	1.03	0.19	5.37	0.00
I(TEAM_BATTING_2B^2)	-0.00	0.00	-5.35	0.00
I(TEAM_BATTING_2B^3)	0.00	0.00	5.19	0.00
I(TEAM_BATTING_3B^2)	0.00	0.00	8.13	0.00
I(TEAM_BATTING_3B^3)	-0.00	0.00	-7.65	0.00
TEAM_BATTING_BB	0.27	0.05	5.75	0.00
I(TEAM_BATTING_BB^2)	-0.00	0.00	-4.63	0.00
I(TEAM_BATTING_BB^3)	0.00	0.00	4.22	0.00
TEAM_BATTING_SO	0.06	0.01	4.41	0.00
I(TEAM_BATTING_SO^2)	-0.00	0.00	-5.43	0.00
TEAM_BASERUN_SB	0.08	0.01	8.37	0.00
I(TEAM_BASERUN_SB^2)	-0.00	0.00	-4.35	0.00
TEAM_PITCHING_H	-0.14	0.03	-5.58	0.00
I(TEAM_PITCHING_H^2)	0.00	0.00	5.02	0.00
I(TEAM_PITCHING_H^3)	-0.00	0.00	-4.35	0.00
TEAM_PITCHING_HR	-0.19	0.04	-4.67	0.00
I(TEAM_PITCHING_HR^2)	0.00	0.00	6.14	0.00
I(TEAM_PITCHING_HR^3)	-0.00	0.00	-5.56	0.00
I(TEAM_PITCHING_BB^2)	-0.00	0.00	-3.20	0.00
I(TEAM_PITCHING_BB^3)	0.00	0.00	2.90	0.00
TEAM_PITCHING_SO	-0.04	0.02	-2.32	0.02
I(TEAM_PITCHING_SO^2)	0.00	0.00	1.50	0.13
I(TEAM_PITCHING_SO^3)	-0.00	0.00	-1.23	0.22
TEAM_FIELDING_E	-0.12	0.02	-7.76	0.00

Standard errors: OLS

	Est.	S.E.	t val.	p
I(TEAM_FIELDING_E^2)	0.00	0.00	6.40	0.00

Standard errors: OLS

```
# Plot results
par(mfrow=c(2,2))
plot(m6)
```



** Automated Reverse Elimination **

Based on residual analysis, all six models produced residuals that are heteroschedastic and not normally distributed. Therefore, we tried a different approach and ran the model without imputation and without transformation. We performed backwards elimination using an automated function to generate a new function call based on removing the parameter with the highest p-value (see Appendix A for R code and full model results).

```
# Initialize list to store model results
lmlist <- vector(mode='list', length=0)

# Build model call, start with full model (without index field)
model_params <- 'TARGET_WINS ~ . - INDEX'

# Iterate until p-values are less than 0.05
while (T) {

  # Lengthen the model results list
  i <- length(lmlist)
  i <- i + 1
  length(lmlist) <- i
  print('------------------------------------------------------------')
  print(model_params)

  # Run the model
  lmlist[[i]] <- lm(model_params, train_df, na.action=na.exclude)
  lmsum <- summary(lmlist[[i]])
  print(summ(lmlist[[i]]))

  # Get the p-values in descending order, check to see if it's less than 0.05
  pvals <- sort(lmsum$coefficients[,4], decreasing=T)
  if (pvals[1] <= 0.05) {

    # Break out of the while loop because the largest p-value is statistically significant
    break

  } else {

    # Find the column with the highest p-value
    remove_col <- names(pvals[1])

    # Change the model call, removing the column with the highest p-value
    model_params <- paste0(model_params, ' - ', remove_col)

  }
}
```

```
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(15,175) = 14.27, p = 0.00
## R² = 0.55
## Adj. R² = 0.51
##
## Standard errors: OLS
## -------------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              60.29   19.68     3.06   0.00
## TEAM_BATTING_H            1.91    2.76     0.69   0.49
## TEAM_BATTING_2B           0.03    0.03     0.87   0.38
## TEAM_BATTING_3B          -0.10    0.08    -1.31   0.19
## TEAM_BATTING_HR          -4.84   10.51    -0.46   0.65
## TEAM_BATTING_BB          -4.46    3.64    -1.23   0.22
## TEAM_BATTING_SO           0.34    2.60     0.13   0.90
## TEAM_BASERUN_SB           0.03    0.03     1.15   0.25
## TEAM_BASERUN_CS          -0.01    0.07    -0.15   0.88
## TEAM_BATTING_HBP          0.08    0.05     1.66   0.10
## TEAM_PITCHING_H          -1.89    2.76    -0.68   0.49
## TEAM_PITCHING_HR          4.93   10.51     0.47   0.64
## TEAM_PITCHING_BB          4.51    3.63     1.24   0.22
## TEAM_PITCHING_SO         -0.37    2.60    -0.14   0.89
## TEAM_FIELDING_E          -0.17    0.04    -4.16   0.00
## TEAM_FIELDING_DP         -0.11    0.04    -2.96   0.00
## -------------------------------------------------------
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(14,176) = 15.37, p = 0.00
## R² = 0.55
## Adj. R² = 0.51
##
## Standard errors: OLS
## -------------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              60.49   19.56     3.09   0.00
## TEAM_BATTING_H            2.17    1.93     1.13   0.26
## TEAM_BATTING_2B           0.03    0.03     0.87   0.39
```

```
## TEAM_BATTING_3B          -0.10    0.08    -1.31   0.19
## TEAM_BATTING_HR          -4.93   10.46    -0.47   0.64
## TEAM_BATTING_BB          -4.46    3.63    -1.23   0.22
## TEAM_BASERUN_SB           0.03    0.03     1.15   0.25
## TEAM_BASERUN_CS          -0.01    0.07    -0.15   0.88
## TEAM_BATTING_HBP          0.08    0.05     1.66   0.10
## TEAM_PITCHING_H          -2.15    1.93    -1.11   0.27
## TEAM_PITCHING_HR          5.01   10.46     0.48   0.63
## TEAM_PITCHING_BB          4.51    3.62     1.25   0.21
## TEAM_PITCHING_SO         -0.03    0.01    -4.29   0.00
## TEAM_FIELDING_E          -0.17    0.04    -4.17   0.00
## TEAM_FIELDING_DP         -0.11    0.04    -2.97   0.00
## ----------------------------------------------------
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO - TEAM_BASERUN_CS"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(13,177) = 16.64, p = 0.00
## R² = 0.55
## Adj. R² = 0.52
##
## Standard errors: OLS
## ----------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              60.13   19.36     3.11   0.00
## TEAM_BATTING_H            2.19    1.92     1.14   0.26
## TEAM_BATTING_2B           0.03    0.03     0.87   0.39
## TEAM_BATTING_3B          -0.10    0.08    -1.35   0.18
## TEAM_BATTING_HR          -4.96   10.43    -0.48   0.63
## TEAM_BATTING_BB          -4.50    3.61    -1.25   0.21
## TEAM_BASERUN_SB           0.03    0.02     1.32   0.19
## TEAM_BATTING_HBP          0.08    0.05     1.67   0.10
## TEAM_PITCHING_H          -2.17    1.92    -1.13   0.26
## TEAM_PITCHING_HR          5.05   10.43     0.48   0.63
## TEAM_PITCHING_BB          4.55    3.60     1.26   0.21
## TEAM_PITCHING_SO         -0.03    0.01    -4.30   0.00
## TEAM_FIELDING_E          -0.17    0.04    -4.32   0.00
## TEAM_FIELDING_DP         -0.11    0.04    -2.99   0.00
## ----------------------------------------------------
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO - TEAM_BASERUN_CS - TEAM_BATTING_HR"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
```

```
## F(12,178) = 18.09, p = 0.00
## R² = 0.55
## Adj. R² = 0.52
##
## Standard errors: OLS
## ------------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              60.23   19.32     3.12   0.00
## TEAM_BATTING_H            1.54    1.35     1.14   0.26
## TEAM_BATTING_2B           0.03    0.03     0.89   0.38
## TEAM_BATTING_3B          -0.11    0.08    -1.40   0.16
## TEAM_BATTING_BB          -4.37    3.59    -1.22   0.23
## TEAM_BASERUN_SB           0.03    0.02     1.44   0.15
## TEAM_BATTING_HBP          0.08    0.05     1.68   0.09
## TEAM_PITCHING_H          -1.52    1.35    -1.12   0.26
## TEAM_PITCHING_HR          0.09    0.03     3.48   0.00
## TEAM_PITCHING_BB          4.42    3.59     1.23   0.22
## TEAM_PITCHING_SO         -0.03    0.01    -4.32   0.00
## TEAM_FIELDING_E          -0.17    0.04    -4.35   0.00
## TEAM_FIELDING_DP         -0.11    0.04    -2.98   0.00
## ------------------------------------------------------
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO - TEAM_BASERUN_CS - TEAM_BATTING_HR - TEAM_BAT
TING_2B"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(11,179) = 19.69, p = 0.00
## R² = 0.55
## Adj. R² = 0.52
##
## Standard errors: OLS
## ------------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              59.11   19.27     3.07   0.00
## TEAM_BATTING_H            1.51    1.35     1.12   0.27
## TEAM_BATTING_3B          -0.11    0.08    -1.44   0.15
## TEAM_BATTING_BB          -4.25    3.58    -1.18   0.24
## TEAM_BASERUN_SB           0.03    0.02     1.24   0.21
## TEAM_BATTING_HBP          0.08    0.05     1.70   0.09
## TEAM_PITCHING_H          -1.48    1.35    -1.09   0.28
## TEAM_PITCHING_HR          0.09    0.03     3.40   0.00
## TEAM_PITCHING_BB          4.30    3.58     1.20   0.23
## TEAM_PITCHING_SO         -0.03    0.01    -4.23   0.00
## TEAM_FIELDING_E          -0.18    0.04    -4.42   0.00
## TEAM_FIELDING_DP         -0.11    0.04    -3.03   0.00
## ------------------------------------------------------
```

```
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO - TEAM_BASERUN_CS - TEAM_BATTING_HR - TEAM_BAT
TING_2B - TEAM_PITCHING_H"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(10,180) = 21.51, p = 0.00
## R² = 0.54
## Adj. R² = 0.52
##
## Standard errors: OLS
## -------------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              56.95   19.18     2.97   0.00
## TEAM_BATTING_H            0.03    0.01     2.67   0.01
## TEAM_BATTING_3B          -0.11    0.08    -1.47   0.14
## TEAM_BATTING_BB          -0.37    0.54    -0.68   0.50
## TEAM_BASERUN_SB           0.03    0.02     1.22   0.22
## TEAM_BATTING_HBP          0.08    0.05     1.61   0.11
## TEAM_PITCHING_HR          0.09    0.03     3.47   0.00
## TEAM_PITCHING_BB          0.42    0.54     0.78   0.43
## TEAM_PITCHING_SO         -0.03    0.01    -4.22   0.00
## TEAM_FIELDING_E          -0.18    0.04    -4.48   0.00
## TEAM_FIELDING_DP         -0.10    0.04    -2.92   0.00
## -------------------------------------------------------
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO - TEAM_BASERUN_CS - TEAM_BATTING_HR - TEAM_BAT
TING_2B - TEAM_PITCHING_H - TEAM_BATTING_BB"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(9,181) = 23.92, p = 0.00
## R² = 0.54
## Adj. R² = 0.52
##
## Standard errors: OLS
## -------------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              57.82   19.11     3.03   0.00
## TEAM_BATTING_H            0.03    0.01     2.63   0.01
## TEAM_BATTING_3B          -0.11    0.08    -1.49   0.14
## TEAM_BASERUN_SB           0.03    0.02     1.25   0.21
## TEAM_BATTING_HBP          0.08    0.05     1.63   0.11
## TEAM_PITCHING_HR          0.09    0.03     3.46   0.00
```

```
## TEAM_PITCHING_BB          0.05    0.01     5.80   0.00
## TEAM_PITCHING_SO         -0.03    0.01    -4.24   0.00
## TEAM_FIELDING_E          -0.18    0.04    -4.45   0.00
## TEAM_FIELDING_DP         -0.11    0.04    -2.98   0.00
## ----------------------------------------------------
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO - TEAM_BASERUN_CS - TEAM_BATTING_HR - TEAM_BAT
TING_2B - TEAM_PITCHING_H - TEAM_BATTING_BB - TEAM_BASERUN_SB"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(8,182) = 26.63, p = 0.00
## R² = 0.54
## Adj. R² = 0.52
##
## Standard errors: OLS
## ----------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              58.85   19.12     3.08   0.00
## TEAM_BATTING_H            0.03    0.01     2.83   0.01
## TEAM_BATTING_3B          -0.10    0.08    -1.38   0.17
## TEAM_BATTING_HBP          0.08    0.05     1.58   0.11
## TEAM_PITCHING_HR          0.08    0.02     3.27   0.00
## TEAM_PITCHING_BB          0.05    0.01     5.80   0.00
## TEAM_PITCHING_SO         -0.03    0.01    -4.22   0.00
## TEAM_FIELDING_E          -0.17    0.04    -4.34   0.00
## TEAM_FIELDING_DP         -0.11    0.04    -3.23   0.00
## ----------------------------------------------------
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO - TEAM_BASERUN_CS - TEAM_BATTING_HR - TEAM_BAT
TING_2B - TEAM_PITCHING_H - TEAM_BATTING_BB - TEAM_BASERUN_SB - TEAM_BATTING_3B"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(7,183) = 30.02, p = 0.00
## R² = 0.53
## Adj. R² = 0.52
##
## Standard errors: OLS
## ----------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              60.95   19.10     3.19   0.00
## TEAM_BATTING_H            0.03    0.01     2.52   0.01
## TEAM_BATTING_HBP          0.09    0.05     1.80   0.07
```

```
## TEAM_PITCHING_HR          0.09    0.02     3.74   0.00
## TEAM_PITCHING_BB          0.06    0.01     6.03   0.00
## TEAM_PITCHING_SO         -0.03    0.01    -4.31   0.00
## TEAM_FIELDING_E          -0.17    0.04    -4.34   0.00
## TEAM_FIELDING_DP         -0.12    0.04    -3.39   0.00
## ----------------------------------------------------
## [1] "------------------------------------------------------------"
## [1] "TARGET_WINS ~ . - INDEX - TEAM_BATTING_SO - TEAM_BASERUN_CS - TEAM_BATTING_HR - TEAM_BAT
TING_2B - TEAM_PITCHING_H - TEAM_BATTING_BB - TEAM_BASERUN_SB - TEAM_BATTING_3B - TEAM_BATTING_H
BP"
## MODEL INFO:
## Observations: 191 (2085 missing obs. deleted)
## Dependent Variable: TARGET_WINS
## Type: OLS linear regression
##
## MODEL FIT:
## F(6,184) = 34.07, p = 0.00
## R² = 0.53
## Adj. R² = 0.51
##
## Standard errors: OLS
## ----------------------------------------------------
##                            Est.    S.E.   t val.      p
## ---------------------- ------- ------- -------- ------
## (Intercept)              63.47   19.17     3.31   0.00
## TEAM_BATTING_H            0.03    0.01     2.54   0.01
## TEAM_PITCHING_HR          0.09    0.02     3.81   0.00
## TEAM_PITCHING_BB          0.06    0.01     5.93   0.00
## TEAM_PITCHING_SO         -0.03    0.01    -4.02   0.00
## TEAM_FIELDING_E          -0.17    0.04    -4.36   0.00
## TEAM_FIELDING_DP         -0.12    0.04    -3.44   0.00
## ----------------------------------------------------
```

Model Selection

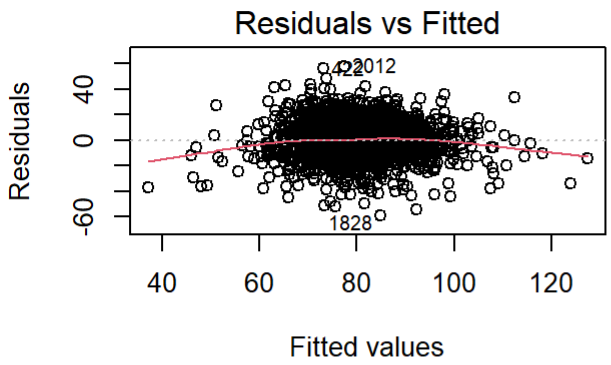
First, create functions to calculate mean squared error and perform model analysis given a linear model.

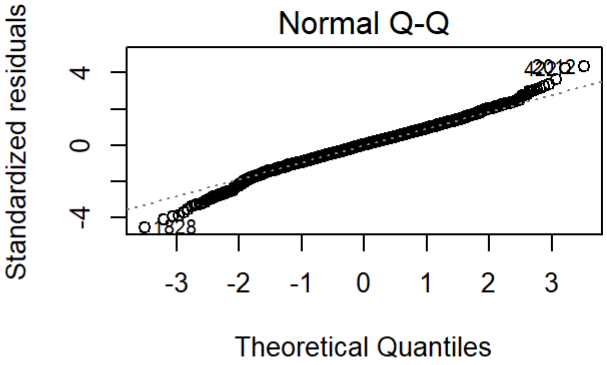
```
# Define function to calculate mean squared error
mse <- function(lmod) {
  return(mean((summary(lmod))$residuals ^ 2))
}

# Define function to aid in model analysis
ModelAnalysis <- function(lmod, x) {

  # Plot residuals
  print('--------------------------------------------------')
  print(lmod$call)
  par(mfrow=c(2,2))
  plot(lmod)

  # Shapiro test to determine normality of residuals
  # null hypothesis: the residuals are normal
  # p-value is small, so reject the null
  # i.e., the residuals are not normal
  st <- shapiro.test(lmod$residuals)
  if (st$p.value <= 0.05) {
    print(paste0("Shapiro test for normality: The p-value of ", st$p.value, " is <= 0.05, so rej
ect the null; i.e., the residuals are NOT normal."))
  } else {
    print(paste0("Shapiro test for normality: The p-value of ", st$p.value, " is > 0.05, so do n
ot reject the null; i.e., the residuals are normal."))
  }

  # Breusch-Pagan test to determine homoschedasticity of residuals
  bp <- bptest(lmod)
  if (bp$p.value > 0.05 & bp$statistic < 10) {
      print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ", bp$p.value, " is
> 0.05 and the test statistic of ", bp$statistic,
          " is < 10, so don't reject the null; i.e., the residuals are homoschedastic."))
  } else if (bp$p.value <= 0.05) {
      print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ", bp$p.value, " is
<= 0.05 and the test statistic is ", bp$statistic,
          ", so reject the null; i.e., the residuals are heteroschedastic."))
  } else {
      print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ", bp$p.value, " an
d test statistic of ", bp$statistic,
          " are inconclusive, so homoschedasticity can't be determined using this test."))
  }

  # Adjusted R-squared
  print(paste0('Adjusted R-squared value: ', (summary(lmod)$adj.r.squared)))

  # Mean squared error
  m <- mse(lmod)
  print(paste0('Mean squared error: ', mse(lmod)))

}
```

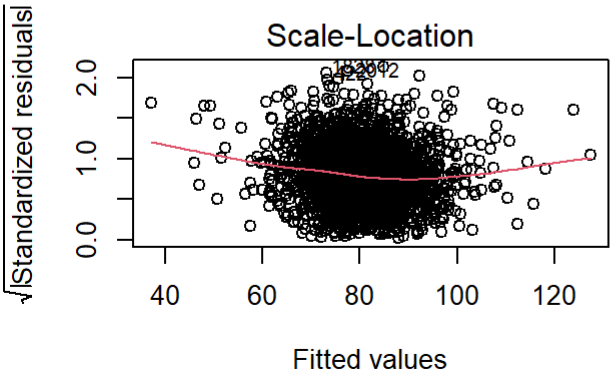
Model analysis on each of the 6 models.

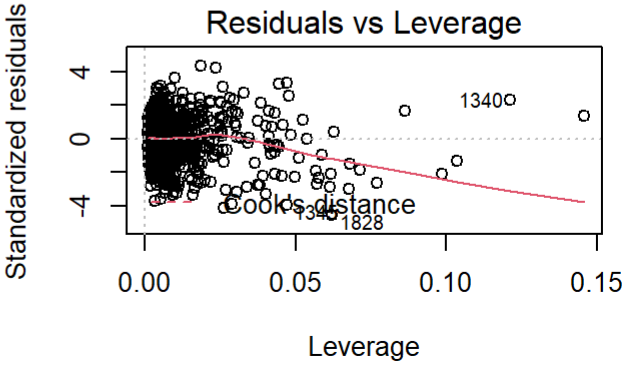
```
# Model analysis
ModelAnalysis(m1)
```

```
## [1] "-------------------------------------------------"
## lm(formula = TARGET_WINS ~ ., data = baseball_df_final, na.action = na.omit)
```





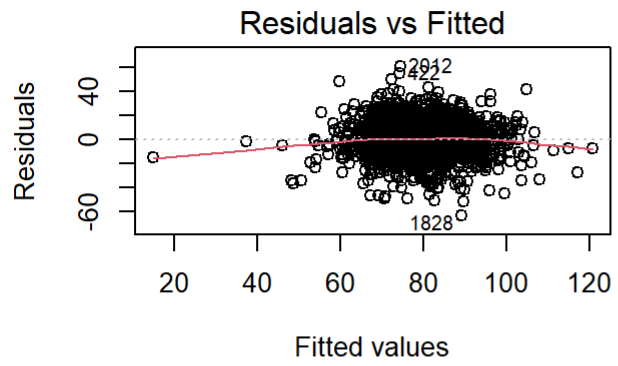


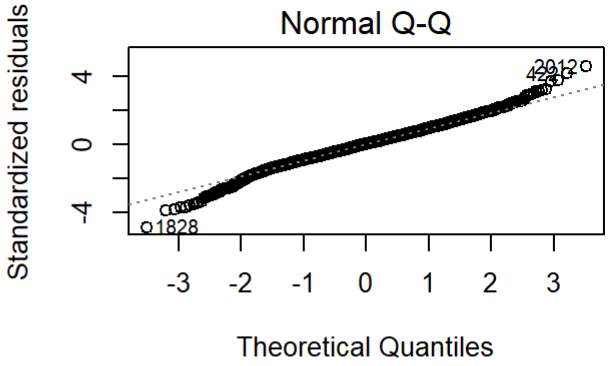


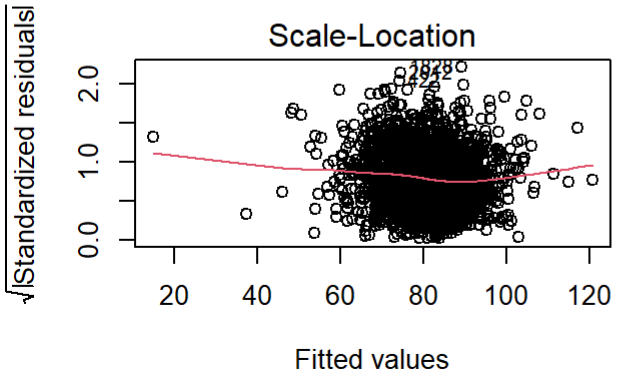

```
## [1] "Shapiro test for normality: The p-value of 1.04556378279577e-10 is <= 0.05, so reject th
e null; i.e., the residuals are NOT normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 7.02419507641643e-59 is <= 0.05
and the test statistic is 312.104580852486, so reject the null; i.e., the residuals are heterosc
hedastic."
## [1] "Adjusted R-squared value: 0.279067578558492"
## [1] "Mean squared error: 177.784835300702"
```

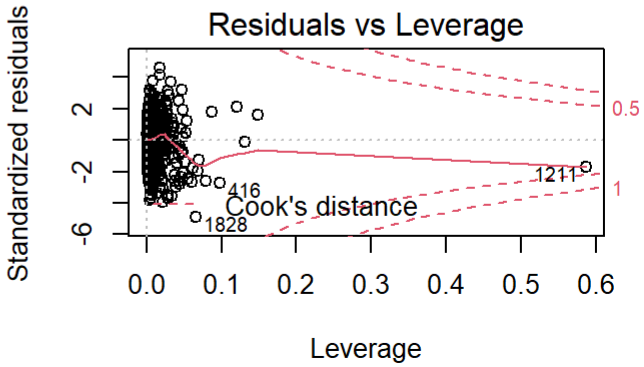
```
ModelAnalysis(m2)
```

```
## [1] "-------------------------------------------------"
## lm(formula = TARGET_WINS ~ ., data = baseball_df_final_log, na.action = na.omit)
```





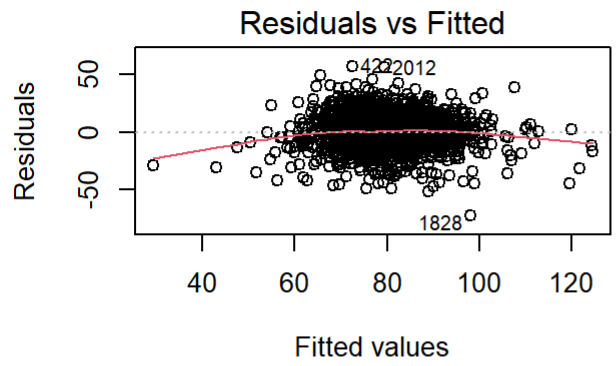


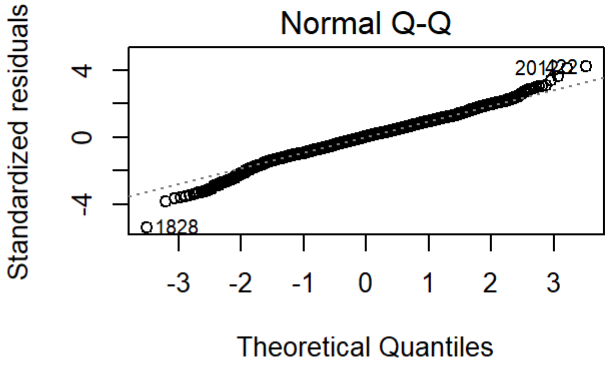


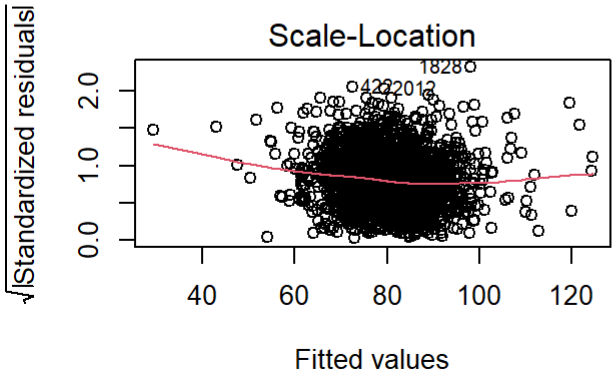

```
## [1] "Shapiro test for normality: The p-value of 7.305674163284e-11 is <= 0.05, so reject the
null; i.e., the residuals are NOT normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 4.06673586911508e-56 is <= 0.05
and the test statistic is 298.910144956467, so reject the null; i.e., the residuals are heterosc
hedastic."
## [1] "Adjusted R-squared value: 0.282197596618816"
## [1] "Mean squared error: 177.012960255562"
```

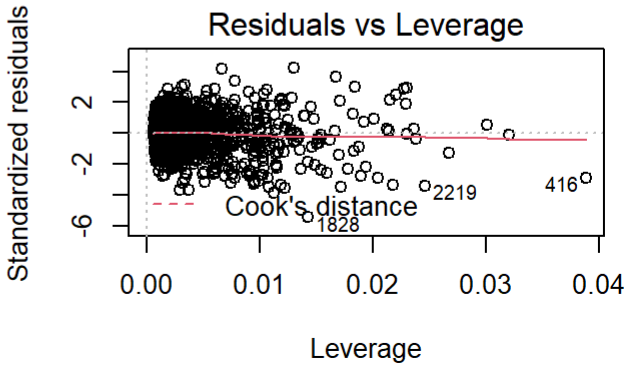
```
ModelAnalysis(m3)
```

```
## [1] "-------------------------------------------------"
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B +
##     TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_FIELDING_E,
##     data = baseball_df_final)
```





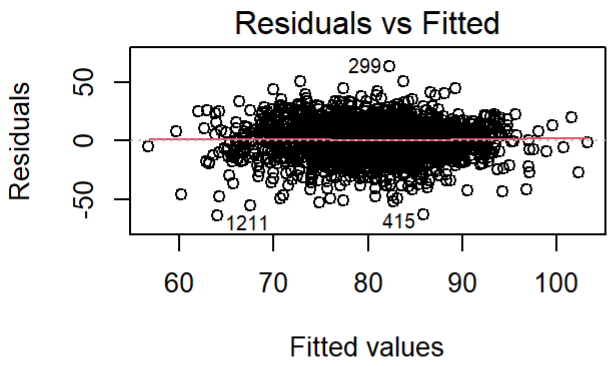


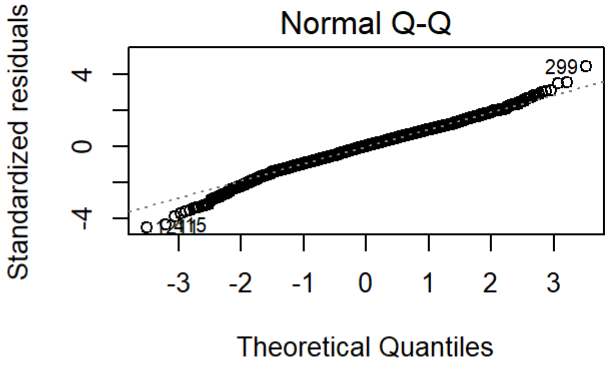


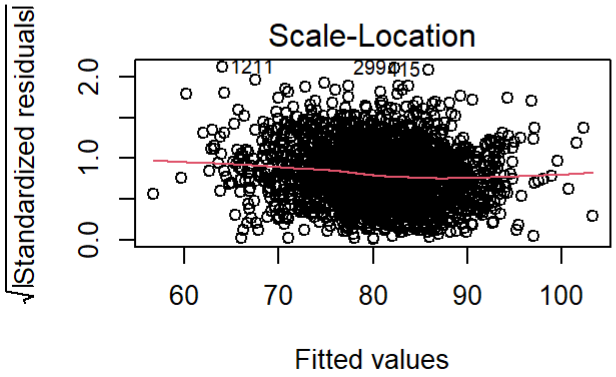

```
## [1] "Shapiro test for normality: The p-value of 3.41098138059316e-11 is <= 0.05, so reject th
e null; i.e., the residuals are NOT normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 8.38203287406739e-55 is <= 0.05
and the test statistic is 267.256057804979, so reject the null; i.e., the residuals are heterosc
hedastic."
## [1] "Adjusted R-squared value: 0.270945335482595"
## [1] "Mean squared error: 180.344185660252"
```

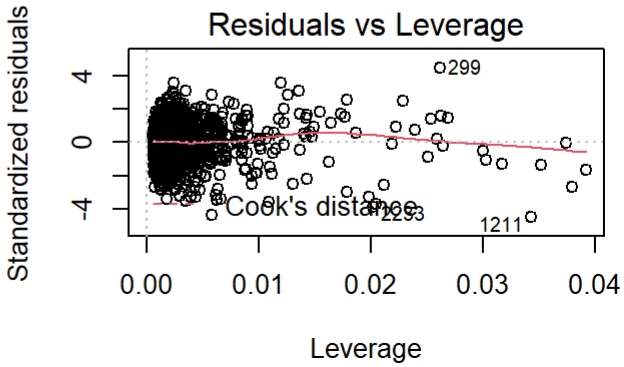
```
ModelAnalysis(m4)
```

```
## [1] "--------------------------------------------------"
## lm(formula = TARGET_WINS ~ TEAM_BATTING_2B + TEAM_PITCHING_H +
##     TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_PITCHING_SO, data = baseball_df_final)
```





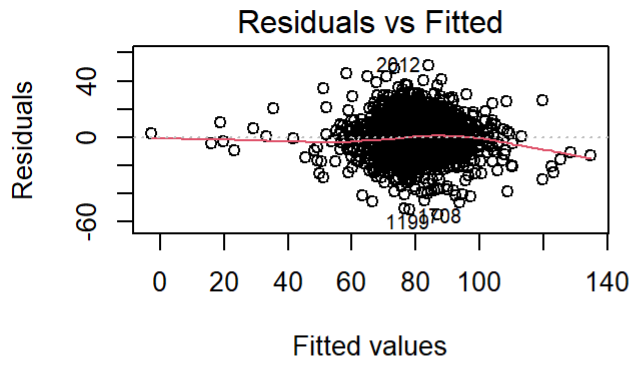


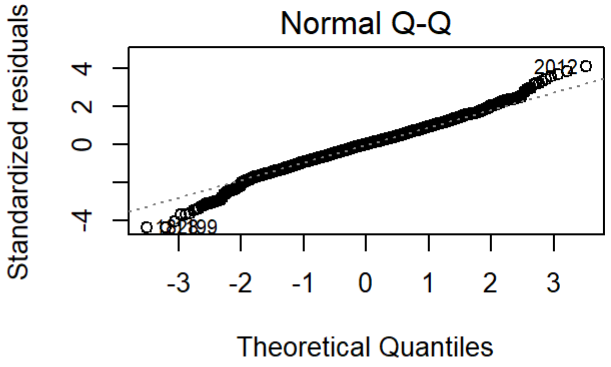


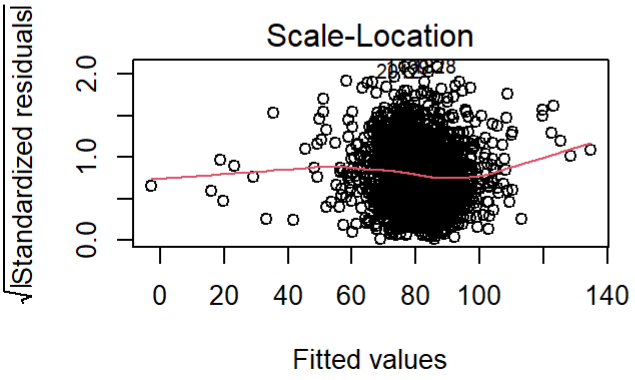

```
## [1] "Shapiro test for normality: The p-value of 6.56700377590433e-10 is <= 0.05, so reject th
e null; i.e., the residuals are NOT normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 2.21852754767312e-44 is <= 0.05
and the test statistic is 214.518079436075, so reject the null; i.e., the residuals are heterosc
hedastic."
## [1] "Adjusted R-squared value: 0.152084762048857"
## [1] "Mean squared error: 209.838828780989"
```

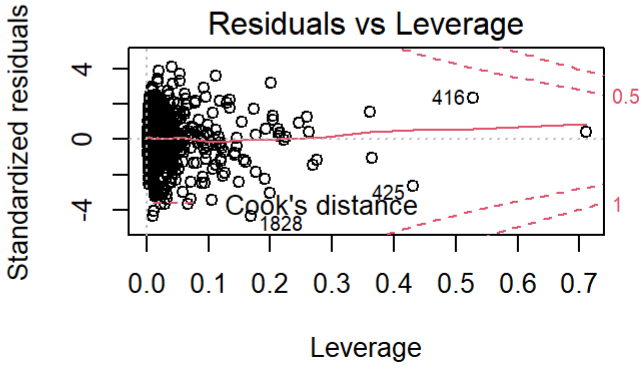
```
ModelAnalysis(m5)
```

```
## [1] "-------------------------------------------------"
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + I(TEAM_BATTING_H^2) +
##     I(TEAM_BATTING_H^3) + TEAM_BATTING_2B + I(TEAM_BATTING_2B^2) +
##     I(TEAM_BATTING_2B^3) + TEAM_BATTING_3B + I(TEAM_BATTING_3B^2) +
##     I(TEAM_BATTING_3B^3) + TEAM_BATTING_HR + I(TEAM_BATTING_HR^2) +
##     I(TEAM_BATTING_HR^3) + TEAM_BATTING_BB + I(TEAM_BATTING_BB^2) +
##     I(TEAM_BATTING_BB^3) + TEAM_BATTING_SO + I(TEAM_BATTING_SO^2) +
##     I(TEAM_BATTING_SO^3) + TEAM_BASERUN_SB + I(TEAM_BASERUN_SB^2) +
##     I(TEAM_BASERUN_SB^3) + TEAM_PITCHING_H + I(TEAM_PITCHING_H^2) +
##     I(TEAM_PITCHING_H^3) + TEAM_PITCHING_HR + I(TEAM_PITCHING_HR^2) +
##     I(TEAM_PITCHING_HR^3) + TEAM_PITCHING_BB + I(TEAM_PITCHING_BB^2) +
##     I(TEAM_PITCHING_BB^3) + TEAM_PITCHING_SO + I(TEAM_PITCHING_SO^2) +
##     I(TEAM_PITCHING_SO^3) + TEAM_FIELDING_E + I(TEAM_FIELDING_E^2) +
##     I(TEAM_FIELDING_E^3), data = baseball_df_final)
```





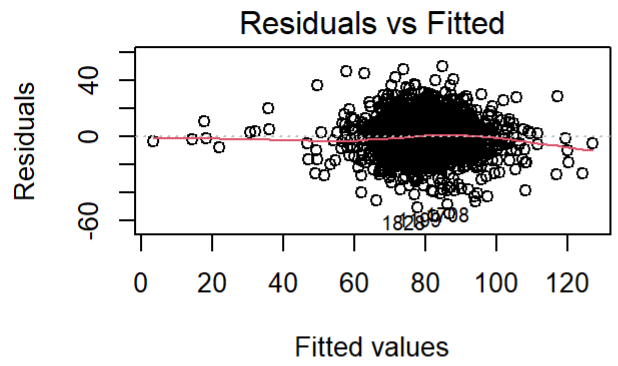


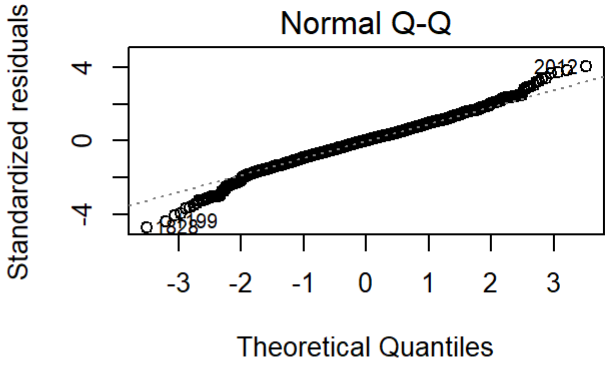


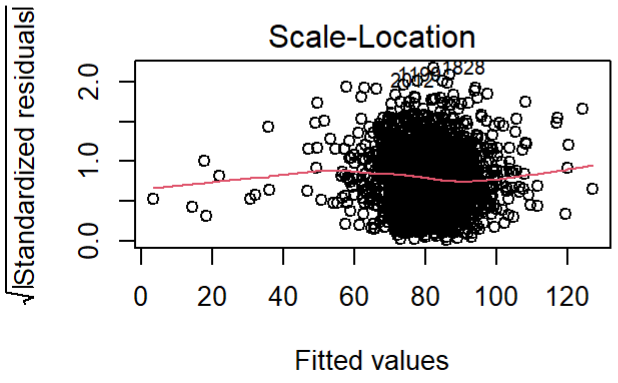

```
## [1] "Shapiro test for normality: The p-value of 5.98340064191167e-10 is <= 0.05, so reject th
e null; i.e., the residuals are NOT normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 1.52321013284566e-46 is <= 0.05
and the test statistic is 316.380936352325, so reject the null; i.e., the residuals are heterosc
hedastic."
## [1] "Adjusted R-squared value: 0.354968093291786"
## [1] "Mean squared error: 157.450066195233"
```

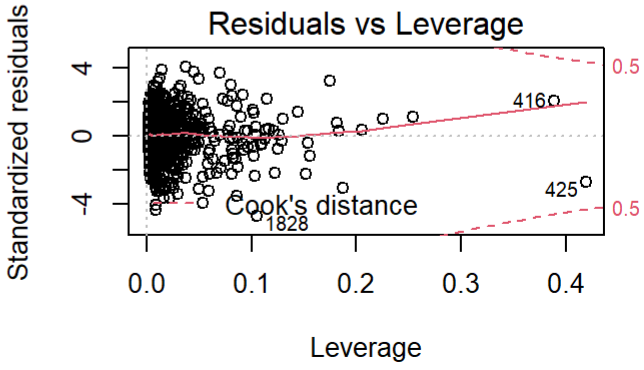
```
ModelAnalysis(m6)
```

```
## [1] "-------------------------------------------------"
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     I(TEAM_BATTING_2B^2) + I(TEAM_BATTING_2B^3) + I(TEAM_BATTING_3B^2) +
##     I(TEAM_BATTING_3B^3) + TEAM_BATTING_BB + I(TEAM_BATTING_BB^2) +
##     I(TEAM_BATTING_BB^3) + TEAM_BATTING_SO + I(TEAM_BATTING_SO^2) +
##     TEAM_BASERUN_SB + I(TEAM_BASERUN_SB^2) + TEAM_PITCHING_H +
##     I(TEAM_PITCHING_H^2) + I(TEAM_PITCHING_H^3) + TEAM_PITCHING_HR +
##     I(TEAM_PITCHING_HR^2) + I(TEAM_PITCHING_HR^3) + I(TEAM_PITCHING_BB^2) +
##     I(TEAM_PITCHING_BB^3) + TEAM_PITCHING_SO + I(TEAM_PITCHING_SO^2) +
##     I(TEAM_PITCHING_SO^3) + TEAM_FIELDING_E + I(TEAM_FIELDING_E^2),
##     data = baseball_df_final)
```





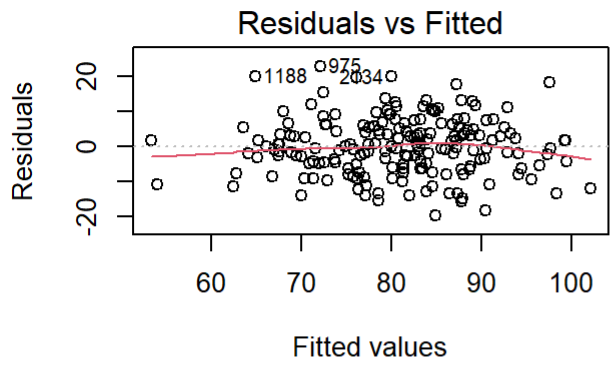


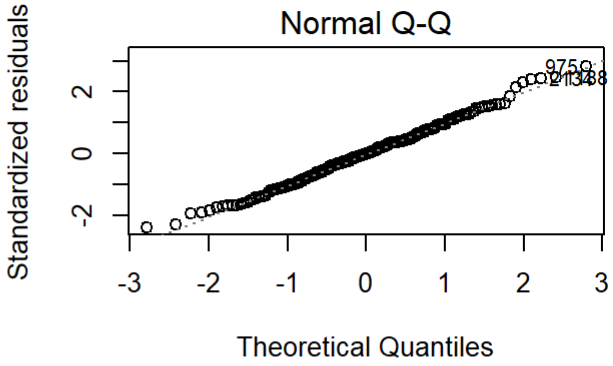


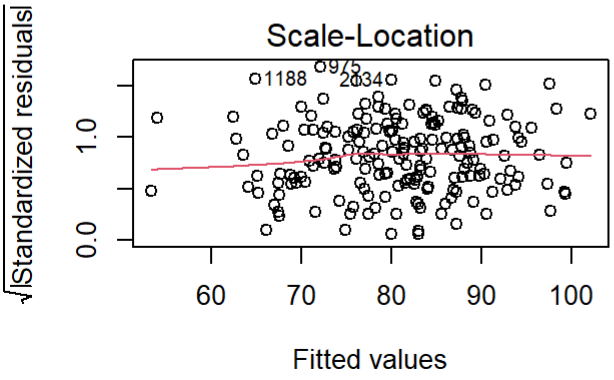

```
## [1] "Shapiro test for normality: The p-value of 1.7370518083911e-10 is <= 0.05, so reject the
null; i.e., the residuals are NOT normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 1.35597031615975e-41 is <= 0.05
and the test statistic is 265.763199270119, so reject the null; i.e., the residuals are heterosc
hedastic."
## [1] "Adjusted R-squared value: 0.355219801227876"
## [1] "Mean squared error: 158.091566759293"
```

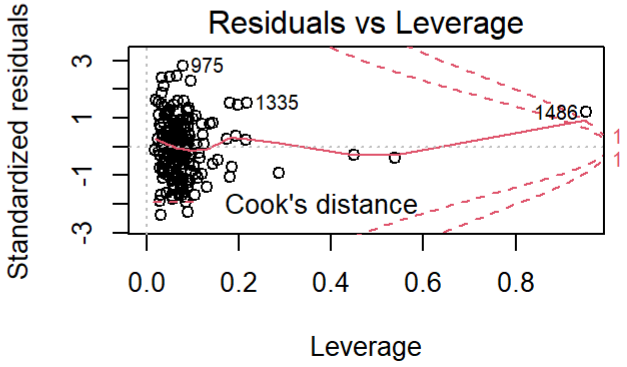
```
for (i in seq(1:length(lmlist))) {
  ModelAnalysis(lmlist[[i]])
}
```

```
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

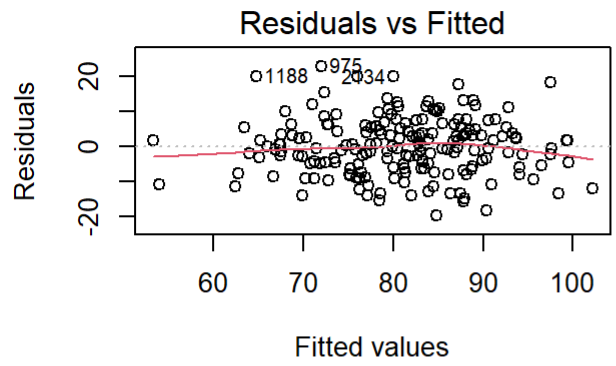


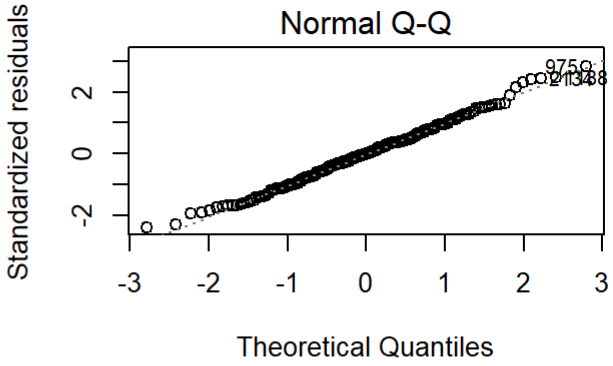


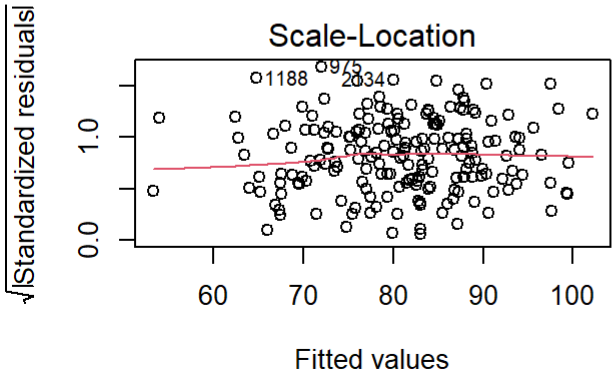


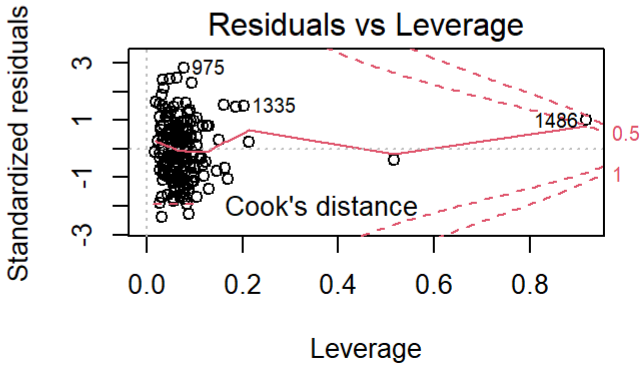



```
## [1] "Shapiro test for normality: The p-value of 0.68635315935079 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.910493611472761 is > 0.05 and
the test statistic of 8.31591414524163 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.51155503324235"
## [1] "Mean squared error: 65.6852879651226"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

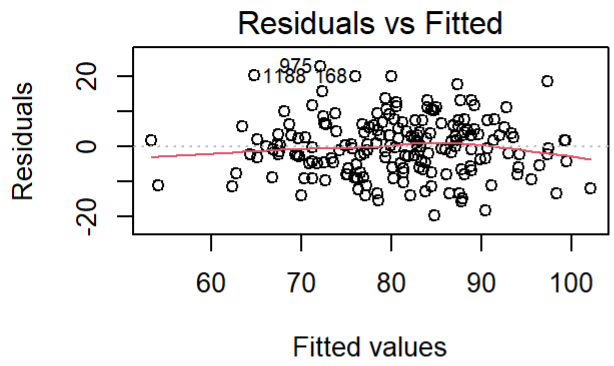


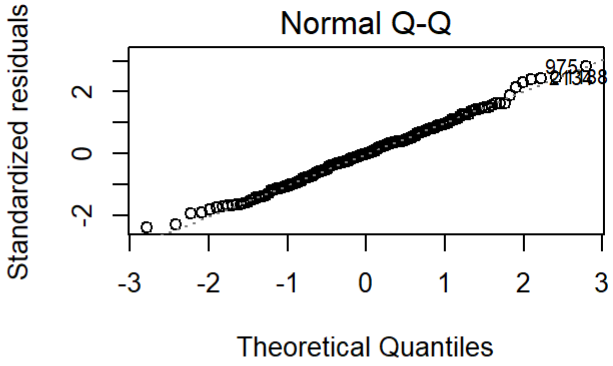


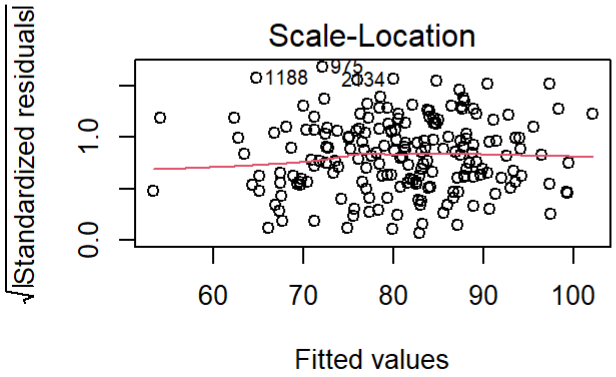


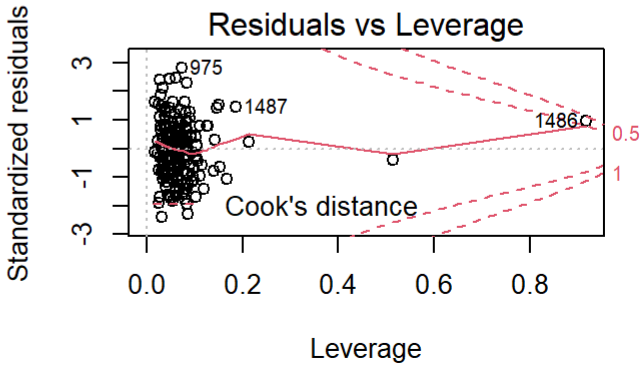



```
## [1] "Shapiro test for normality: The p-value of 0.68069919601348 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.873992010509635 is > 0.05 and
the test statistic of 8.28435224275875 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.514282234876868"
## [1] "Mean squared error: 65.6917870970535"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

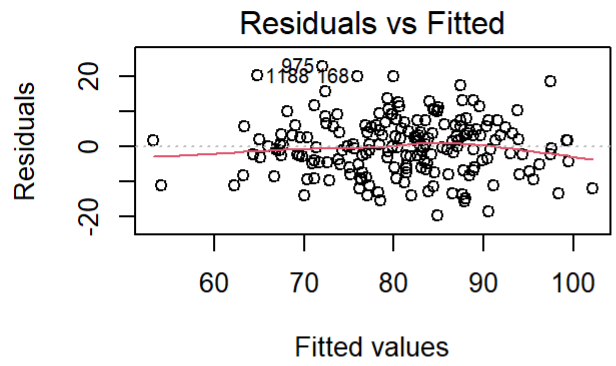


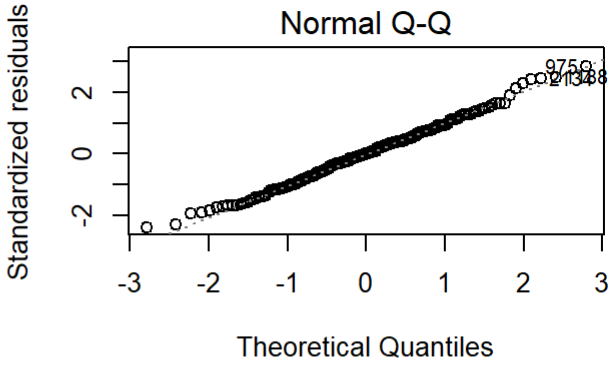


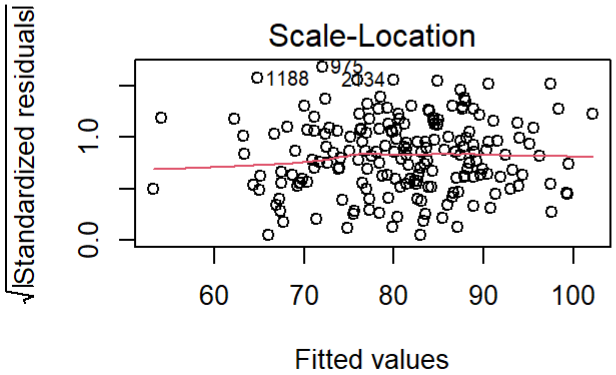


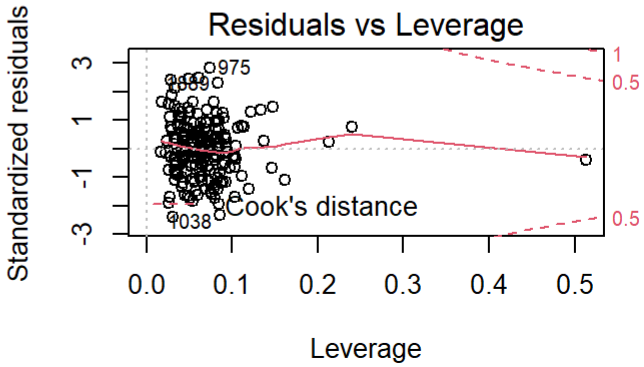



```
## [1] "Shapiro test for normality: The p-value of 0.662651426694407 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.831798626885522 is > 0.05 and
the test statistic of 8.17822169758864 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.516965248982435"
## [1] "Mean squared error: 65.7001049832636"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

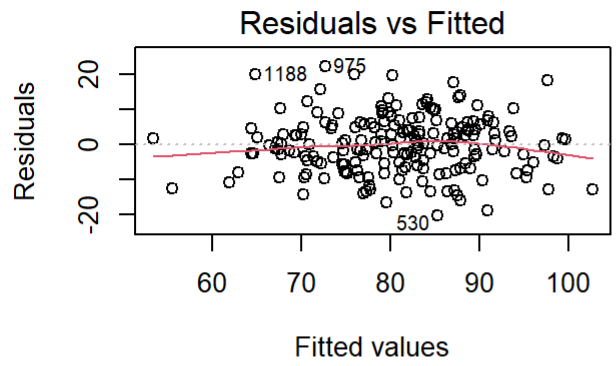


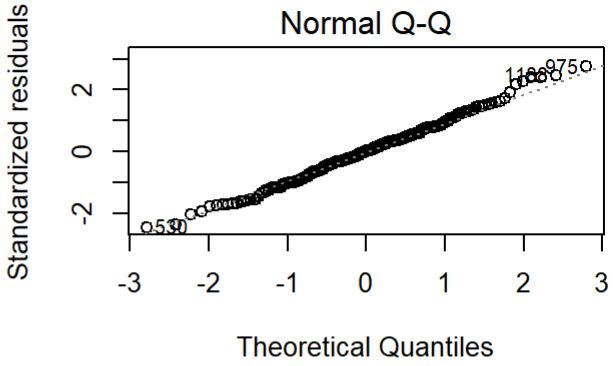


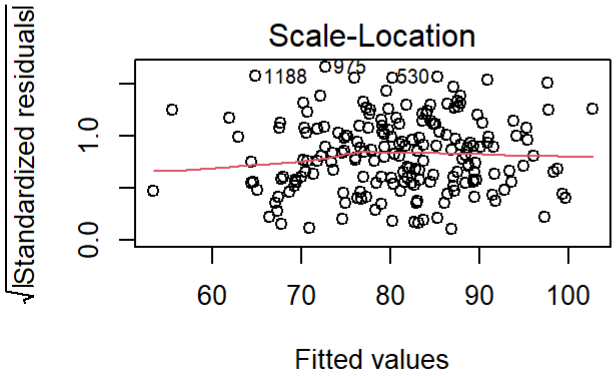


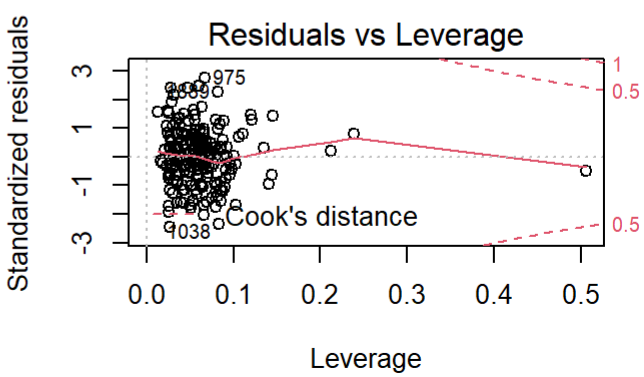



```
## [1] "Shapiro test for normality: The p-value of 0.66417583364204 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.804662852401702 is > 0.05 and
the test statistic of 7.74584732591302 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.519064259515682"
## [1] "Mean squared error: 65.7841815398255"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

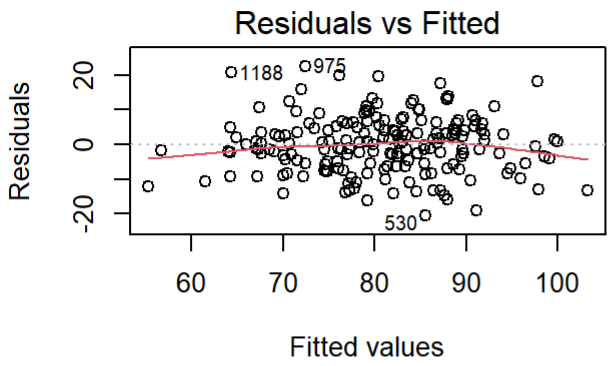


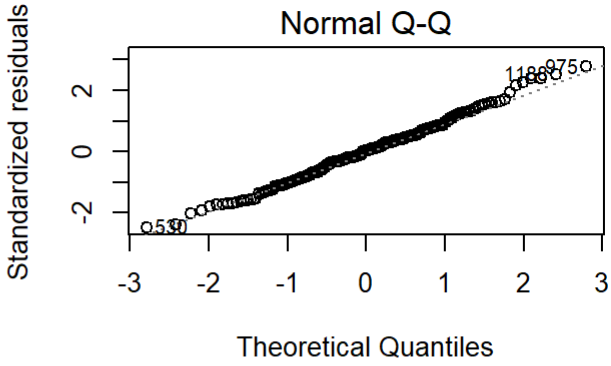


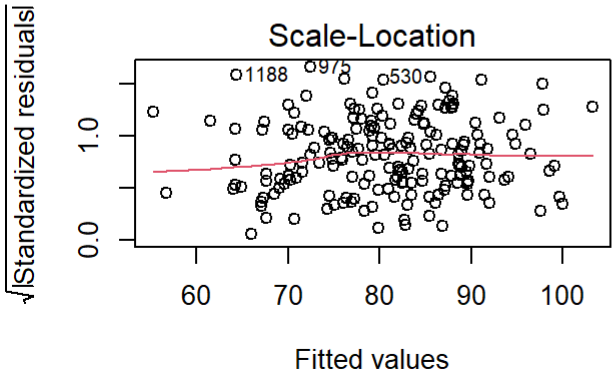


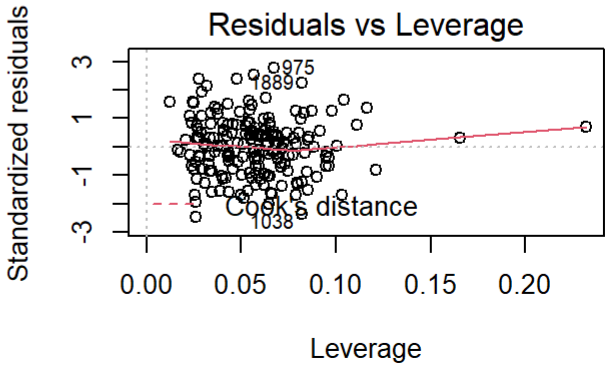



```
## [1] "Shapiro test for normality: The p-value of 0.734611363207077 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.768163982781637 is > 0.05 and
the test statistic of 7.37251750461826 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.51963939406369"
## [1] "Mean squared error: 66.074644587272"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

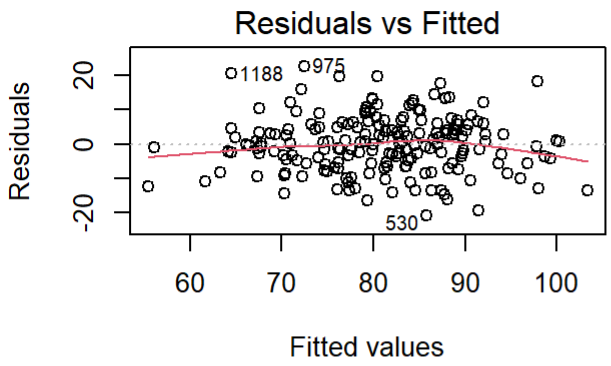


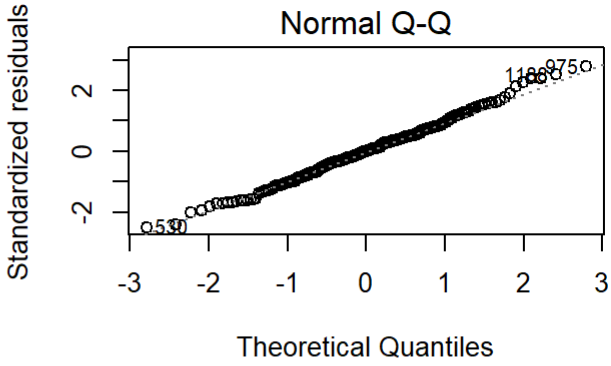


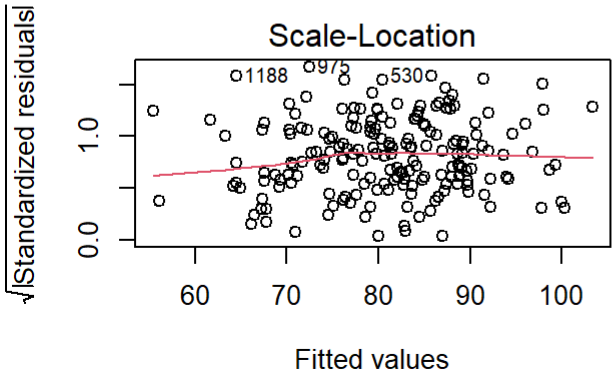


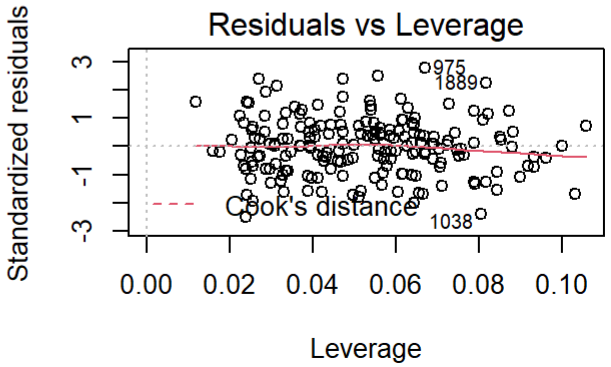



```
## [1] "Shapiro test for normality: The p-value of 0.709526095384809 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.749341648287353 is > 0.05 and
the test statistic of 6.7443247263116 is < 10, so don't reject the null; i.e., the residuals are
homoschedastic."
## [1] "Adjusted R-squared value: 0.519110799458231"
## [1] "Mean squared error: 66.5168922133216"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

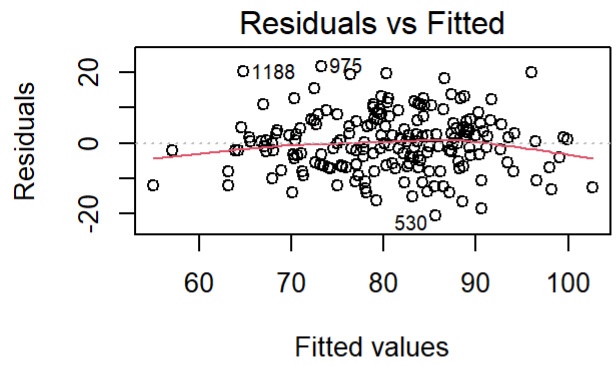


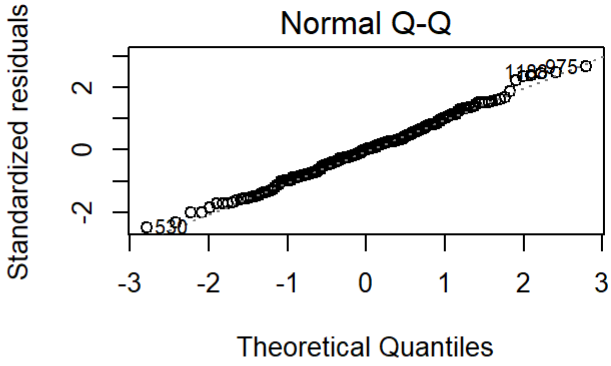


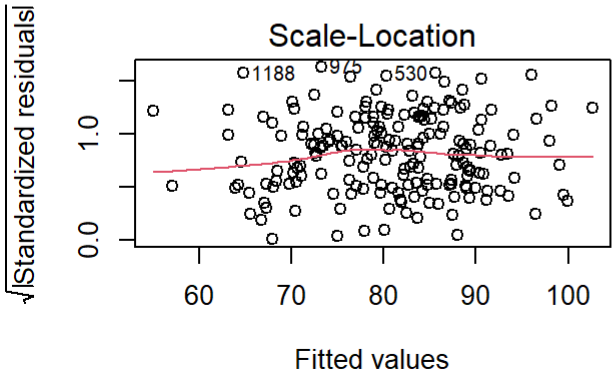


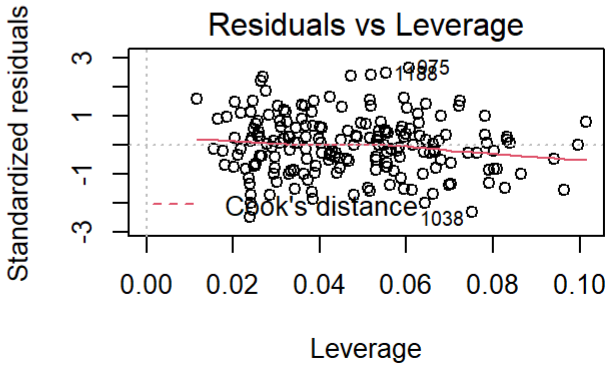



```
## [1] "Shapiro test for normality: The p-value of 0.805047520602417 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.65276403028975 is > 0.05 and
the test statistic of 6.84970313136276 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.520530922541804"
## [1] "Mean squared error: 66.68890690788"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

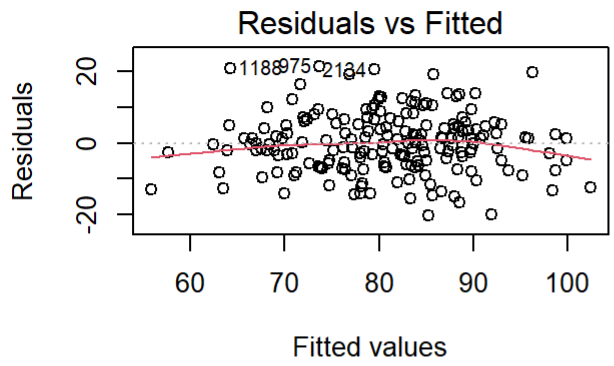


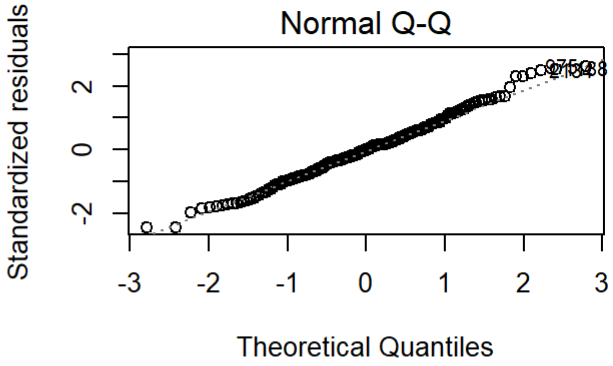


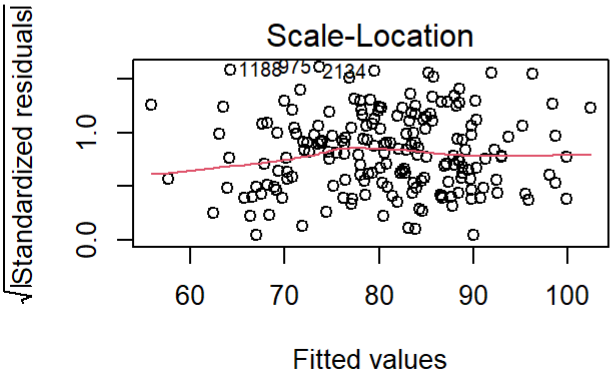


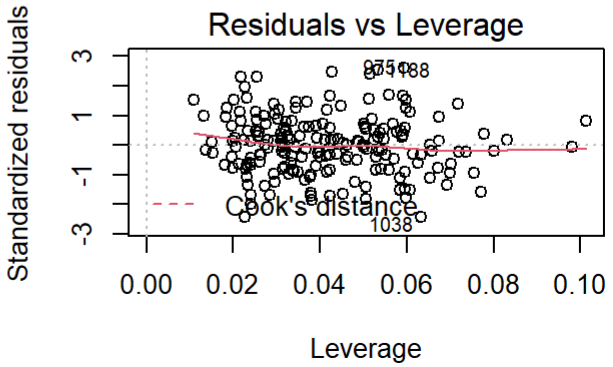



```
## [1] "Shapiro test for normality: The p-value of 0.570692045035763 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.699118509944636 is > 0.05 and
the test statistic of 5.53536741466518 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.5190476857415"
## [1] "Mean squared error: 67.2647957210824"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```

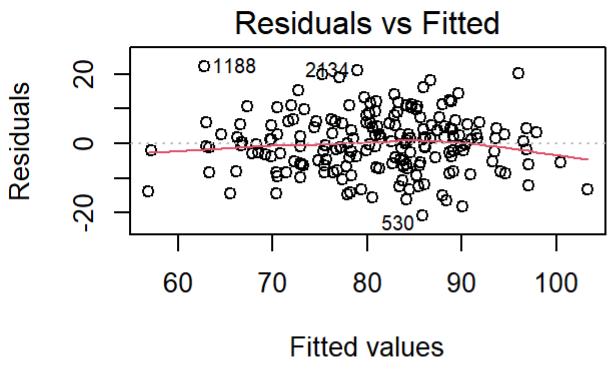


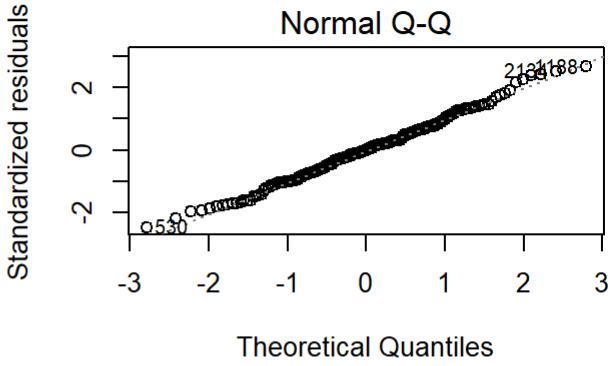


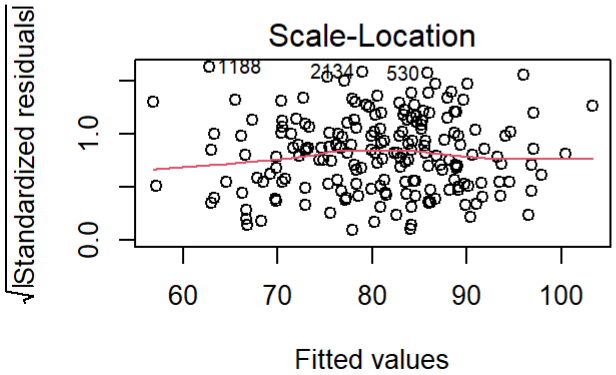


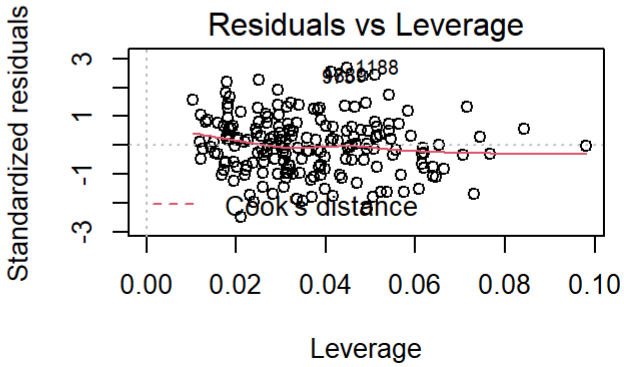



```
## [1] "Shapiro test for normality: The p-value of 0.463312391913536 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.692585827536369 is > 0.05 and
the test statistic of 4.73236465454997 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.516706529055344"
## [1] "Mean squared error: 67.9636099250672"
## [1] "-------------------------------------------------"
## lm(formula = model_params, data = train_df, na.action = na.exclude)
```










```
## [1] "Shapiro test for normality: The p-value of 0.62656351890728 is > 0.05, so do not reject
the null; i.e., the residuals are normal."
## [1] "Breusch-Pagan test for homoschedasticity: The p-value of 0.791416870165656 is > 0.05 and
the test statistic of 3.13738284092506 is < 10, so don't reject the null; i.e., the residuals ar
e homoschedastic."
## [1] "Adjusted R-squared value: 0.510864680511561"
## [1] "Mean squared error: 69.1610005069486"
```

```
# ANOVA
anova(m1, m6)
```

```
## Analysis of Variance Table
##
## Model 1: TARGET_WINS ~ INDEX + TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
##     TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB +
##     TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_PITCHING_SO +
##     TEAM_FIELDING_E
## Model 2: TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + I(TEAM_BATTING_2B^2) +
##     I(TEAM_BATTING_2B^3) + I(TEAM_BATTING_3B^2) + I(TEAM_BATTING_3B^3) +
##     TEAM_BATTING_BB + I(TEAM_BATTING_BB^2) + I(TEAM_BATTING_BB^3) +
##     TEAM_BATTING_SO + I(TEAM_BATTING_SO^2) + TEAM_BASERUN_SB +
##     I(TEAM_BASERUN_SB^2) + TEAM_PITCHING_H + I(TEAM_PITCHING_H^2) +
##     I(TEAM_PITCHING_H^3) + TEAM_PITCHING_HR + I(TEAM_PITCHING_HR^2) +
##     I(TEAM_PITCHING_HR^3) + I(TEAM_PITCHING_BB^2) + I(TEAM_PITCHING_BB^3) +
##     TEAM_PITCHING_SO + I(TEAM_PITCHING_SO^2) + I(TEAM_PITCHING_SO^3) +
##     TEAM_FIELDING_E + I(TEAM_FIELDING_E^2)
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1   2262 404638
## 2   2249 359816 13     44822 21.55 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Selected model parameters

```
# Selected model param
summary(m6)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     I(TEAM_BATTING_2B^2) + I(TEAM_BATTING_2B^3) + I(TEAM_BATTING_3B^2) +
##     I(TEAM_BATTING_3B^3) + TEAM_BATTING_BB + I(TEAM_BATTING_BB^2) +
##     I(TEAM_BATTING_BB^3) + TEAM_BATTING_SO + I(TEAM_BATTING_SO^2) +
##     TEAM_BASERUN_SB + I(TEAM_BASERUN_SB^2) + TEAM_PITCHING_H +
##     I(TEAM_PITCHING_H^2) + I(TEAM_PITCHING_H^3) + TEAM_PITCHING_HR +
##     I(TEAM_PITCHING_HR^2) + I(TEAM_PITCHING_HR^3) + I(TEAM_PITCHING_BB^2) +
##     I(TEAM_PITCHING_BB^3) + TEAM_PITCHING_SO + I(TEAM_PITCHING_SO^2) +
##     I(TEAM_PITCHING_SO^3) + TEAM_FIELDING_E + I(TEAM_FIELDING_E^2),
##     data = baseball_df_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.348  -7.814   0.316   7.825  50.193
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            6.692e+00  2.375e+01   0.282  0.77810
## TEAM_BATTING_H         5.207e-02  4.005e-03  13.000  < 2e-16 ***
## TEAM_BATTING_2B        1.029e+00  1.915e-01   5.374 8.48e-08 ***
## I(TEAM_BATTING_2B^2)  -4.071e-03  7.608e-04  -5.351 9.62e-08 ***
## I(TEAM_BATTING_2B^3)   5.147e-06  9.925e-07   5.186 2.34e-07 ***
## I(TEAM_BATTING_3B^2)   2.650e-03  3.262e-04   8.125 7.29e-16 ***
## I(TEAM_BATTING_3B^3)  -1.490e-05  1.947e-06  -7.650 2.95e-14 ***
## TEAM_BATTING_BB        2.661e-01  4.625e-02   5.753 9.98e-09 ***
## I(TEAM_BATTING_BB^2)  -4.716e-04  1.019e-04  -4.629 3.88e-06 ***
## I(TEAM_BATTING_BB^3)   3.037e-07  7.190e-08   4.224 2.49e-05 ***
## TEAM_BATTING_SO        5.674e-02  1.287e-02   4.409 1.09e-05 ***
## I(TEAM_BATTING_SO^2)  -3.999e-05  7.365e-06  -5.430 6.23e-08 ***
## TEAM_BASERUN_SB        7.643e-02  9.135e-03   8.366  < 2e-16 ***
## I(TEAM_BASERUN_SB^2)  -7.267e-05  1.670e-05  -4.352 1.41e-05 ***
## TEAM_PITCHING_H       -1.415e-01  2.536e-02  -5.578 2.72e-08 ***
## I(TEAM_PITCHING_H^2)   5.448e-05  1.086e-05   5.015 5.72e-07 ***
## I(TEAM_PITCHING_H^3)  -6.375e-09  1.464e-09  -4.354 1.40e-05 ***
## TEAM_PITCHING_HR      -1.925e-01  4.124e-02  -4.668 3.22e-06 ***
## I(TEAM_PITCHING_HR^2)  1.983e-03  3.229e-04   6.142 9.62e-10 ***
## I(TEAM_PITCHING_HR^3) -4.317e-06  7.763e-07  -5.561 2.99e-08 ***
## I(TEAM_PITCHING_BB^2) -2.853e-04  8.912e-05  -3.201  0.00139 **
## I(TEAM_PITCHING_BB^3)  3.735e-07  1.286e-07   2.904  0.00372 **
## TEAM_PITCHING_SO      -4.079e-02  1.759e-02  -2.318  0.02052 *
## I(TEAM_PITCHING_SO^2)  3.352e-05  2.231e-05   1.502  0.13312
## I(TEAM_PITCHING_SO^3) -1.094e-08  8.914e-09  -1.228  0.21970
## TEAM_FIELDING_E       -1.200e-01  1.547e-02  -7.757 1.31e-14 ***
## I(TEAM_FIELDING_E^2)   1.317e-04  2.058e-05   6.401 1.88e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.65 on 2249 degrees of freedom
## Multiple R-squared:  0.3626, Adjusted R-squared:  0.3552
## F-statistic: 49.21 on 26 and 2249 DF,  p-value: < 2.2e-16
```

Predictions on evaluation data

First, impute missing values.

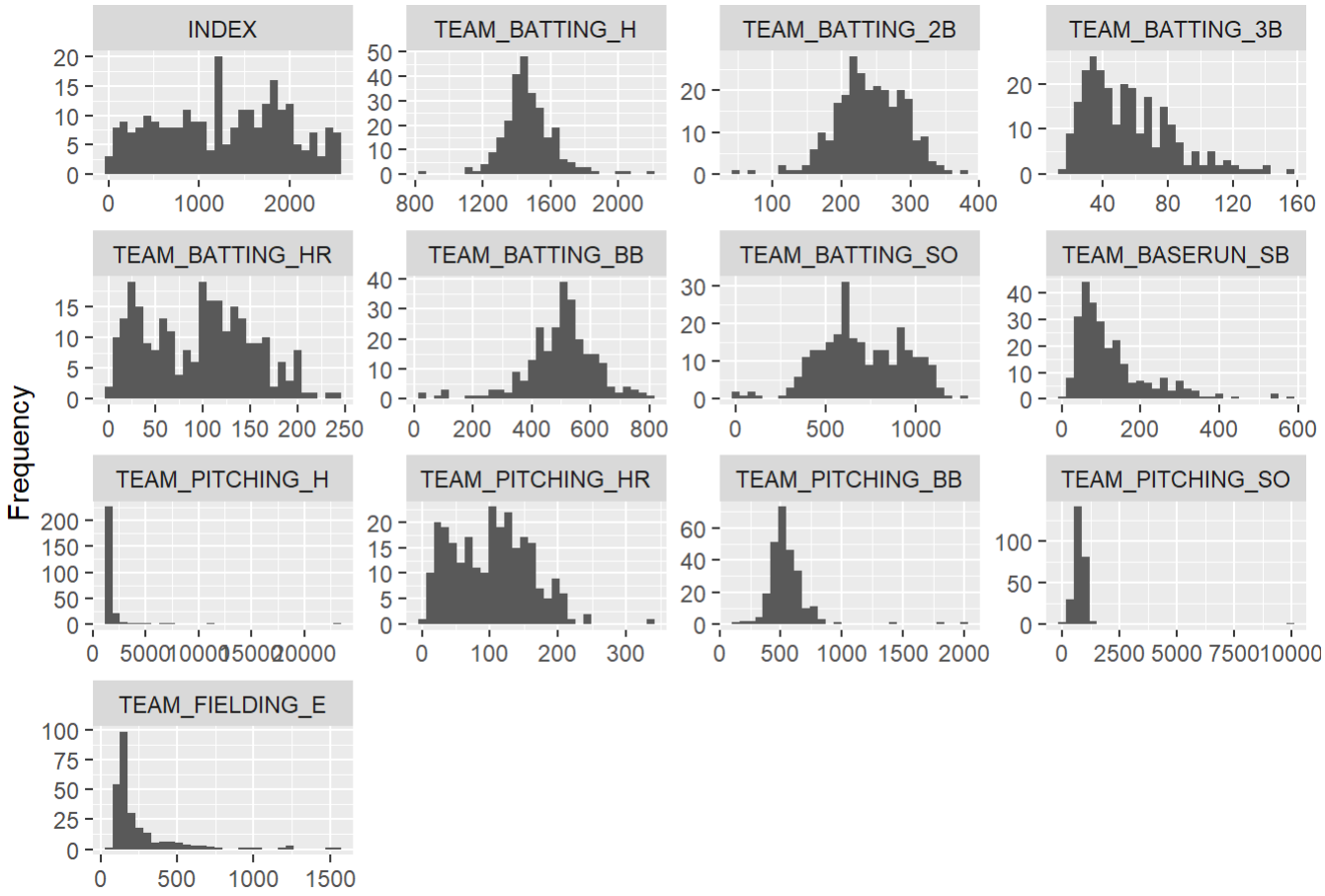
```
# New DF with Missing Removed for eval data
eval_df_mv <- eval_df[, !names(eval_df) %in% c('TEAM_BATTING_HBP','TEAM_BASERUN_CS','TEAM_FIELDI
NG_DP')]

# Impute NAs with Median
eval_df_imputed <- mice(eval_df_mv, m=5, maxit = 5, method = 'pmm')
```

```
##
##  iter imp variable
##   1   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   2   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   3   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   4   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   5   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
```

```
eval_df_final <- complete(eval_df_imputed)

# Plot
ggplot(melt(eval_df_final), aes(x=value)) + geom_histogram() + facet_wrap(~variable, scale='fre
e') + labs(x='', y='Frequency')
```

```
## Using  as id variables
```

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




```
# Replace Error Maxs
eval_df_final$TEAM_PITCHING_H[eval_df_final$TEAM_PITCHING_H > 3*sd(eval_df_final$TEAM_PITCHING_
H)] <- median(eval_df_final$TEAM_PITCHING_H)
eval_df_final$TEAM_PITCHING_BB[eval_df_final$TEAM_PITCHING_BB > 3*sd(eval_df_final$TEAM_PITCHING
_BB)] <- median(eval_df_final$TEAM_PITCHING_BB)
eval_df_final$TEAM_PITCHING_SO[eval_df_final$TEAM_PITCHING_SO > 3*sd(eval_df_final$TEAM_PITCHING
_SO)] <- median(eval_df_final$TEAM_PITCHING_SO)
eval_df_final$TEAM_FIELDING_E[eval_df_final$TEAM_FIELDING_E > 3*sd(eval_df_final$TEAM_FIELDING_
E)] <- median(eval_df_final$TEAM_FIELDING_E)

# Summary
summary(eval_df_final)
```

```
##      INDEX      TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
##  Min.   :   9   Min.   : 819   Min.   : 44.0   Min.   : 14.00
##  1st Qu.: 708   1st Qu.:1387   1st Qu.:210.0   1st Qu.: 35.00
##  Median :1249   Median :1455   Median :239.0   Median : 52.00
##  Mean   :1264   Mean   :1469   Mean   :241.3   Mean   : 55.91
##  3rd Qu.:1832   3rd Qu.:1548   3rd Qu.:278.5   3rd Qu.: 72.00
##  Max.   :2525   Max.   :2170   Max.   :376.0   Max.   :155.00
##  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB
##  Min.   :  0.00   Min.   : 15.0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 44.50   1st Qu.:436.5   1st Qu.: 532.5   1st Qu.: 60.5
##  Median :101.00   Median :509.0   Median : 677.0   Median : 95.0
##  Mean   : 95.63   Mean   :499.0   Mean   : 699.8   Mean   :127.9
##  3rd Qu.:135.50   3rd Qu.:565.5   3rd Qu.: 904.5   3rd Qu.:155.5
##  Max.   :242.00   Max.   :792.0   Max.   :1268.0   Max.   :580.0
##  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
##  Min.   :1155    Min.   :  0.0    Min.   :136      Min.   :   0.0
##  1st Qu.:1426    1st Qu.: 52.0    1st Qu.:471      1st Qu.: 608.5
##  Median :1515    Median :104.0    Median :526      Median : 731.0
##  Mean   :1602    Mean   :102.1    Mean   :490      Mean   : 754.4
##  3rd Qu.:1654    3rd Qu.:142.5    3rd Qu.:526      3rd Qu.: 927.5
##  Max.   :4120    Max.   :336.0    Max.   :526      Max.   :1462.0
##  TEAM_FIELDING_E
##  Min.   : 73.0
##  1st Qu.:131.0
##  Median :163.0
##  Mean   :206.5
##  3rd Qu.:225.0
##  Max.   :680.0
```

```
# Predict wins
eval_df_final$TARGET_WINS <- round(predict(m6, eval_df_final), 0)
eval_pred <-
  eval_df_final %>% select(TARGET_WINS, everything()) %>%
  arrange(INDEX)
eval_pred
```

```
##     TARGET_WINS INDEX TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
## 1            61     9           1209             170              33
## 2            64    10           1221             151              29
## 3            72    14           1395             183              29
## 4            86    47           1539             309              29
## 5            67    60           1445             203              68
## 6            57    63           1431             236              53
## 7            80    74           1430             219              55
## 8            62    83           1385             158              42
## 9            75    98           1259             177              78
## 10           69   120           1397             212              42
## 11           70   123           1427             243              40
## 12           85   135           1496             239              55
## 13           86   138           1420             223              57
## 14           83   140           1460             232              22
## 15           81   151           1411             195              22
## 16           81   153           1434             192              30
## 17           70   171           1297             204              22
## 18           79   184           1446             284              25
## 19           66   193           1276             162              52
## 20           87   213           1715             322              72
## 21           82   217           1520             295              68
## 22           81   226           1597             291              38
## 23           80   230           1453             256              67
## 24           71   241           1378             225              26
## 25           83   291           1516             277              24
## 26           88   294           1556             288              20
## 27           62   300           1499             183              28
## 28           71   348           1464             263              58
## 29           81   350           1558             318              66
## 30           67   357           1502             308              36
## 31           91   367           1596             320              58
## 32           85   368           1546             260              59
## 33           87   372           1516             282              53
## 34           90   382           1550             275              47
## 35           80   388           1447             260              54
## 36           85   396           1450             252              28
## 37           75   398           1347             239              36
## 38           93   403           1561             260              56
## 39           86   407           1578             252              26
## 40           87   410           1598             259              45
## 41           81   412           1497             322              21
## 42           89   414           1569             310              39
## 43           30   436           1119             118              33
## 44          103   440           1609             196             120
## 45           84   476           1514             175              70
## 46           92   479           1657             237             119
## 47           96   481           1746             213             106
## 48           73   501           1319             224              70
## 49           73   503           1293             204              70
## 50           76   506           1420             235              70
## 51           74   519           1496             269              54
```

```
## 52          80   522          1625          289           38
## 53          74   550          1391          239           50
## 54          77   554          1319          203           43
## 55          72   566          1411          251           35
## 56          80   578          1420          221           41
## 57          87   596          1552          206          106
## 58          80   599          1280          203           72
## 59          61   605          1120          122           61
## 60          80   607          1390          183           84
## 61          85   614          1554          252           81
## 62          88   644          1410          218           69
## 63          86   692          1507          262           28
## 64          85   699          1481          284           19
## 65          89   700          1450          253           23
## 66          90   716          1637          260           93
## 67          69   721          1436          202           82
## 68          79   722          1600          218           89
## 69          79   729          1348          168           76
## 70          91   731          1460          191          111
## 71          85   746          1621          255          126
## 72          76   763          1433          241           49
## 73          85   774          1440          232           48
## 74          90   776          1479          211           34
## 75          76   788          1573          281           36
## 76          81   789          1558          224           42
## 77          86   792          1385          225           46
## 78          81   811          1419          250           27
## 79          74   835          1284          198           61
## 80          82   837          1403          200           68
## 81          82   861          1631          358           48
## 82          86   862          1666          343           82
## 83         100   863          1804          376           86
## 84          76   871          1534          284           53
## 85          86   879          1472          222           52
## 86          80   887          1489          229           21
## 87          83   892          1367          198           21
## 88          84   904          1485          222           46
## 89          90   909          1458          225           32
## 90          90   925          1530          334           30
## 91          72   940          1421          160           72
## 92          99   951          1869          301          122
## 93          64   976          1400          169           66
## 94          83   981          1494          193           81
## 95          82   983          1449          223           62
## 96          81   984          1385          200           76
## 97          95   989          1443          218           99
## 98          95   995          1825          284          106
## 99          88  1000          1627          296           95
## 100         90  1001          1623          299          106
## 101         80  1007          1556          298           82
## 102         69  1016          1381          228           39
## 103         81  1027          1556          272           46
```

```
## 104          84  1033          1416          206           32
## 105          71  1070          1413          257           21
## 106          70  1081          1504          253          102
## 107          56  1084          1193          165           68
## 108          75  1098          1461          325           30
## 109          86  1150          1458          294           36
## 110          55  1160          1295          237           64
## 111          88  1169          1431          263           58
## 112          87  1172          1469          305           59
## 113          95  1174          1633          266           59
## 114          91  1176          1603          295           58
## 115          84  1178          1487          269           52
## 116          79  1184          1474          318           44
## 117          90  1193          1594          296           52
## 118          82  1196          1415          285           42
## 119          72  1199          1445          289           34
## 120          78  1207          1362          199           81
## 121          88  1218          1572          195          106
## 122          68  1223          1209          168           56
## 123          64  1226          1242          155           69
## 124          55  1227          1098          116           63
## 125          73  1229          1235          175           77
## 126          83  1241          1651          247           80
## 127          83  1244          1712          265           85
## 128          76  1246          1391          206           78
## 129          84  1248          1625          299           73
## 130          91  1249          1740          319           77
## 131          80  1253          1626          303           55
## 132          78  1261          1471          277           36
## 133          74  1305          1373          232           14
## 134          76  1314          1466          215           35
## 135          89  1323          1450          226           30
## 136          70  1328          1474          223           57
## 137          79  1353          1335          228           49
## 138          77  1363          1455          233           36
## 139          90  1371          1477          272           35
## 140          81  1372          1426          240           25
## 141          69  1389          1255          183           61
## 142          71  1393          1264          141           79
## 143          91  1421          1695          310           89
## 144          72  1431          1460          274           66
## 145          75  1437          1349          237           46
## 146          73  1442          1340          226           40
## 147          77  1450          1396          257           42
## 148          83  1463          1472          259           47
## 149          84  1464          1544          256           46
## 150          82  1470          1453          282           41
## 151          85  1471          1446          257           39
## 152          78  1484          1468          289           30
## 153         -20  1495          1546           44           29
## 154          69  1507          1372          195           31
## 155          76  1514          1365          203           29
```

```
## 156          70  1526          1314          172           26
## 157          88  1549          1469          323           41
## 158          70  1552          1382          185           86
## 159          86  1556          1642          218          135
## 160          68  1564          1324          153           65
## 161         105  1585          1770          313          116
## 162         116  1586          1765          293           83
## 163          90  1590          1590          277           76
## 164         112  1591          1775          334           88
## 165         102  1592          1635          297           77
## 166          94  1603          1557          264           79
## 167          89  1612          1485          210           57
## 168          87  1634          1461          229           41
## 169          76  1645          1322          208           19
## 170          82  1647          1462          281           18
## 171          90  1673          1537          217          115
## 172          89  1674          1495          236           85
## 173          76  1687          1468          280           70
## 174          86  1688          1689          296           74
## 175          76  1700          1533          301           59
## 176          77  1708          1379          229           55
## 177          81  1713          1373          223           37
## 178          73  1717          1394          215           43
## 179          73  1721          1371          223           36
## 180          82  1730          1400          210           28
## 181          89  1737          1327          209           33
## 182          85  1748          1432          263           33
## 183          87  1749          1474          251           22
## 184          85  1763          1450          279           28
## 185         106  1768          2025          292          140
## 186          93  1778          1669          281          102
## 187          88  1780          1631          291           79
## 188          75  1782          1420          299           79
## 189          66  1784          1312          230           52
## 190         112  1794          2058          336           90
## 191          68  1803          1351          181           58
## 192          81  1804          1452          199           87
## 193          74  1819          1466          242           57
## 194          72  1832          1534          256           44
## 195          70  1833          1609          311           38
## 196          63  1844          1344          207           28
## 197          71  1847          1438          239           41
## 198          87  1854          1368          225           53
## 199          81  1855          1381          218           52
## 200          83  1857          1498          250           59
## 201          72  1864          1389          206           53
## 202          79  1865          1448          224           49
## 203          73  1869          1307          225           58
## 204          88  1880          1517          250           38
## 205          81  1881          1417          245           25
## 206          84  1882          1352          209           45
## 207          73  1894          1458          296           34
```

```
## 208          74  1896          1390          290           35
## 209          76  1916          1475          257           80
## 210          74  1918          1378          178           85
## 211          93  1921          1817          277          155
## 212          88  1926          1711          213          133
## 213          84  1938          1415          217          112
## 214          65  1979          1263          190           32
## 215          72  1982          1328          221           63
## 216          84  1987          1571          248           59
## 217          86  1997          1522          235           70
## 218          93  2004          1550          278           57
## 219          77  2011          1412          237           33
## 220          79  2015          1344          243           46
## 221          71  2022          1441          276           30
## 222          74  2025          1395          271           35
## 223          80  2027          1506          320           31
## 224          76  2031          1437          269           39
## 225          86  2036          2170          241           70
## 226          79  2066          1324          194           53
## 227          79  2073          1442          239           25
## 228          75  2087          1413          279           37
## 229          81  2092          1416          269           39
## 230          80  2125          1523          216           97
## 231          74  2148          1294          169           51
## 232          92  2162          1668          251           98
## 233          86  2191          1422          215           53
## 234          90  2203          1524          231           31
## 235          79  2218          1392          227           41
## 236          74  2221          1318          200           44
## 237          77  2225          1499          229           26
## 238          81  2232          1345          215           48
## 239          85  2267          1620          210          139
## 240          67  2291          1339          185           80
## 241          85  2299          1621          272           86
## 242          86  2317          1585          288           62
## 243          80  2318          1576          269           46
## 244          84  2353          1541          300           49
## 245          62  2403          1149          175           18
## 246          85  2411          1626          265           27
## 247          79  2415          1461          228           29
## 248          84  2424          1472          284           39
## 249          75  2441          1366          218           39
## 250          87  2464          1489          287           36
## 251          78  2465          1457          305           38
## 252          66  2472          1454          220           52
## 253          92  2481          1642          221           98
## 254           6  2487           819           72           72
## 255          66  2500          1251          162           23
## 256          80  2501          1345          190           23
## 257          79  2520          1381          263           37
## 258          84  2521          1410          270           36
## 259          75  2525          1423          339           34
```

```
##     TEAM_BATTING_HR TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB
## 1                83             447            1080               62
## 2                88             516             929               54
## 3                93             509             816               59
## 4               159             486             914              148
## 5                 5              95             416              246
## 6                10             215             377              129
## 7                37             568             527              365
## 8                33             356             609              185
## 9                23             466             689              150
## 10               58             452             584               52
## 11               50             495             640               64
## 12              164             462             670               48
## 13              186             511             751               31
## 14              176             503             680               27
## 15              141             485             665               59
## 16              153             434             747               57
## 17              130             491            1008               84
## 18              166             565            1041               77
## 19               17             383             592              138
## 20              116             527             397               90
## 21               49             628             459               77
## 22               98             629             563               54
## 23              105             653             651               40
## 24              118             533             677               18
## 25              152             431             902               89
## 26              164             474             878              121
## 27                3              83               0              307
## 28               47             385             479               63
## 29               32             634             439               83
## 30               39             432             602               45
## 31              130             718             596               70
## 32              110             630             541               72
## 33              115             723             695               47
## 34              146             765             723               29
## 35              148             532             935               39
## 36              203             594             855               50
## 37              130             546             897               69
## 38              214             531             911               66
## 39              135             567             780               48
## 40              181             500             842               38
## 41              145             599             711               41
## 42              124             623             728               65
## 43                7              37               0              129
## 44               62             781             599              536
## 45               80             615             612              392
## 46               41             593             334              325
## 47               69             526             429              324
## 48               56             416             677              176
## 49               18             437             630              134
## 50               36             450             443              121
## 51               76             412             500               55
```

```
## 52   80  517  486  72
## 53  145  499 1041  70
## 54  130  415  854  41
## 55  107  471  912  93
## 56  104  417  816  77
## 57   38  566  401 334
## 58   15  392  616 227
## 59    7  427  569 194
## 60   18  445  609 216
## 61   29  494  414 174
## 62   45  738  627  65
## 63  159  573  907 107
## 64  242  499 1030  78
## 65  200  435 1002 137
## 66   26  487  288 446
## 67   44  376  681 160
## 68   21  344  538 152
## 69   23  506  689 296
## 70   22  612  629 306
## 71   37  478  350  54
## 72   45  468  501  52
## 73  155  586  679  49
## 74  232  555  799  47
## 75  106  379  938  59
## 76  171  474 1042  79
## 77  130  637  961 147
## 78  164  488 1006 124
## 79   19  383  634 186
## 80   10  390  579 201
## 81  105  553  455  55
## 82   98  487  600  67
## 83  129  541  494  69
## 84   74  539  624  50
## 85  156  659  788  48
## 86  134  467  603  61
## 87  156  506  857 109
## 88  101  534  692  88
## 89  109  651  625 151
## 90  198  630 1061 143
## 91   30  523  508 289
## 92   58  347  127 399
## 93   26  431  344 156
## 94   12  340  397 207
## 95   20  423  455 298
## 96   29  483  523 262
## 97   24  716  554 254
## 98   61  616  398 101
## 99   38  630  445  93
## 100  54  622  445 149
## 101  60  500  550  72
## 102  80  535  501  41
## 103 114  532  634  32
```

```
## 104  168  610  775  36
## 105  204  546 1268  87
## 106   33  262  482  92
## 107   45  299 1011 307
## 108  166  470 1145  89
## 109  187  590  999  89
## 110   25  360  814 129
## 111  118  591  675 155
## 112   98  498  644 216
## 113  115  508  709 185
## 114  132  442  758 133
## 115  117  400  832 106
## 116  101  501  884 108
## 117  152  538  938 128
## 118  140  524  921 140
## 119  126  424 1008  53
## 120   29  408  508 386
## 121   30  522  288 297
## 122   16  435  634 217
## 123   20  368  569 132
## 124   29  340  845 119
## 125   26  457  743 159
## 126   59  357  335  83
## 127   68  463  406  39
## 128   41  390  523 112
## 129  105  534  481  85
## 130  128  506  569  56
## 131   84  584  592  59
## 132   65  602  509  85
## 133  130  478  966 155
## 134  158  527 1151 143
## 135  203  536 1092 102
## 136   18  259  391  92
## 137  120  500  909 106
## 138   97  435  677  52
## 139   82  511  779 256
## 140  125  555  932 138
## 141   11  304  814 161
## 142    9  392  630 181
## 143   66  610  421 110
## 144   63  538  674  54
## 145   53  610  639  50
## 146  117  554  771  14
## 147  150  554  969  92
## 148   82  604  684  99
## 149  112  526  693  66
## 150  141  502  779  68
## 151  196  501  977  81
## 152  106  506  990 119
## 153    0   15   44   0
## 154  103  353  932  36
## 155   98  547  958  89
```

```
## 156  112  436 1031  141
## 157  200  547 1071  146
## 158   32  326  642  259
## 159   29  449  459  252
## 160   17  437  527  201
## 161  160  677  599   96
## 162  164  792  587  146
## 163  113  657  510   74
## 164  193  741  629   82
## 165  183  746  639   63
## 166  146  655  503   25
## 167  153  591  746   52
## 168  152  515  597   66
## 169  147  427 1027  119
## 170  163  536  903   78
## 171   23  517  445  275
## 172   35  565  579  234
## 173   66  565  488   60
## 174   59  580  343  103
## 175  104  536  567   64
## 176   64  636  592   39
## 177   94  718  590   55
## 178  118  505  765   42
## 179  116  540  783   17
## 180  148  617  953  100
## 181  114  596  823  343
## 182  199  593 1056  140
## 183  156  580  926  129
## 184  205  609 1008   46
## 185   32  259   70  259
## 186   35  391  473  580
## 187   52  650  604  307
## 188    5  233  587  259
## 189   29  324  591  536
## 190   75  573  324  341
## 191   25  402  612  169
## 192   17  433  482  192
## 193   68  300  562  106
## 194   64  406  511   59
## 195   61  433  581   57
## 196   59  472  527   57
## 197   96  463  629   72
## 198  139  686  708   46
## 199  127  615  708   47
## 200  130  603  916   54
## 201  145  497 1098   46
## 202  117  510  969   56
## 203  102  522 1073   72
## 204  104  563  654  156
## 205  112  506  831  128
## 206  125  640  906  143
## 207  106  559  995   81
```

```
## 208  116  519 1032  92
## 209   52  515  573 284
## 210   35  512  604 246
## 211   60  541  259 319
## 212   29  418  375 195
## 213   52  552  613 168
## 214   97  511  762  45
## 215   96  495  686  23
## 216  126  511  786  36
## 217  130  444  871  66
## 218  133  474  878 260
## 219   98  438  841  96
## 220  111  560  959 120
## 221  141  513 1094  95
## 222  107  393 1060 159
## 223  168  564 1032  86
## 224  143  418 1073  63
## 225   13  111  102  92
## 226   94  537  775 101
## 227  136  484  917  96
## 228  157  602 1177 131
## 229  130  600  977  99
## 230   33  360  712 129
## 231   24  546  498 217
## 232   79  497  413 145
## 233  140  660  662  44
## 234  200  513  807  72
## 235  134  568  842  90
## 236   80  512  845 101
## 237  112  528  980 126
## 238  141  471  973  95
## 239   66  542  355 233
## 240   34  413  579 149
## 241   95  503  545  87
## 242  105  572  498  39
## 243   67  542  513  58
## 244  101  451  781 117
## 245   59  529  974 133
## 246  125  483  593  92
## 247  121  423  812  82
## 248  181  483  984 113
## 249   99  451  649  28
## 250  195  470 1094 156
## 251  187  522 1142  71
## 252    9   97  393  92
## 253   56  638  451 319
## 254   18  198 1107 129
## 255   95  492  860  71
## 256  125  695  777  77
## 257  102  463  976 196
## 258  122  542  860 228
## 259  172  420 1084  75
```

```
##     TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
## 1              1209               83              447             1080
## 2              1221               88              516              929
## 3              1395               93              509              816
## 4              1539              159              486              914
## 5              3902               14              257             1123
## 6              2793               20              420              736
## 7              1544               40              526              569
## 8              1626               39              418              715
## 9              1342               25              497              734
## 10             1489               62              482              622
## 11             1501               53              526              673
## 12             1574              173              486              705
## 13             1494              196              526              790
## 14             1536              185              526              715
## 15             1411              141              485              665
## 16             1434              153              434              747
## 17             1313              132              497             1021
## 18             1464              168              526             1054
## 19             1351               18              406              714
## 20             1816              123              526              420
## 21             1620               52              526              489
## 22             1702              104              526              600
## 23             1559              113              526              698
## 24             1450              124              526              712
## 25             1516              152              431              902
## 26             1556              164              474              878
## 27             1515               10              286                0
## 28             1540               49              405              504
## 29             1639               34              526              462
## 30             1601               42              460              642
## 31             1679              137              526              627
## 32             1648              117              526              577
## 33             1595              121              526              731
## 34             1631              154              526              761
## 35             1465              150              526              947
## 36             1450              203              526              855
## 37             1408              136              526              938
## 38             1571              215              526              917
## 39             2367              203              526             1170
## 40             1598              181              500              842
## 41             1506              146              526              715
## 42             1569              124              526              728
## 43             4120               26              136                0
## 44             1931               74              526              719
## 45             1803               95              526              729
## 46             2114               52              526              426
## 47             2176               86              526              535
## 48             1397               59              440              717
## 49             1360               19              460              663
## 50             1494               38              473              466
## 51             1574               80              433              526
```

```
## 52            1709            84            526            511
## 53            1391           145            499           1041
## 54            1319           130            415            854
## 55            1411           107            471            912
## 56            1420           104            417            816
## 57            1849            45            526            478
## 58            1346            16            412            648
## 59            1186             7            452            621
## 60            1462            19            468            642
## 61            1798            34            526            479
## 62            1483            47            526            660
## 63            1516           160            526            913
## 64            1481           242            499           1030
## 65            1450           200            435           1002
## 66            2088            33            526            367
## 67            1674            51            438            794
## 68            1851            24            398            623
## 69            1427            24            526            758
## 70            1546            23            526            666
## 71            1705            39            503            368
## 72            1507            47            492            527
## 73            1515           163            526            714
## 74            1556           244            526            841
## 75            1573           106            379            938
## 76            1558           171            474           1042
## 77            1457           137            526           1011
## 78            1419           164            488           1006
## 79            1351            20            403            715
## 80            1495            11            416            527
## 81            1716           110            526            479
## 82            1764           104            516            635
## 83            1898           136            526            520
## 84            1614            78            526            656
## 85            1548           164            526            829
## 86            1566           141            491            634
## 87            1367           156            506            857
## 88            1494           102            526            696
## 89            1458           109            526            625
## 90            1530           198            526           1061
## 91            1731            37            526            619
## 92            1515           336            526            735
## 93            1680            31            517            413
## 94            1614            13            367            479
## 95            1544            21            451            535
## 96            1457            31            508            588
## 97            1518            25            526            583
## 98            1932            65            526            421
## 99            1712            40            526            468
## 100           1718            57            526            471
## 101           1637            63            526            579
## 102           1453            84            526            527
## 103           1637           120            526            667
```

```
## 104          1490          177          526          815
## 105          1413          204          526         1268
## 106          2901           64          505          930
## 107          1726           65          432         1462
## 108          1461          166          470         1145
## 109          1458          187          526          999
## 110          1734           33          482         1090
## 111          1431          118          526          675
## 112          1469           98          498          644
## 113          1633          115          508          709
## 114          1603          132          442          758
## 115          1487          117          400          832
## 116          1483          102          504          889
## 117          1604          153          526          944
## 118          1415          140          526          921
## 119          1445          126          424         1008
## 120          1576           34          472          588
## 121          1721           33          526          315
## 122          1280           17          461          731
## 123          1359           22          403          860
## 124          1155           31          358          929
## 125          1299           27          481          782
## 126          1737           62          376          352
## 127          1813           72          490          430
## 128          1473           43          413          554
## 129          1721          111          526          509
## 130          1830          135          526          599
## 131          1733           90          526          631
## 132          1547           68          526          535
## 133          1373          130          478          966
## 134          1649          178          526         1295
## 135          1450          203          526         1092
## 136          2985           36          526          792
## 137          1335          120          500          909
## 138          1464           98          438          681
## 139          1477           82          511          779
## 140          1426          125          526          932
## 141          1346           12          326          873
## 142          1347           10          418          623
## 143          1795           70          526          446
## 144          1536           66          526          709
## 145          1419           56          526          672
## 146          1410          123          526          811
## 147          1396          150          526          969
## 148          1472           82          526          684
## 149          1544          112          526          693
## 150          1453          141          502          779
## 151          1446          196          501          977
## 152          1486          107          512         1002
## 153          1515            0          221          648
## 154          1372          103          353          932
## 155          1365           98          526          958
```

```
## 156  1314  112  436  1031
## 157  1469  200  526  1071
## 158  2073   48  489   963
## 159  2000   35  526   559
## 160  1420   18  469   662
## 161  1862  168  526   630
## 162  1869  174  526   622
## 163  1729  123  526   554
## 164  1879  204  526   666
## 165  1720  193  526   672
## 166  1638  154  526   529
## 167  1562  161  526   785
## 168  1479  154  526   604
## 169  1322  147  427  1027
## 170  1462  163  526   903
## 171  1638   25  526   622
## 172  1583   37  526   613
## 173  1585   71  526   527
## 174  1777   62  526   361
## 175  1634  111  526   604
## 176  1451   67  526   623
## 177  1444   99  526   621
## 178  1466  124  526   805
## 179  1442  122  526   824
## 180  1400  148  526   953
## 181  1335  115  526   828
## 182  1432  199  526  1056
## 183  1474  156  526   926
## 184  1450  205  526  1008
## 185  1515  173  526   378
## 186  2033   43  476   576
## 187  1987   63  526   736
## 188  2347    8  385   970
## 189  1932   43  477   870
## 190  2545   93  526   401
## 191  1440   27  428   675
## 192  1548   18  461   577
## 193  1552   72  318   595
## 194  1635   68  433   545
## 195  1749   66  471   632
## 196  1414   62  497   554
## 197  1513  101  487   662
## 198  1439  146  526   745
## 199  1453  134  526   745
## 200  1576  137  526   964
## 201  1398  146  500  1105
## 202  1448  117  510   969
## 203  1315  103  526  1080
## 204  2297  157  526   990
## 205  1417  112  506   831
## 206  1352  125  526   906
## 207  1640  119  526  1119
```

```
## 208          1390          116          526          1032
## 209          1810           64          526           703
## 210          1654           42          526           725
## 211          2264           75          526           323
## 212          1860           32          454           408
## 213          1489           55          526           645
## 214          1329          102          526           802
## 215          1397          101          526           722
## 216          1653          133          526           827
## 217          1522          130          444           871
## 218          1550          133          474           878
## 219          1412           98          438           841
## 220          1361          112          526           971
## 221          1621          159          526          1231
## 222          1395          107          393          1060
## 223          1506          168          526          1032
## 224          1446          144          421          1080
## 225          1515           41          353           324
## 226          1332           95          526           780
## 227          1442          136          484           917
## 228          1413          157          526          1177
## 229          1416          130          526           977
## 230          2203           48          526          1030
## 231          1370           25          526           642
## 232          1766           84          526           437
## 233          1496          147          526           696
## 234          1496          196          504           792
## 235          1392          134          526           842
## 236          1326           80          515           850
## 237          1499          112          526           980
## 238          1345          141          471           973
## 239          1988           81          526           436
## 240          1682           43          526           727
## 241          1705          100          526           573
## 242          1667          110          526           524
## 243          1658           70          526           540
## 244          1541          101          451           781
## 245          1209           62          526          1025
## 246          1636          126          486           597
## 247          1470          122          426           817
## 248          1472          181          483           984
## 249          1374          100          454           653
## 250          1489          195          470          1094
## 251          1457          187          526          1142
## 252          3141           19          210           849
## 253          2031           69          526           558
## 254          1515          162          526           731
## 255          1299           99          511           893
## 256          1345          125          526           777
## 257          1381          102          463           976
## 258          1410          122          526           860
## 259          1423          172          420          1084
```

```
##     TEAM_FIELDING_E
## 1               140
## 2               135
## 3               156
## 4               124
## 5               616
## 6               572
## 7               490
## 8               328
## 9               226
## 10              184
## 11              200
## 12              150
## 13              137
## 14              125
## 15              115
## 16              146
## 17              154
## 18              115
## 19              301
## 20              232
## 21              166
## 22              155
## 23              179
## 24              160
## 25              105
## 26              102
## 27              163
## 28              232
## 29              218
## 30              199
## 31              178
## 32              167
## 33              146
## 34              178
## 35              130
## 36              156
## 37              136
## 38              133
## 39              137
## 40              143
## 41              130
## 42               93
## 43              163
## 44              470
## 45              413
## 46              537
## 47              500
## 48              284
## 49              281
## 50              237
## 51              177
```

```
## 52              154
## 53              162
## 54              119
## 55              174
## 56              114
## 57              411
## 58              250
## 59              332
## 60              304
## 61              200
## 62              142
## 63              126
## 64              100
## 65               94
## 66              321
## 67              414
## 68              373
## 69              327
## 70              314
## 71              193
## 72              127
## 73              144
## 74              119
## 75              144
## 76              168
## 77              116
## 78              125
## 79              270
## 80              262
## 81              179
## 82              184
## 83              191
## 84              173
## 85              163
## 86              133
## 87              114
## 88              131
## 89              123
## 90              110
## 91              445
## 92              163
## 93              398
## 94              285
## 95              286
## 96              296
## 97              271
## 98              245
## 99              207
## 100             221
## 101             187
## 102             203
## 103             138
```

```
## 104             130
## 105             135
## 106             652
## 107             163
## 108             103
## 109             101
## 110             609
## 111             155
## 112             150
## 113             141
## 114             127
## 115             129
## 116             123
## 117             126
## 118             130
## 119             125
## 120             581
## 121             344
## 122             363
## 123             287
## 124             254
## 125             246
## 126             219
## 127             221
## 128             239
## 129             203
## 130             178
## 131             192
## 132             145
## 133             179
## 134             146
## 135              73
## 136             163
## 137             127
## 138             137
## 139              89
## 140             131
## 141             336
## 142             294
## 143             193
## 144             222
## 145             137
## 146             135
## 147             172
## 148             146
## 149             134
## 150             120
## 151             118
## 152              93
## 153             163
## 154             166
## 155             112
```

```
## 156             151
## 157             104
## 158             680
## 159             488
## 160             352
## 161             219
## 162             177
## 163             164
## 164             173
## 165             178
## 166             126
## 167             129
## 168             124
## 169             126
## 170             112
## 171             280
## 172             224
## 173             187
## 174             204
## 175             224
## 176             150
## 177             147
## 178             175
## 179             134
## 180             137
## 181             145
## 182             142
## 183             105
## 184             102
## 185             163
## 186             643
## 187             566
## 188             163
## 189             658
## 190             456
## 191             427
## 192             293
## 193             246
## 194             195
## 195             214
## 196             246
## 197             221
## 198             116
## 199             151
## 200             136
## 201             158
## 202             113
## 203             113
## 204             130
## 205             121
## 206             152
## 207             108
```

```
## 208             105
## 209             471
## 210             570
## 211             441
## 212             392
## 213             243
## 214             190
## 215             175
## 216             135
## 217             137
## 218             145
## 219             128
## 220             126
## 221             136
## 222             140
## 223             132
## 224             104
## 225             163
## 226             141
## 227             135
## 228             141
## 229             109
## 230             163
## 231             244
## 232             198
## 233             144
## 234             139
## 235             178
## 236             157
## 237             169
## 238             108
## 239             523
## 240             276
## 241             208
## 242             118
## 243             143
## 244             122
## 245             175
## 246             148
## 247             139
## 248             130
## 249             131
## 250              97
## 251             107
## 252             163
## 253             492
## 254             163
## 255             139
## 256             163
## 257             124
## 258             159
## 259             131
```

References

- https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html (https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html)
- https://r-coder.com/correlation-plot-r/ (https://r-coder.com/correlation-plot-r/)