Homework # 1 (Write-up)

Al Haque

2023-02-25

R write-up

Data Exploration:

The dataset I analyzed contains information about a professional baseball team from 1871 to 2006, with 17 numeric columns and 2,076 observations. One column TEAM_BATTING_HBP, has a significant amount of missing data. I noticed that some columns had a large number of outliers, particularly TEAM_PITCHING_H which indicates hits allowed.The distribution of the predictor variable TARGET_WINS was normally distributed, while some of the other variables were skewed.The correlation matrix revealed high correlation between some variables such as TEAM_BATTING_HR and TEAM_PITCHING_HR but most of the matrix had missing data. After cleaning the data, I noticed high correlation between some predictors and thus avoided including them in the regression model. But most of the variables were not correlated with the predictor TARGET_WINS.

Data Preparation

I deleted the columns TEAM_BATTING_HBP and TEAM_BASERUN_CS since the majority of their observations contained missing values.For the other columns with missing data, I used the MICE Package in R to impute the missing values. Specifically, I used a mix of predictive mean matching,classification and regression trees on TEAM_FIELDING_DP and TEAM_PITCHING_SO,since they had the most missing values after removing the other columns. I then removed the remaining variables and observations with negative or zero values since I wanted to perform a Box-Cox transformation on the data.

Build Models

I created five linear regression models. The first model included all predictor variables against the response. Then, I used stepwise selection to remove insignificant predictors. Next, I applied the box-cox transformation and transformed the y variable to the power of 1.3536, which maximized the log-likelihood of the transformed data and improved the model slightly. The coefficients of the model had both positive and negative slopes. Since some predictors increase/decrease a team's chance of winning. For instance, in my final model, TEAM_BATTING_H had a slope of 0.263 meaning that for every base hit by the batter, the win increased by 0.263. This outcome was expected as a hit by the batter can increase their chances of scoring and ultimately winning the game.

Model Selection:

For my final model, I selected the model with the box-cox transformation. This model included all significant variables and had the lowest-root-mean-squared error (RMSE) compared to the other models,with a score of 12.43907.The diagnostic checks for this model showed that all assumptions were met, as the residuals were clearly scattered with no distinct patterns in the plot, and the QQ plot was normal. Additionally the F-statictics for the model was 158 and the adjusted R-squared value was 0.33.

The equation of the model is:

```
Y^.13536 = 83.63 + TEAM_BATTING_H * 0.263 + TEAM_BATTING_HR * 0.49 +
```

0.0709 * TEAM_BATTING_BB + TEAM_BATTING_SO * (-0.09) + TEAM_BASERUN_SB * 0.293 + TEAM_FIELDING_E * (-0.188) + TEAM_FIELDING_DP + (-0.693)

Using the model for my predictions I had to apply the inverse box-cox transformation in order to get the actual predicted value for the TARGET_WINS so that I can better interpret the values. I.e (Y^(1/.13536).

Sources Citiaton:

Here were some websites that helped me with my analysis and the data imputation:

Wu, Songhao. "Multi-Collinearity in Regression." Medium, Towards Data Science, 5 June 2021, https://towardsdatascience.com/multi-collinearity-in-regression-fe7a2c1467ea.

"Imputation in R: Top 3 Ways for Imputing Missing Data." Machine Learning, R Programming, 8 Oct. 2021, https://appsilon.com/imputation-in-r/.

Appendix:

Here is my R code stored as an appendix:

Introduction

(Data Exploration):

The training dataset contains seventeen columns and two thousand seventy six observations about a professional baseball team throughout the years of 1871 to 2006

```r
## Step 1 call in your libraries and import the data from csv and read it into R
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
training <- read.csv('https://raw.githubusercontent.com/AldataSci/Baseball-Data/main/moneyball-training-
```

Looking at the structure of the dataset we can see they are all integer columns and one of the columns TEAM_BATTING_HBP contains a lot of NA values for the head of the data..

```
str(training)
```

```
## 'data.frame':    2276 obs. of  17 variables:
##  $ INDEX           : int  1 2 3 4 5 6 7 8 11 12 ...
##  $ TARGET_WINS     : int  39 70 86 70 82 75 80 85 86 76 ...
##  $ TEAM_BATTING_H  : int  1445 1339 1377 1387 1297 1279 1244 1273 1391 1271 ...
##  $ TEAM_BATTING_2B : int  194 219 232 209 186 200 179 171 197 213 ...
##  $ TEAM_BATTING_3B : int  39 22 35 38 27 36 54 37 40 18 ...
##  $ TEAM_BATTING_HR : int  13 190 137 96 102 92 122 115 114 96 ...
##  $ TEAM_BATTING_BB : int  143 685 602 451 472 443 525 456 447 441 ...
##  $ TEAM_BATTING_SO : int  842 1075 917 922 920 973 1062 1027 922 827 ...
##  $ TEAM_BASERUN_SB : int  NA 37 46 43 49 107 80 40 69 72 ...
##  $ TEAM_BASERUN_CS : int  NA 28 27 30 39 59 54 36 27 34 ...
##  $ TEAM_BATTING_HBP: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TEAM_PITCHING_H : int  9364 1347 1377 1396 1297 1279 1244 1281 1391 1271 ...
##  $ TEAM_PITCHING_HR: int  84 191 137 97 102 92 122 116 114 96 ...
##  $ TEAM_PITCHING_BB: int  927 689 602 454 472 443 525 459 447 441 ...
##  $ TEAM_PITCHING_SO: int  5456 1082 917 928 920 973 1062 1033 922 827 ...
##  $ TEAM_FIELDING_E : int  1011 193 175 164 138 123 136 112 127 131 ...
##  $ TEAM_FIELDING_DP: int  NA 155 153 156 168 149 186 136 169 159 ...
```

A quick glance at the summary statistics of the column.

```
## OK one of the columns has over 2,085 missing values out of 2276 of its columns..
## TEAM_BATTING_HBP which is the column for Batters hit by pitch (may have to remove this column..)
summary(training)
```

```
##      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
```

We can see that HBP contains 2085 missing values followed by TEAM_BASERUN_CS so I may have to omit those columns from the dataset.

```r
## Easier to see all the missing values
sapply(training,function(x) sum(is.na(x)))
```

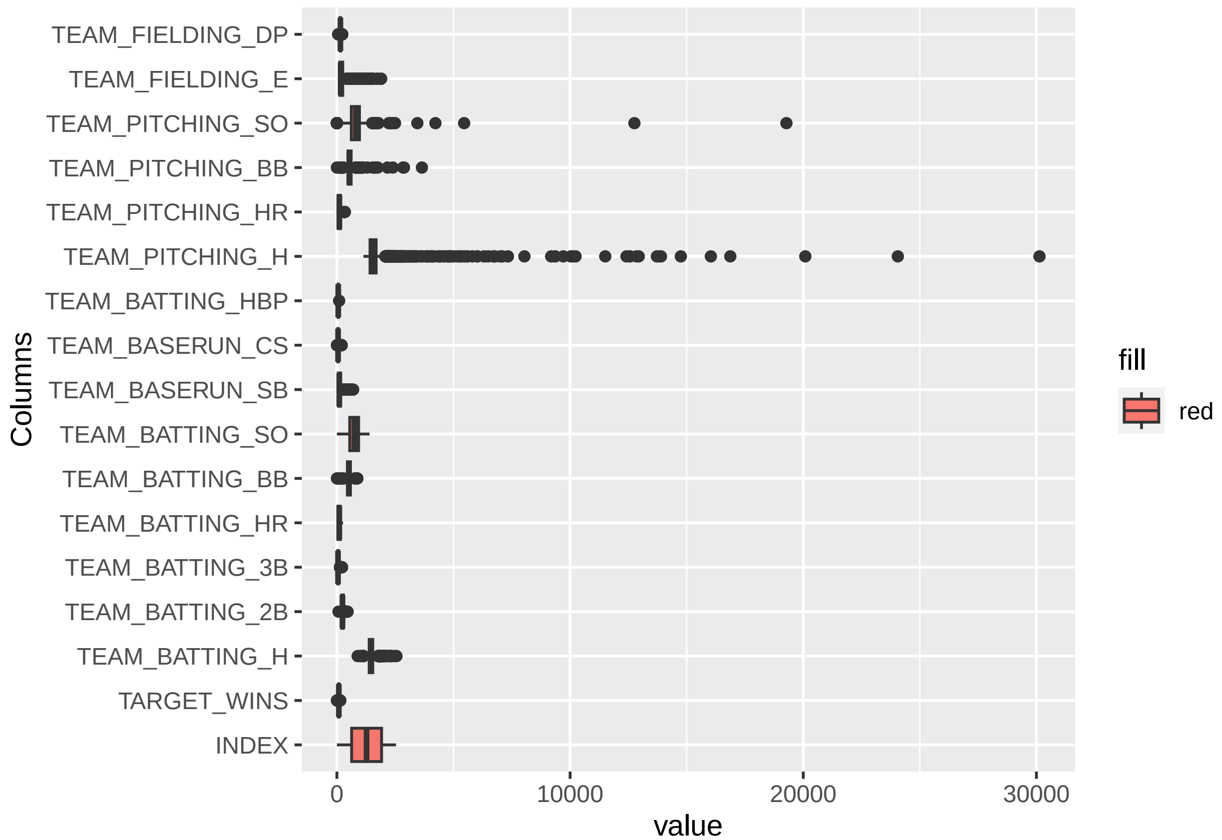
```
##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
##                0                0                0                0
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##                0                0                0              102
##  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H
##              131              772             2085                0
## TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##                0                0              102                0
## TEAM_FIELDING_DP
##              286
```

From the boxplot the column of TEAM_PITCHING_H has a lot of outliers, I may consider removing this column from the model in order to not sway it.

```
## Let's try the ggplot method and melt-method..
data_long <- melt(training)
```

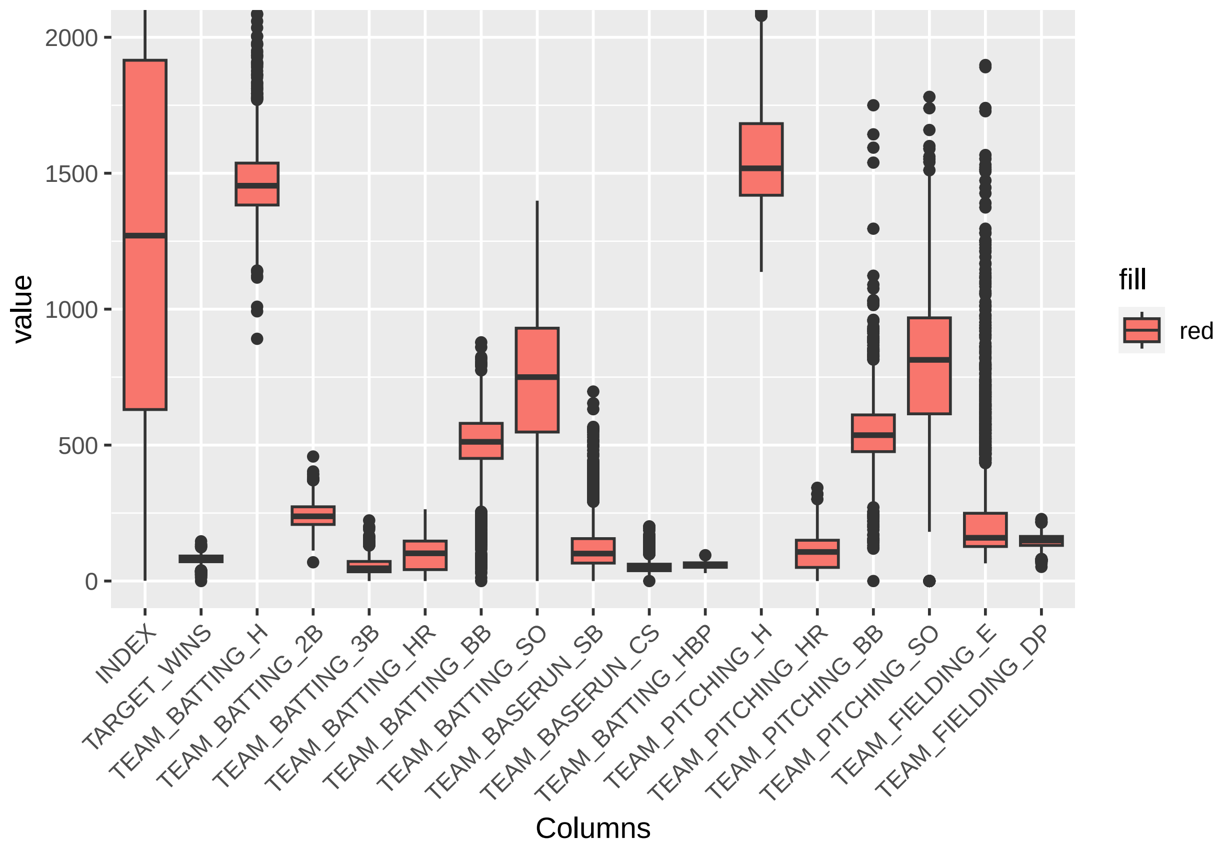
```
## No id variables; using all as measure variables
```

```
##plot boxplot with ggplot.. ## there are a lot of outliers in TEAM_PITCHING_H
gg <- ggplot(data_long,aes(x=variable,y=value,fill = "red")) + geom_boxplot() + coord_flip() + xlab("Co
gg
```




```
gg + coord_cartesian(ylim = c(0,2000)) + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```

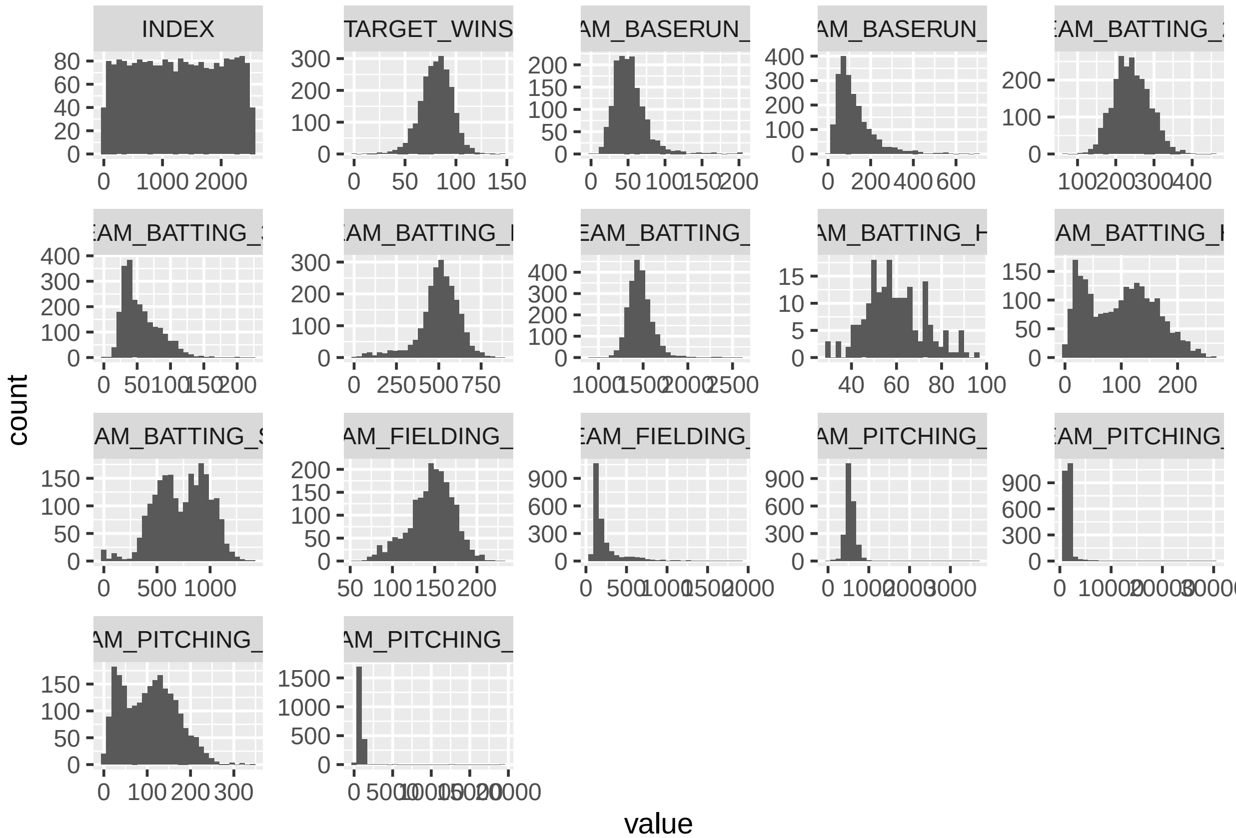



```
data_gathered <- training %>%
  gather(variable,value)
```

The histograms have various distribution but the predictor variable TARGET_WINS is normally distributed but some of the others are skewed like TEAM_FIELDING_E and etc.

```
## each panel can have its own scale when we use scale = "Free"
histograms <- ggplot(data_gathered,aes(x=value)) + geom_histogram() +
  facet_wrap(~variable,scale="free")
histograms
```

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

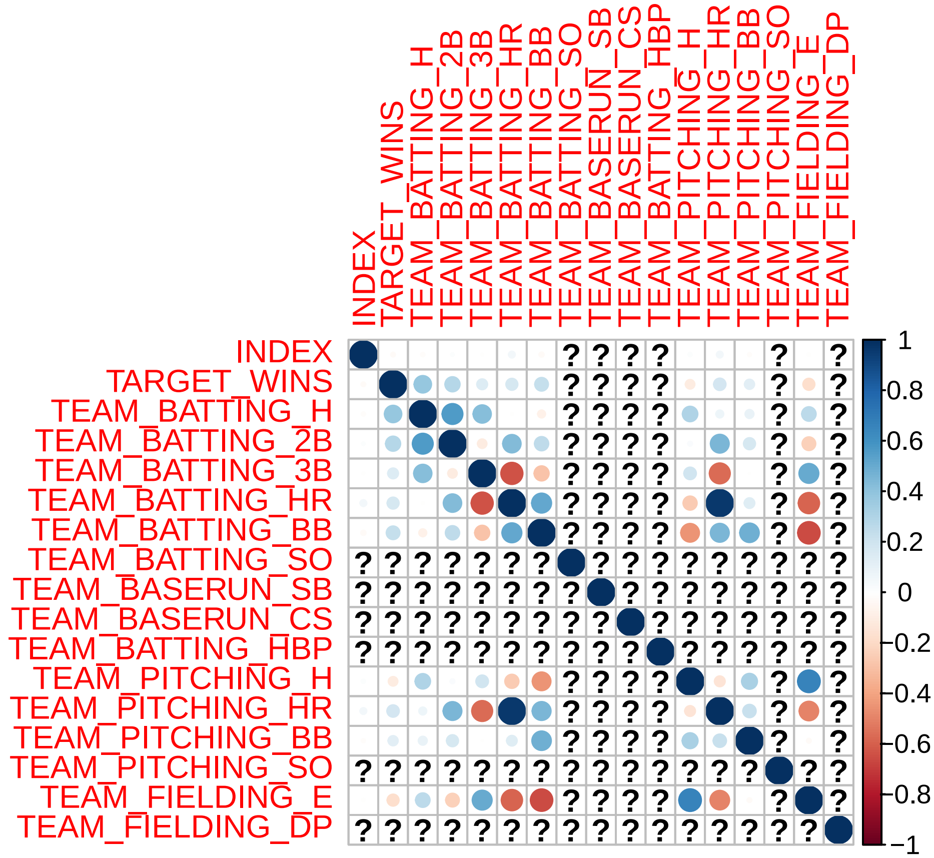


The correlation matrix shows a lot of question marks which shows missing data in the columns,

```
## Let's create a correlation matrix with our data..
sum(is.na(training))
```

```
## [1] 3478
```

```
## there are a lot of missing data in these columns... i'm gonna have to remove some of those columns..
corrplot(cor(training))
```



Part II Data Preparation:

Removal of NA values

I've removed the columns of HBP and CS since they contained a lot of missing values

```
## Cleaning the data and imputating some of the data.. i'm going to remove columns TEAM_BATTING_HBP and

Training <- training %>%
  dplyr::select(-c(TEAM_BATTING_HBP,TEAM_BASERUN_CS))
```

```
sapply(Training,function(x) sum(is.na(x)))
```

```
##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
##                0                0                0                0
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##                0                0                0              102
##  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##              131                0                0                0
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##              102                0              286
```

Imputation using MICE

I am going to try imputing the missing values with the MICE package and I will use predictive mean matching, cart: Classification and regression trees and lasso linear regression and for each I will see which imputation method closely resembles the distribution of the normal data and choose that method to impute the missing values.

```r
## Now I will imputate the data with the mice package..
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```r
mice_imputed <- data.frame(
original = Training$TEAM_FIELDING_DP,
imp_pmm = complete(mice(Training,method ="pmm"))$TEAM_FIELDING_DP,
imp_cart = complete(mice(Training,method ="cart"))$TEAM_FIELDING_DP,
imp_lasso = complete(mice(Training,method ="lasso.norm"))$TEAM_FIELDING_DP
)
```

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

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

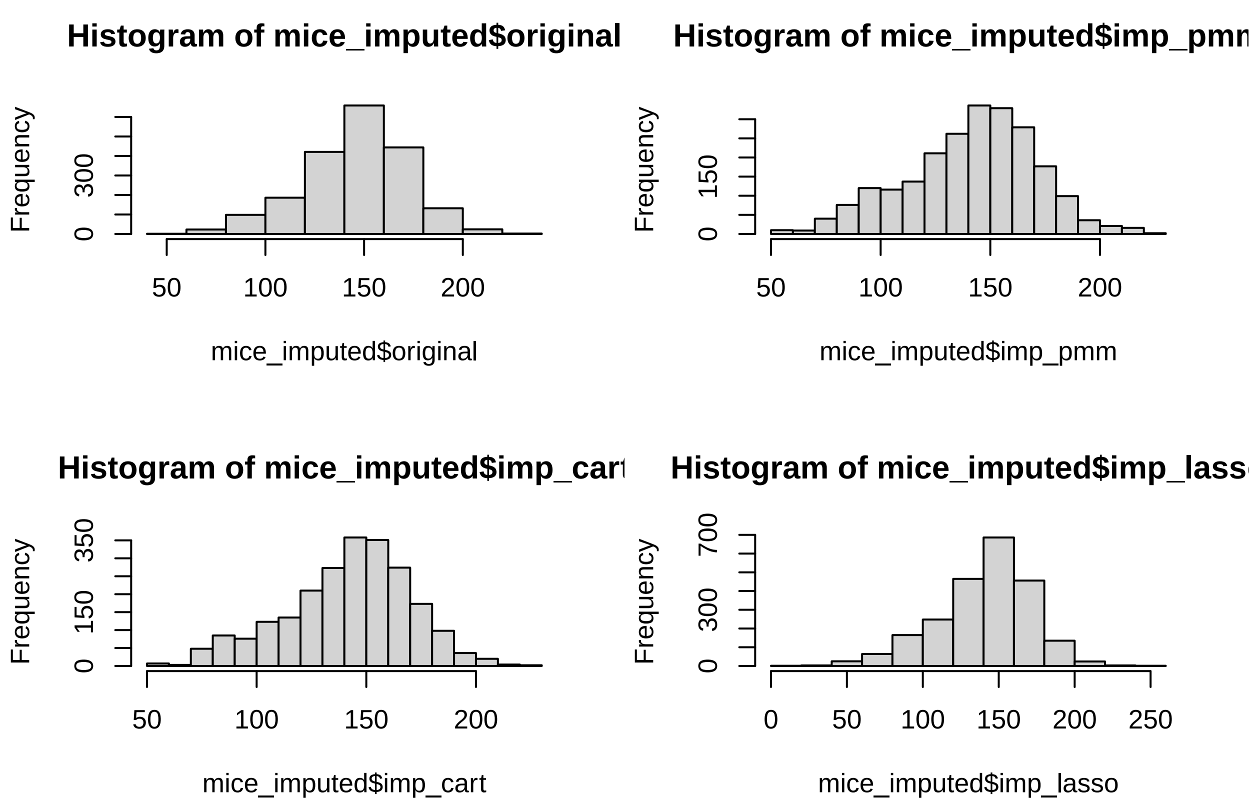
```
##    5    3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##    5    4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##    5    5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
```

```
head(mice_imputed)
```

```
##   original imp_pmm imp_cart imp_lasso
## 1       NA     162       94  107.9932
## 2      155     155      155  155.0000
## 3      153     153      153  153.0000
## 4      156     156      156  156.0000
## 5      168     168      168  168.0000
## 6      149     149      149  149.0000
```

I am going to compare the distribution of the original and then figure which distribution resembles the original.

```
## compare the distribution between each imputation and see which one resembles the original the most..
## I think the imp_cart looks smiliar to the original histogram so I will use those values.
par(mfrow=c(2,2))
hist(mice_imputed$original)
hist(mice_imputed$imp_pmm)
hist(mice_imputed$imp_cart)
hist(mice_imputed$imp_lasso)
```



```r
## replace the values with the imputed values..
Training$TEAM_FIELDING_DP <- mice_imputed$imp_cart
```

```r
## now I will imputate the rest of the columns with the same method..
sapply(Training,function(x) sum(is.na(x)))
```

```
##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
##                0                0                0                0
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##                0                0                0              102
##  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##              131                0                0                0
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##              102                0                0
```

```r
## i will imputate the TEAM_BASERUN_SB which is stolen bases..
mice_imputed2 <- data.frame(
original = Training$TEAM_BASERUN_SB,
imp_pmm = complete(mice(Training,method ="pmm"))$TEAM_BASERUN_SB,
imp_cart = complete(mice(Training,method ="cart"))$TEAM_BASERUN_SB,
imp_lasso = complete(mice(Training,method ="lasso.norm"))$TEAM_BASERUN_SB
)
```

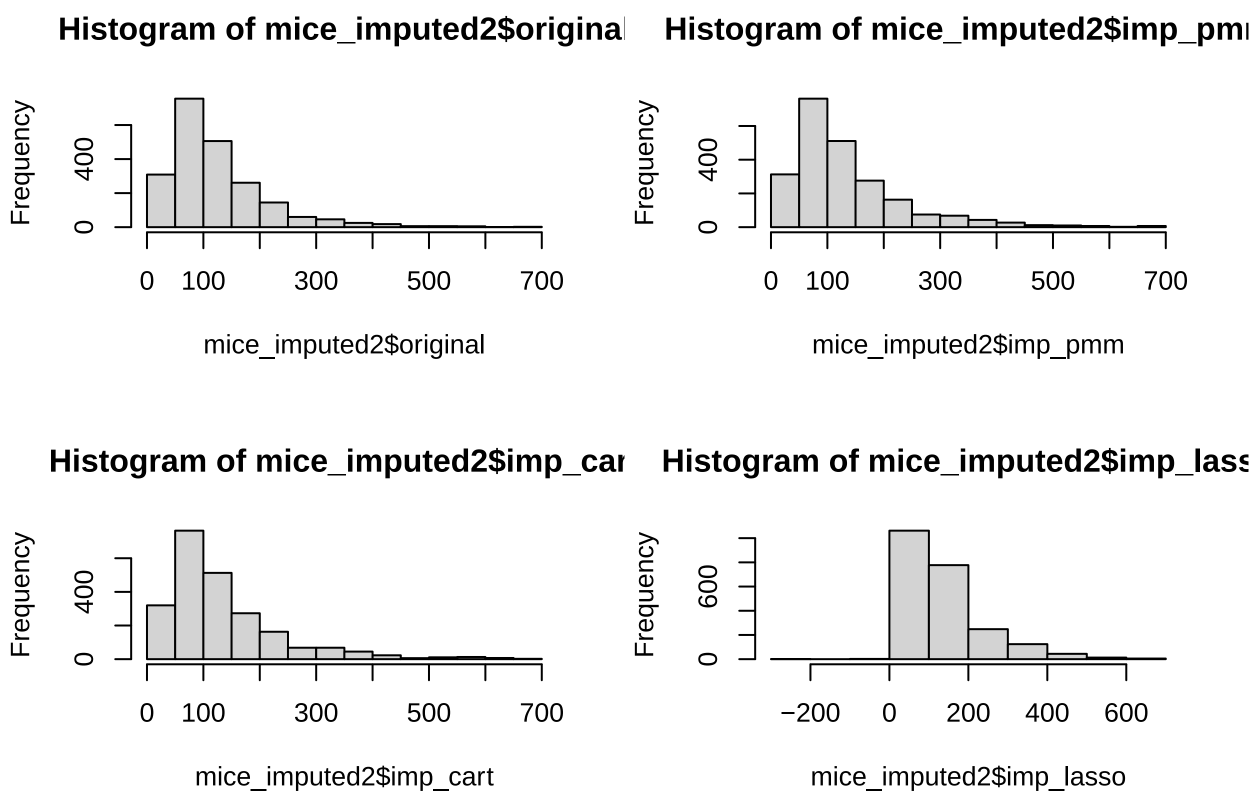
```
##
##  iter imp variable
##   1   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##   1   5  TEAM_BATTING_SO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

```
head(mice_imputed2)
```

```
##   original imp_pmm imp_cart imp_lasso
## 1       NA     170      226  249.1743
## 2       37      37       37   37.0000
## 3       46      46       46   46.0000
## 4       43      43       43   43.0000
## 5       49      49       49   49.0000
## 6      107     107      107  107.0000
```

```
## I will impute that value with imp_cart since they resemble the original histogram..
par(mfrow=c(2,2))
hist(mice_imputed2$original)
hist(mice_imputed2$imp_pmm)
hist(mice_imputed2$imp_cart)
hist(mice_imputed2$imp_lasso)
```




```
## imputate BASERUN_SB with this value since the distributions looks smiliar
Training$TEAM_BASERUN_SB <- mice_imputed2$imp_pmm
```

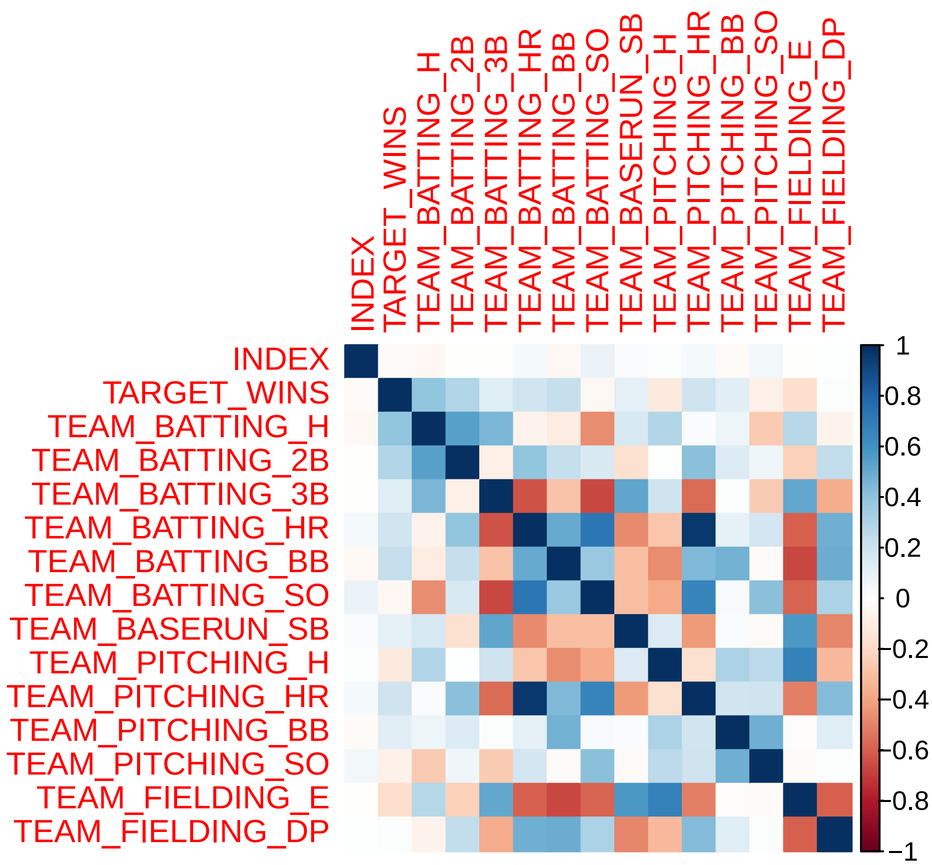
```
## looking at the empty values again I think i should be fine with it this time..
sapply(Training,function(x) sum(is.na(x)))
```

```
##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
```

```
##                0                0                0                0
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##                0                0                0              102
##  TEAM_BASERUN_SB  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##                0                0                0                0
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##              102                0                0
```

```
## now I want to look at the correlation matrix again and see if I can gleam any valuable information..
Training <- na.omit(Training)

corrplot(cor(Training),method = "color")
```



Part III (Model-Creation)

```
## I am going to split the training data set into training and testing datasets...
## 70% in Training and 30% in Testing..
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'

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

```
set.seed(123)
index <- createDataPartition(Training$TARGET_WINS,p=0.7,list = FALSE)

Ttraining <- Training[index,]
Ttest <- Training[-index,]
```

Model I (All the Predictors minus the Index)

```
## It went up only a little bit.. but that's fine..
mod1 <- lm(TARGET_WINS ~ .-INDEX,data=Ttraining)
summary(mod1)
```

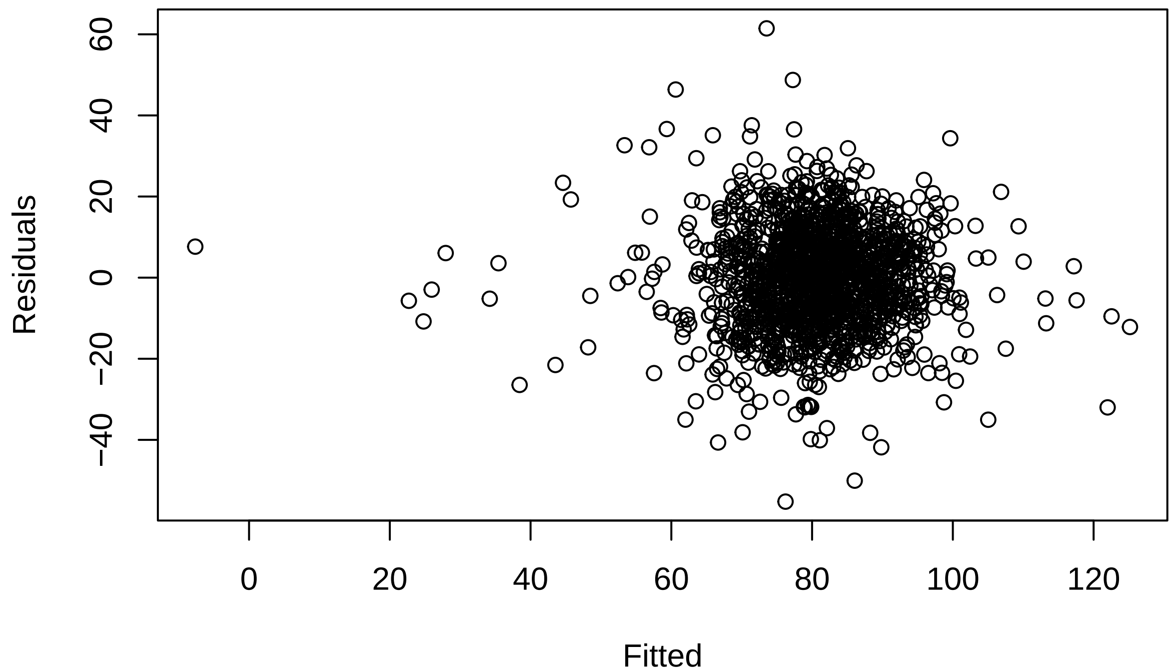
```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = Ttraining)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.065  -8.200   0.437   8.093  61.651
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.0797898  6.1342483   5.393 8.05e-08 ***
## TEAM_BATTING_H    0.0447046  0.0041959  10.654  < 2e-16 ***
## TEAM_BATTING_2B  -0.0298541  0.0108428  -2.753  0.00597 **
## TEAM_BATTING_3B   0.0532230  0.0196231   2.712  0.00676 **
## TEAM_BATTING_HR   0.0687673  0.0297574   2.311  0.02097 *
## TEAM_BATTING_BB   0.0123637  0.0065007   1.902  0.05737 .
## TEAM_BATTING_SO  -0.0164987  0.0029327  -5.626 2.20e-08 ***
## TEAM_BASERUN_SB   0.0498024  0.0048285  10.314  < 2e-16 ***
## TEAM_PITCHING_H   0.0002335  0.0004628   0.504  0.61403
## TEAM_PITCHING_HR  0.0251116  0.0259878   0.966  0.33406
## TEAM_PITCHING_BB -0.0029723  0.0045075  -0.659  0.50973
## TEAM_PITCHING_SO  0.0030900  0.0010069   3.069  0.00219 **
## TEAM_FIELDING_E  -0.0383253  0.0032218 -11.896  < 2e-16 ***
## TEAM_FIELDING_DP -0.1141424  0.0156104  -7.312 4.25e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.46 on 1510 degrees of freedom
## Multiple R-squared:  0.3781, Adjusted R-squared:  0.3728
## F-statistic: 70.63 on 13 and 1510 DF,  p-value: < 2.2e-16
```

Model II (Getting rid of the not signficant variables)

```r
## I will get rid of the not so signficant variables so TEAM_PITCHING_HR and TEAM_PITCHING_BB and the R
mod2 <- lm(TARGET_WINS ~ .-INDEX-TEAM_PITCHING_H-TEAM_PITCHING_HR-TEAM_PITCHING_BB,data=Ttraining)
summary(mod2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX - TEAM_PITCHING_H - TEAM_PITCHING_HR -
##     TEAM_PITCHING_BB, data = Ttraining)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.225  -8.096   0.425   8.097  61.462
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      32.4857932  6.0123335   5.403 7.59e-08 ***
## TEAM_BATTING_H    0.0453569  0.0041369  10.964  < 2e-16 ***
## TEAM_BATTING_2B  -0.0297532  0.0107860  -2.759  0.00588 **
## TEAM_BATTING_3B   0.0541551  0.0191700   2.825  0.00479 **
## TEAM_BATTING_HR   0.0959229  0.0111630   8.593  < 2e-16 ***
## TEAM_BATTING_BB   0.0087541  0.0037633   2.326  0.02014 *
## TEAM_BATTING_SO  -0.0162013  0.0027826  -5.822 7.07e-09 ***
## TEAM_BASERUN_SB   0.0487345  0.0044967  10.838  < 2e-16 ***
## TEAM_PITCHING_SO  0.0027849  0.0005906   4.716 2.63e-06 ***
## TEAM_FIELDING_E  -0.0373546  0.0025575 -14.606  < 2e-16 ***
## TEAM_FIELDING_DP -0.1139331  0.0155588  -7.323 3.93e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.46 on 1513 degrees of freedom
## Multiple R-squared:  0.3776, Adjusted R-squared:  0.3735
## F-statistic:  91.8 on 10 and 1513 DF,  p-value: < 2.2e-16
```

```r
plot(fitted(mod2),residuals(mod2),xlab="Fitted",ylab="Residuals")
```



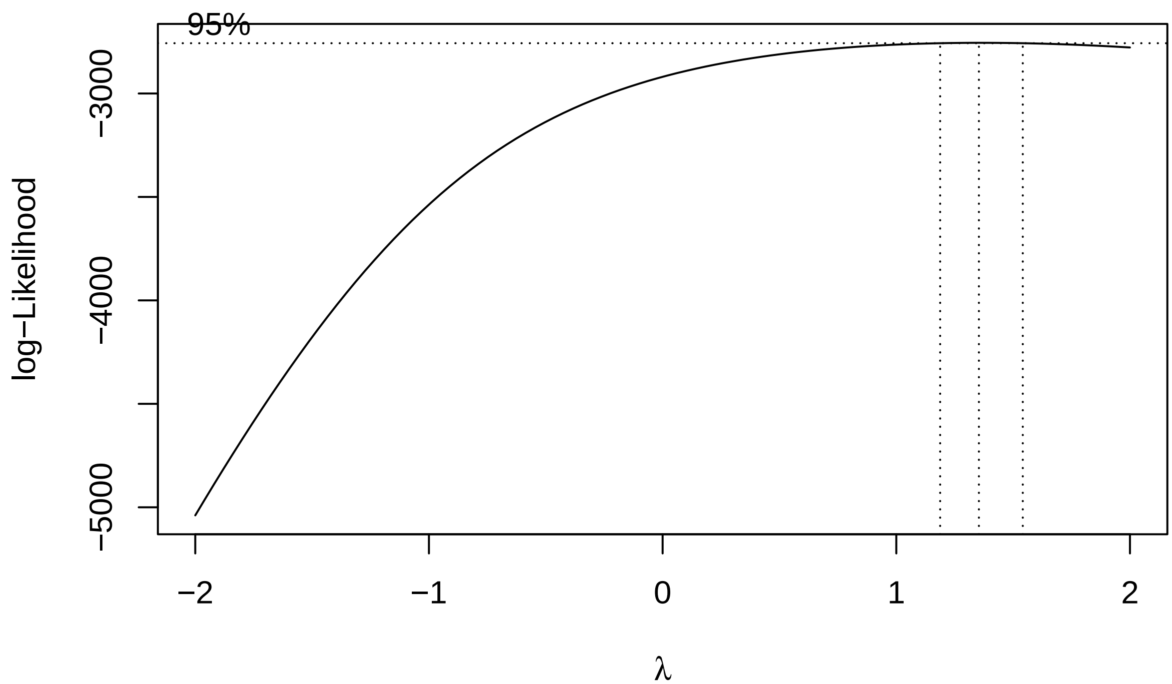

```
## attempt a box-cox transformation..
Ttraining <- Ttraining %>%
  filter(TARGET_WINS != 0)
Ttest <- Ttest %>%
  filter(TARGET_WINS != 0)
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

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

```
set.seed(123)
bcox <-boxcox(mod2,plotit = T)
```




```r
val <- cbind(bcox$x,bcox$y)

## sort the values in ascending-order.. our lambda value is 1.1919 that maxmizes the log-likelihood of t
head(val[order(-bcox$y),])
```

```
##          [,1]      [,2]
## [1,] 1.353535 -2755.097
## [2,] 1.393939 -2755.155
## [3,] 1.313131 -2755.239
## [4,] 1.434343 -2755.409
## [5,] 1.272727 -2755.587
## [6,] 1.474747 -2755.854
```

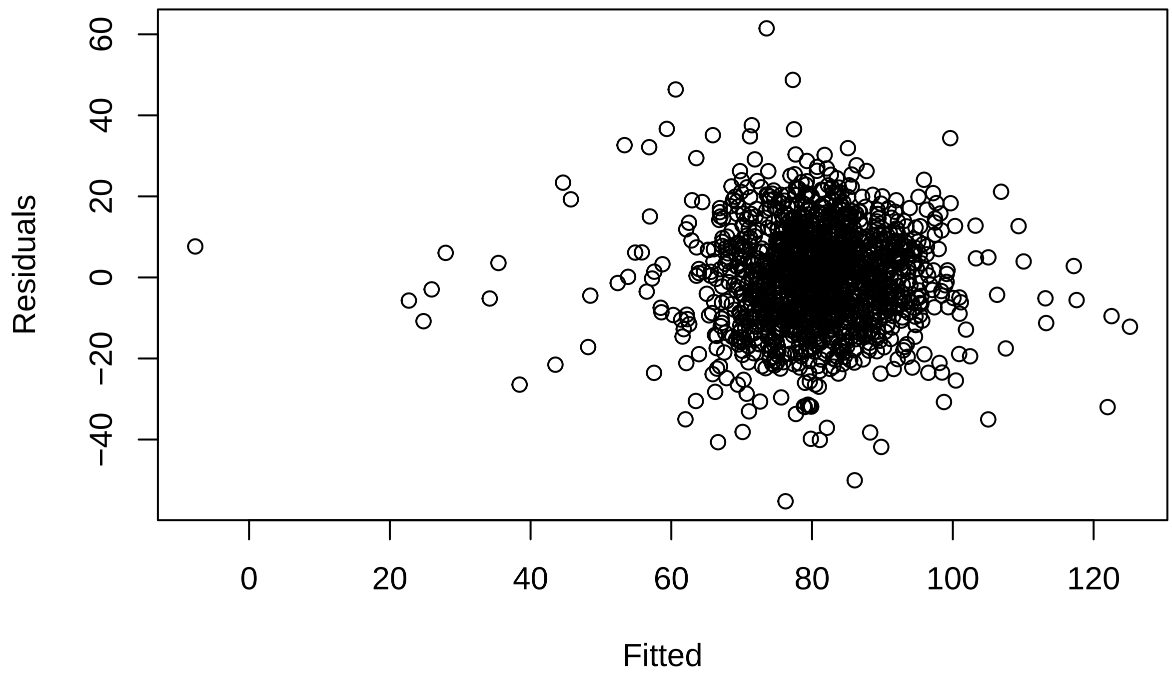
Model III (Box-Cox Transformation)

```r
## Let use the lambda value on our model to see if it improves the model even if its a little bit.
bmod3 <- lm(TARGET_WINS ^(1.3536) ~ .-INDEX-TEAM_PITCHING_H-TEAM_PITCHING_HR-TEAM_PITCHING_BB,data=Ttrai
summary(bmod3)
```

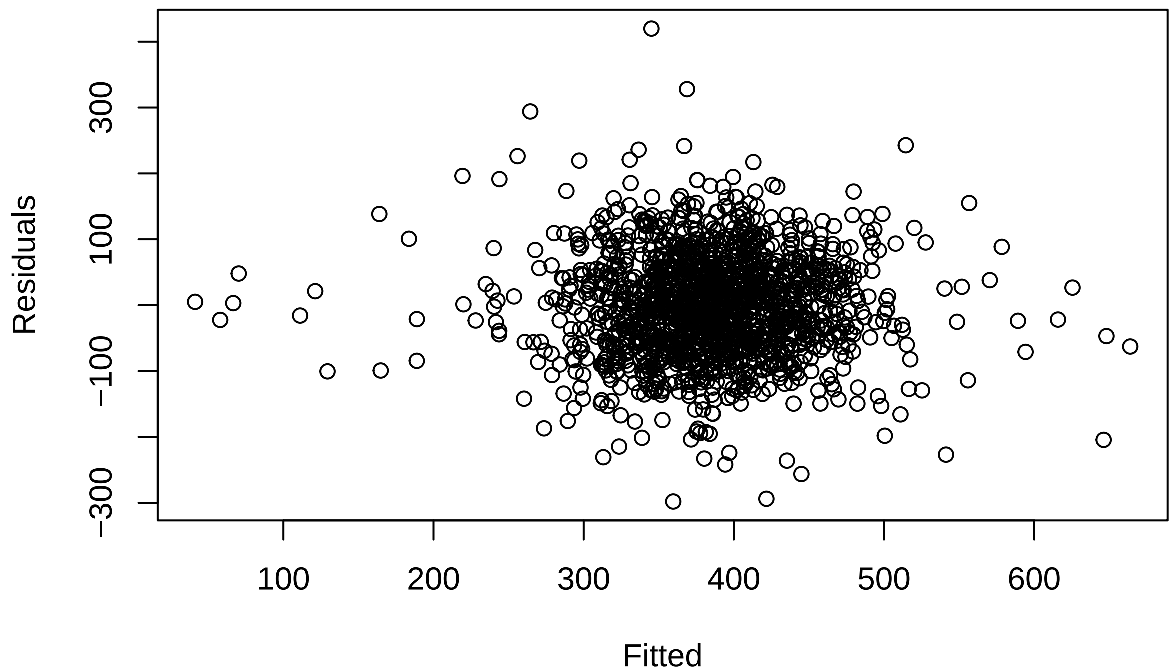
```
##
## Call:
## lm(formula = TARGET_WINS^(1.3536) ~ . - INDEX - TEAM_PITCHING_H -
##     TEAM_PITCHING_HR - TEAM_PITCHING_BB, data = Ttraining)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -298.01  -52.96    0.63   51.10  419.81
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      80.090953  38.438293   2.084  0.03736 *
## TEAM_BATTING_H    0.286111   0.026653  10.735  < 2e-16 ***
## TEAM_BATTING_2B  -0.190792   0.068632  -2.780  0.00550 **
## TEAM_BATTING_3B   0.336196   0.121006   2.778  0.00553 **
## TEAM_BATTING_HR   0.620028   0.070630   8.778  < 2e-16 ***
## TEAM_BATTING_BB   0.057501   0.023761   2.420  0.01564 *
## TEAM_BATTING_SO  -0.106011   0.017617  -6.018 2.22e-09 ***
## TEAM_BASERUN_SB   0.298591   0.028520  10.470  < 2e-16 ***
## TEAM_PITCHING_SO  0.018111   0.003741   4.841 1.42e-06 ***
## TEAM_FIELDING_E  -0.219602   0.016611 -13.221  < 2e-16 ***
## TEAM_FIELDING_DP -0.729388   0.098202  -7.427 1.84e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.58 on 1512 degrees of freedom
## Multiple R-squared:  0.3582, Adjusted R-squared:  0.354
## F-statistic: 84.41 on 10 and 1512 DF,  p-value: < 2.2e-16
```

```r
## it looks a bit better
plot(fitted(mod2),residuals(mod2),xlab="Fitted",ylab="Residuals")
```




```
plot(fitted(bmod3),residuals(bmod3),xlab="Fitted",ylab="Residuals")
```



Model Four (Removing the less signficant variables..)

```
## This looks good I think, I removed the other least signficant variables..
bmod4 <- lm(TARGET_WINS ^(1.3536) ~ .-INDEX-TEAM_PITCHING_H-TEAM_PITCHING_HR-TEAM_PITCHING_BB-TEAM_BATTI
summary(bmod4)
```

```
##
## Call:
## lm(formula = TARGET_WINS^(1.3536) ~ . - INDEX - TEAM_PITCHING_H -
##     TEAM_PITCHING_HR - TEAM_PITCHING_BB - TEAM_BATTING_3B, data = Training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -301.51  -53.51   -0.25   51.45  400.37
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     71.692402  32.416374   2.212 0.027098 *
## TEAM_BATTING_H   0.293285   0.020862  14.058  < 2e-16 ***
## TEAM_BATTING_2B -0.136742   0.055850  -2.448 0.014429 *
## TEAM_BATTING_HR  0.529705   0.055628   9.522  < 2e-16 ***
## TEAM_BATTING_BB  0.072091   0.019410   3.714 0.000209 ***
## TEAM_BATTING_SO -0.107557   0.014858  -7.239 6.26e-13 ***
## TEAM_BASERUN_SB  0.316364   0.022578  14.012  < 2e-16 ***
```

```
## TEAM_PITCHING_SO  0.016180   0.003599   4.495 7.32e-06 ***
## TEAM_FIELDING_E  -0.206376   0.013554 -15.226  < 2e-16 ***
## TEAM_FIELDING_DP -0.710063   0.081535  -8.709  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.71 on 2164 degrees of freedom
## Multiple R-squared:  0.3527, Adjusted R-squared:   0.35
## F-statistic:   131 on 9 and 2164 DF,  p-value: < 2.2e-16
```

Model Five (Removing the more of the less signficant variables..)

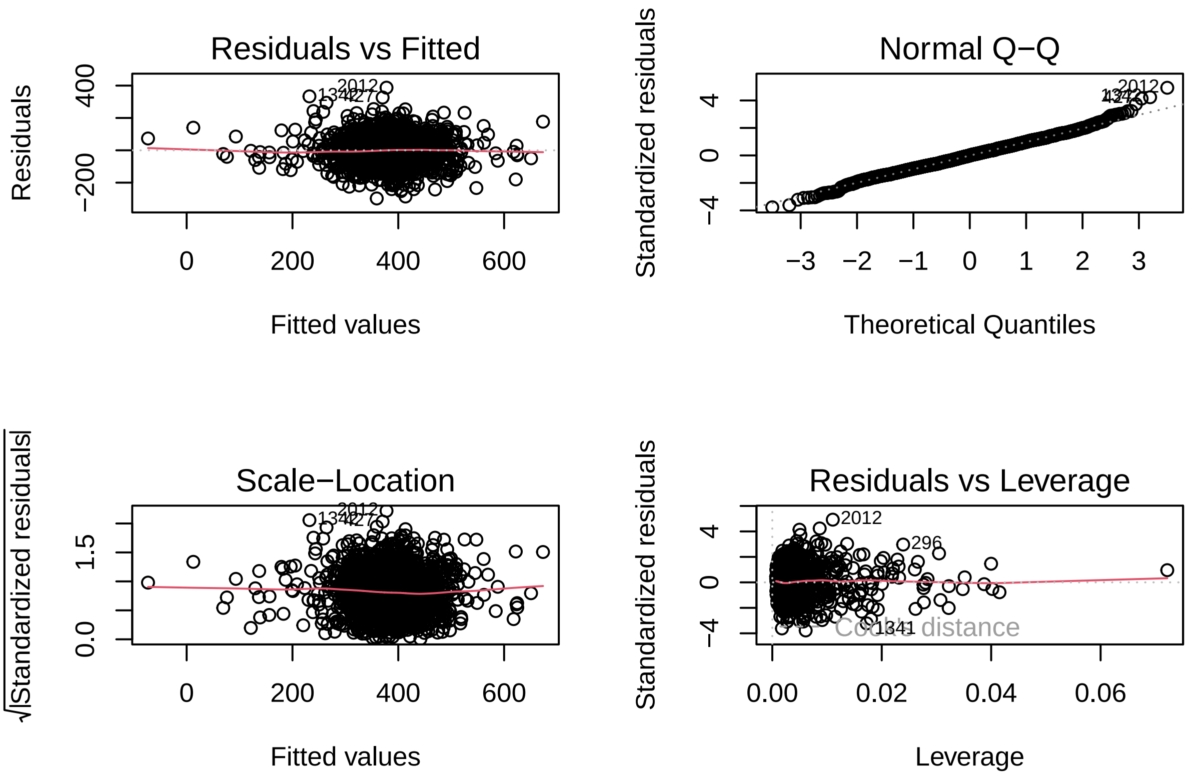
```
## Here I removed the least signficant variables and I'm curious now..
bmod5 <- lm(TARGET_WINS ^(1.3536) ~ .-INDEX-TEAM_PITCHING_H-TEAM_PITCHING_HR-TEAM_PITCHING_BB-TEAM_BATTI
summary(bmod5)
```

```
##
## Call:
## lm(formula = TARGET_WINS^(1.3536) ~ . - INDEX - TEAM_PITCHING_H -
##     TEAM_PITCHING_HR - TEAM_PITCHING_BB - TEAM_BATTING_3B - TEAM_BATTING_2B -
##     TEAM_PITCHING_SO, data = Training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -297.68  -53.64   -0.07   51.55  387.18
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      91.45928   30.96522   2.954 0.003175 **
## TEAM_BATTING_H    0.25769    0.01557  16.551  < 2e-16 ***
## TEAM_BATTING_HR   0.50531    0.05563   9.083  < 2e-16 ***
## TEAM_BATTING_BB   0.06879    0.01949   3.530 0.000425 ***
## TEAM_BATTING_SO  -0.09250    0.01346  -6.875  8.1e-12 ***
## TEAM_BASERUN_SB   0.31547    0.02257  13.980  < 2e-16 ***
## TEAM_FIELDING_E  -0.18928    0.01315 -14.396  < 2e-16 ***
## TEAM_FIELDING_DP -0.69850    0.08176  -8.544  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.1 on 2166 degrees of freedom
## Multiple R-squared:  0.3457, Adjusted R-squared:  0.3436
## F-statistic: 163.5 on 7 and 2166 DF,  p-value: < 2.2e-16
```

Looking at the diagnostics

I think the model fits all the assumptions but with some outliers here and there in the cook's distance chart.

```
par(mfrow=c(2,2))
plot(bmod5)
```



(Part IV) Model selection.. (using RMSE)

I have calculated the Root Mean Squared Error in this section and I've compared against the model I've found interesting. I choose bmod4 because it had the lowest rmse then the others.

```
## I will then use mod,mod2,bmod4 and compare each rmse

## import the caret library..

library(caret)

predictions_1 <- predict(mod1,Ttest)
head(predictions_1)
```

```
##        1        2        3        4        5        6
## 60.12130 74.53094 66.89645 66.19797 69.58318 86.86470
```

```
rmse <- RMSE(predictions_1,Ttest$TARGET_WINS)
rmse
```

```
## [1] 12.4036
```

```
## create the next predictions with mod4

predictions_2 <- predict(mod2,Ttest)
head(predictions_2)
```

```
##        1        2        3        4        5        6
## 58.22799 74.64703 66.90706 66.18559 69.38404 87.02146
```

```
rmse2 <- RMSE(predictions_2,Ttest$TARGET_WINS)
rmse2
```

```
## [1] 12.39892
```

```
## make sure to inverse the box-cox transformation
predictions_3 <- predict(bmod4,Ttest)

## make sure to inverse the box-cox transformation
inv_box_pred <- predictions_3 ^(1/1.3536)
rmse3 <- RMSE(inv_box_pred,Ttest$TARGET_WINS)
head(inv_box_pred)
```

```
##        1        2        3        4        5        6
## 61.21916 75.47602 67.34057 64.70292 71.21794 87.85228
```

```
rmse3
```

```
## [1] 12.33834
```

```
predictions_4 <- predict(bmod5,Ttest)

## make sure to inverse the box-cox transformation
inv_box_pred2 <- predictions_4 ^(1/1.3536)
rmse4 <- RMSE(inv_box_pred2,Ttest$TARGET_WINS)
head(inv_box_pred)
```

```
##        1        2        3        4        5        6
## 61.21916 75.47602 67.34057 64.70292 71.21794 87.85228
```

```
rmse4
```

```
## [1] 12.32826
```

Cleaning The testing dataset

I went to clean the testing dataset in a manner smiliar to the way I have cleaned the training dataset in which I deleted the empty columns and imputate some others and omitted the rest.

```r
## Will predict values with mod4,mod5,and mod6..
Test <- read.csv("https://raw.githubusercontent.com/AldataSci/Baseball-Data/main/moneyball-evaluation-da

## before I do that I have to clean the test data for the linear regression model.. I will clean it in 

str(Test)
```

```
## 'data.frame':    259 obs. of  16 variables:
##  $ INDEX           : int  9 10 14 47 60 63 74 83 98 120 ...
##  $ TEAM_BATTING_H  : int  1209 1221 1395 1539 1445 1431 1430 1385 1259 1397 ...
##  $ TEAM_BATTING_2B : int  170 151 183 309 203 236 219 158 177 212 ...
##  $ TEAM_BATTING_3B : int  33 29 29 29 68 53 55 42 78 42 ...
##  $ TEAM_BATTING_HR : int  83 88 93 159 5 10 37 33 23 58 ...
##  $ TEAM_BATTING_BB : int  447 516 509 486 95 215 568 356 466 452 ...
##  $ TEAM_BATTING_SO : int  1080 929 816 914 416 377 527 609 689 584 ...
##  $ TEAM_BASERUN_SB : int  62 54 59 148 NA NA 365 185 150 52 ...
##  $ TEAM_BASERUN_CS : int  50 39 47 57 NA NA NA NA NA NA ...
##  $ TEAM_BATTING_HBP: int  NA NA NA 42 NA NA NA NA NA NA ...
##  $ TEAM_PITCHING_H : int  1209 1221 1395 1539 3902 2793 1544 1626 1342 1489 ...
##  $ TEAM_PITCHING_HR: int  83 88 93 159 14 20 40 39 25 62 ...
##  $ TEAM_PITCHING_BB: int  447 516 509 486 257 420 613 418 497 482 ...
##  $ TEAM_PITCHING_SO: int  1080 929 816 914 1123 736 569 715 734 622 ...
##  $ TEAM_FIELDING_E : int  140 135 156 124 616 572 490 328 226 184 ...
##  $ TEAM_FIELDING_DP: int  156 164 153 154 130 105 NA 104 132 145 ...
```

```r
## remove the HBP column again and imputate the
sapply(Test,function(x) sum(is.na(x)))
```

```
##            INDEX   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##                0                0                0                0
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##                0                0               18               13
##  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H TEAM_PITCHING_HR
##               87              240                0                0
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##                0               18                0               31
```

```r
## remove hbp and Cs
Test <- Test %>%
  dplyr::select(-c(TEAM_BATTING_HBP,TEAM_BASERUN_CS))
```

```r
sapply(Test,function(x) sum(is.na(x)))
```

```
##            INDEX   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##                0                0                0                0
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##                0                0               18               13
##  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
##                0                0                0               18
##  TEAM_FIELDING_E TEAM_FIELDING_DP
##                0               31
```

```r
## now we imputate..

library(mice)
mice_imputed3 <- data.frame(
original = Test$TEAM_FIELDING_DP,
imp_pmm = complete(mice(Test,method ="pmm"))$TEAM_FIELDING_DP,
imp_cart = complete(mice(Test,method ="cart"))$TEAM_FIELDING_DP,
imp_lasso = complete(mice(Test,method ="lasso.norm"))$TEAM_FIELDING_DP
)
```

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

```
##   3   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
```

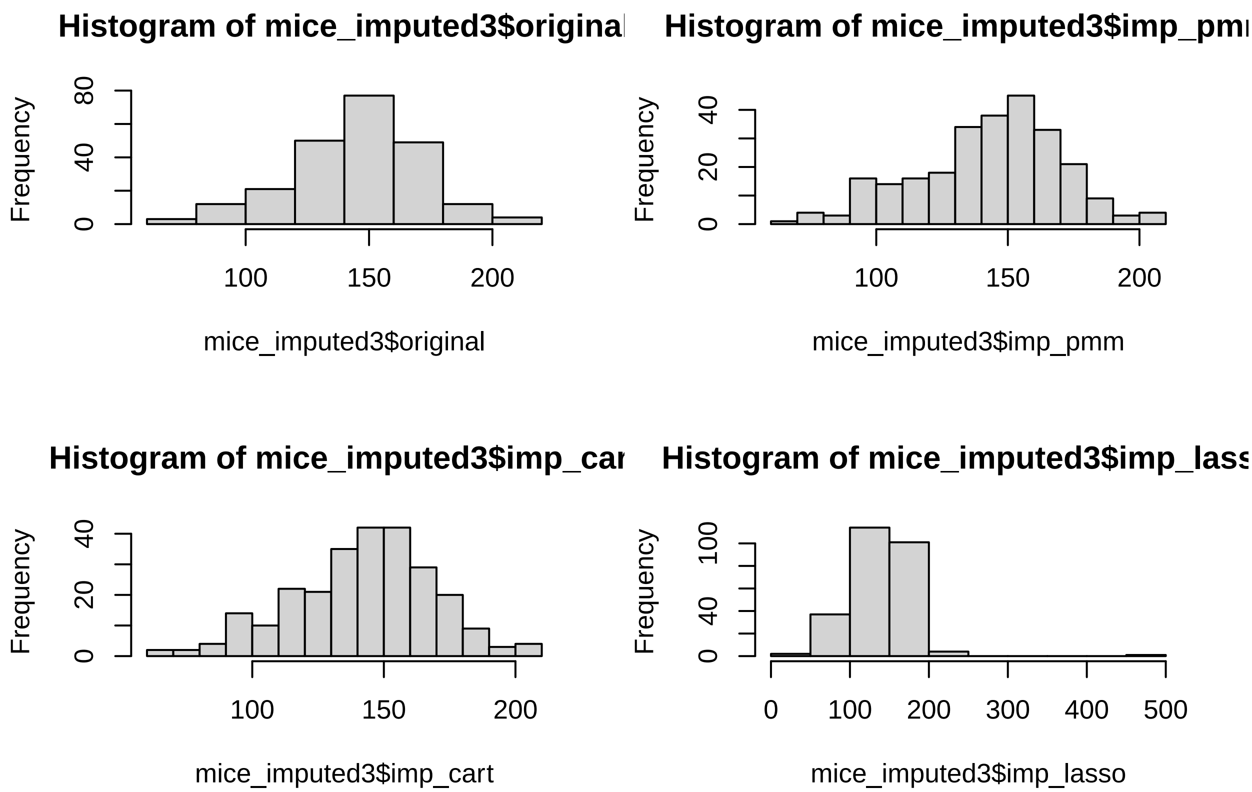
```
## Warning: Number of logged events: 13
```

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

```
head(mice_imputed3)
```

```
##   original imp_pmm imp_cart imp_lasso
## 1      156     156      156       156
## 2      164     164      164       164
## 3      153     153      153       153
## 4      154     154      154       154
## 5      130     130      130       130
## 6      105     105      105       105
```

```
par(mfrow=c(2,2))
hist(mice_imputed3$original)
hist(mice_imputed3$imp_pmm)
hist(mice_imputed3$imp_cart)
hist(mice_imputed3$imp_lasso)
```




```
## Since the imp_cart looks smiliar to the original distribution I will use that then..

Test$TEAM_FIELDING_DP <- mice_imputed3$imp_cart
```

```
## now we imputate the next column.. which is BASERUN_SB

mice_imputed4 <- data.frame(
original = Test$TEAM_BASERUN_SB,
imp_pmm = complete(mice(Test,method ="pmm"))$TEAM_BASERUN_SB,
imp_cart = complete(mice(Test,method ="cart"))$TEAM_BASERUN_SB,
imp_lasso = complete(mice(Test,method ="lasso.norm"))$TEAM_BASERUN_SB
)
```

```
##
##  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
```

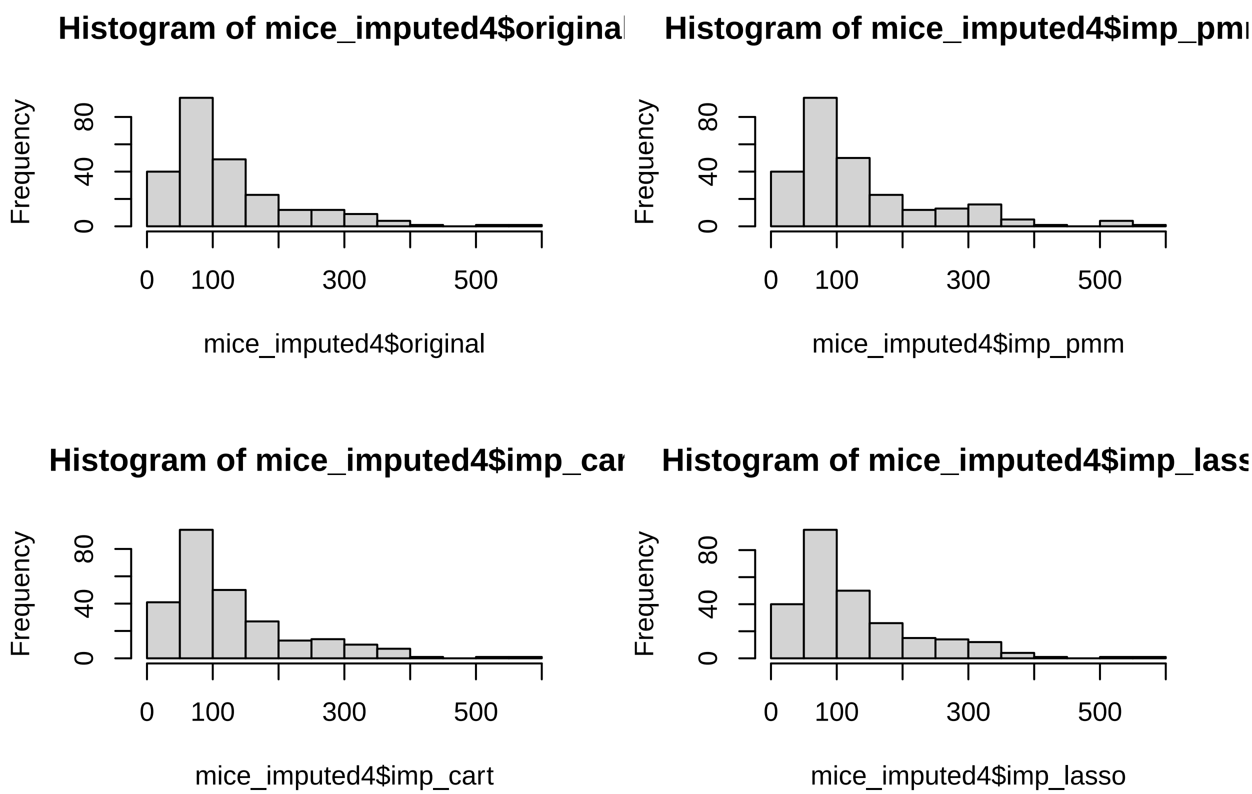
```
##    1   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##    1   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##    2   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##    2   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO
##    2   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  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
##
##  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
##
##  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
```

```r
head(mice_imputed4)
```

```
##   original imp_pmm imp_cart imp_lasso
## 1       62      62       62   62.0000
## 2       54      54       54   54.0000
## 3       59      59       59   59.0000
## 4      148     148      148  148.0000
## 5       NA     319      119  150.3924
## 6       NA     307      298  240.3488
```

```r
par(mfrow=c(2,2))
hist(mice_imputed4$original)
hist(mice_imputed4$imp_pmm)
hist(mice_imputed4$imp_cart)
hist(mice_imputed4$imp_lasso)
```




```
## I will use imp_pmm again and replace those columns with those imputed values..
Test$TEAM_BASERUN_SB <- mice_imputed4$imp_pmm
```

```
sapply(Test,function(x) sum(is.na(x)))
```

```
##            INDEX   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##                0                0                0                0
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##                0                0               18                0
##  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
##                0                0                0               18
##  TEAM_FIELDING_E TEAM_FIELDING_DP
##                0                0
```

```
## Then I will remove some of the columns since I had imputated most of the columns..

Testt <- na.omit(Test)


sapply(Testt,function(x) sum(is.na(Testt)))
```

```
##            INDEX   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##                0                0                0                0
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
```

```
##                0                 0                 0                 0
##   TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
##                0                 0                 0                 0
##   TEAM_FIELDING_E TEAM_FIELDING_DP
##                0                 0
```

Creating predictions with the cleaned Test Data..

Finally, I used the model and I created predictions with the test dataset.

```r
set.seed(123)
pred <- predict(bmod5,newdata=Testt)


## I have to revert the transformation back..
actual_predictions <- pred ^ (1/1.3536)

actual_predictions
```

```
##         1         2         3         4         5         6         7         8
##  60.80944  63.75216  74.11000  88.65487  72.16059  77.42823  86.28548  76.85627
##         9        10        11        12        13        14        15        16
##  68.96860  73.85995  69.89725  82.39285  81.17726  83.84187  85.98562  77.94292
##        17        18        20        21        22        23        24        25
##  74.36270  79.40131  91.89593  82.25320  85.14062  79.80909  73.20556  83.43505
##        26        27        28        29        30        31        32        33
##  89.20818  62.67432  75.54723  84.58322  76.48267  91.08926  85.75685  82.27941
##        34        35        36        37        38        39        40        41
##  83.96554  78.84116  87.33181  76.11699  89.10459  85.04215  90.73503  85.40392
##        42        43        44        45        46        47        48        49
##  91.36515  20.75504 102.83892  91.40985  93.73382  98.20219  76.97309  68.59335
##        50        51        52        53        54        55        56        57
##  79.95505  77.65632  86.73014  75.75885  73.01913  75.68632  78.41573  92.25520
##        58        61        62        63        64        65        66        67
##  76.29594  87.55781  72.85582  88.72138  87.21918  85.60452 103.46486  73.45176
##        68        70        71        72        73        74        75        76
##  78.93152  86.72107  82.31174  70.89558  78.00517  89.43964  80.58307  83.36438
##        77        78        81        82        83        84        85        86
##  81.85810  84.31052  87.20861  87.54967  96.48819  75.03624  84.07945  82.34493
##        87        88        89        90        91        92        93        97
##  83.95957  83.44861  90.24485  91.71583  81.93266  85.94270  74.83325  87.20153
##        98        99       100       101       102       103       104       105
##  99.29148  85.34539  86.07364  79.04429  75.55087  83.97628  83.91146  79.12417
##       106       107       108       109       110       111       112       113
##  77.35001  63.42791  78.19793  87.27129  57.34491  85.49080  87.77110  93.70522
##       114       115       116       117       118       119       120       121
##  91.47305  81.08838  79.36064  85.63479  82.13145  74.58369  81.07720  94.19316
##       125       126       127       128       129       130       131       132
##  67.21962  87.15531  89.02325  76.17558  92.72980  90.88547  86.05337  81.55979
##       133       134       135       136       137       138       139       140
```

```
## 81.61826  83.93668  86.78713  77.18046  73.91630  77.91914  89.50153  81.98250
##       141       143       144       145       146       147       148       149
## 64.35104  90.01582  72.61803  72.02380  71.71261  77.60975  79.67355  79.20551
##       150       151       152       153       154       155       156       157
## 83.81609  82.55143  81.21633  42.45745  68.85590  76.45661  70.54302  90.39955
##       158       159       161       162       163       164       165       166
## 81.14781  89.69906 100.25466 105.06338  93.01537 101.88795  96.43841  88.36029
##       167       168       169       170       172       173       174       175
## 80.55496  82.55205  74.14865  82.39395  88.43825  80.80561  93.87117  84.15889
##       176       177       178       179       180       181       182       183
## 73.19405  78.64945  70.38943  73.90726  79.53411  90.49726  89.15136  86.64794
##       184       185       186       187       188       189       190       193
## 85.46430  85.78372  96.23285  86.72172  55.16891  69.97461 113.83683  77.23967
##       194       195       196       197       198       199       200       201
## 78.23356  81.17814  69.66306  79.28533  84.01701  79.23928  82.27913  72.92033
##       202       203       204       205       206       207       208       209
## 77.79870  71.48299  90.14026  82.41870  83.29129  77.88275  78.00326  83.24782
##       210       211       212       213       214       215       216       217
## 69.83404 105.88469  94.06061  79.61090  65.24934  67.34264  81.88118  77.04449
##       218       219       220       221       222       223       224       225
## 93.96015  78.00180  78.45314  78.00917  74.99659  82.36323  72.70547  76.43890
##       226       227       228       229       230       232       233       234
## 74.53187  82.17820  79.54310  81.84636  70.80286  91.36059  78.42364  89.34777
##       235       236       237       238       239       240       241       242
## 80.31995  74.71053  82.71944  76.68497  90.06882  71.13006  87.61516  86.30449
##       243       244       245       246       247       248       249       250
## 83.95199  82.17514  61.55618  88.85764  81.37896  85.80634  73.18700  84.72581
##       251       252       253       254       255       256       257       258
## 80.94501  64.82077  90.05452  30.99694  69.30838  77.61814  83.60266  85.99413
##       259
## 78.59148
```

```
## And that is all!! done...
```