Assignment01

Anil Akyildirim, John K. Hancock, John Suh, Emmanuel Hayble-Gomes, Chunjie Nan

2/26/2020

Introduction

In this assignment, we are tasked to explore, analyze and model a major league baseball dataset which contains around 2000 records where each record presents a baseball team from 1871 to 2006. Each observation provides the perforamce of the team for that particular year with all the statistics for the performance of 162 game season. The problem statement for the main objective is that "Can we predict the number of wins for the team with the given attributes of each record?". In order to provide a solution for the problem, our goal is to build a linear regression model on the training data that creates this prediction.

About the Data

The data set are provided in csv format as moneyball-evaluation-data and moneyball-training-data where we will explore, preperate and create our model with the training data and further test the model with the evaluation data. Below is short description of the variables within the datasets.

**INDEX: Identification Variable(Do not use)

**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

Data Exploration

Descriptive Statistics

```r
# load libraries
library(ggplot2)
library(ggcorrplot)
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(statsr)
```

```
## Warning: package 'statsr' was built under R version 3.6.2

## Loading required package: BayesFactor

## Warning: package 'BayesFactor' was built under R version 3.6.2

## Loading required package: coda

## Loading required package: Matrix

## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey
##
## Type BFManual() to open the manual.
## ************
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 3.6.2

## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 3.6.2

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

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

```
library(rcompanion)
```

```
## Warning: package 'rcompanion' was built under R version 3.6.2

##
## Attaching package: 'rcompanion'

## The following object is masked from 'package:psych':
##
##     phi
```

```
library(caret)
```

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

```
library(MASS)
```

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

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

```
library(imputeTS)
```

```
## Warning: package 'imputeTS' was built under R version 3.6.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##
## Attaching package: 'imputeTS'
```

```
## The following object is masked from 'package:zoo':
##
##     na.locf
```

```
library(rsample)
```

```
## Warning: package 'rsample' was built under R version 3.6.2
```

```
library(huxtable)
```

```
## Warning: package 'huxtable' was built under R version 3.6.2
```

```
##
## Attaching package: 'huxtable'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     theme_grey
```

```
library(glmnet)
```

```
## Loaded glmnet 3.0-1

##
## Attaching package: 'glmnet'

## The following object is masked from 'package:imputeTS':
##
##     na.replace
```

```
library(sjPlot)
```

```
## Warning: package 'sjPlot' was built under R version 3.6.2

## Install package "strengejacke" from GitHub (`devtools::install_github("strengejacke/strengejacke")`)

##
## Attaching package: 'sjPlot'

## The following object is masked from 'package:huxtable':
##
##     font_size
```

```
library(modelr)
```

```
##
## Attaching package: 'modelr'

## The following object is masked from 'package:psych':
##
##     heights
```

```
# Load data sets

baseball_eva <- read.csv("https://raw.githubusercontent.com/anilak1978/data621/master/moneyball-evaluati
baseball_train <- read.csv("https://raw.githubusercontent.com/anilak1978/data621/master/moneyball-traini
```

We can start exploring our training data set by looking at basic descriptive statistics.

```
# look at training dataset structure
str(baseball_train)
```

```
## '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 ...
```

We have 2276 observations and 17 variables. All of our variables are integer type as expected.

```
# look at descriptive statistics
metastats <- data.frame(describe(baseball_train))
metastats <- tibble::rownames_to_column(metastats, "STATS")
metastats["pct_missing"] <- round(metastats["n"]/2276, 3)
head(metastats)
```

STATS	vars	n	mean	sd	median	trimmed	mad	min	max	ra
INDEX	1	2.28e+03	1.27e+03	736	1.27e+03	1.27e+03	953	1	2.54e+03	2.53e
TARGET_WINS	2	2.28e+03	80.8	15.8	82	81.3	14.8	0	146	146
TEAM_BATTING_H	3	2.28e+03	1.47e+03	145	1.45e+03	1.46e+03	114	891	2.55e+03	1.66e
TEAM_BATTING_2B	4	2.28e+03	241	46.8	238	240	47.4	69	458	389
TEAM_BATTING_3B	5	2.28e+03	55.2	27.9	47	52.2	23.7	0	223	223
TEAM_BATTING_HR	6	2.28e+03	99.6	60.5	102	97.4	78.6	0	264	264

With the descriptive statistics, we are able to see mean, standard deviation, median, min, max values and percentage of each missing value of each variable. For example, when we look at TEAM_BATTING_H, we see that average 1469 Base hits by batters, with standard deviation of 144, median of 1454 with maximum base hits of 2554.

```
# Look for missing values
colSums(is.na(baseball_train))
```

```
##            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
```

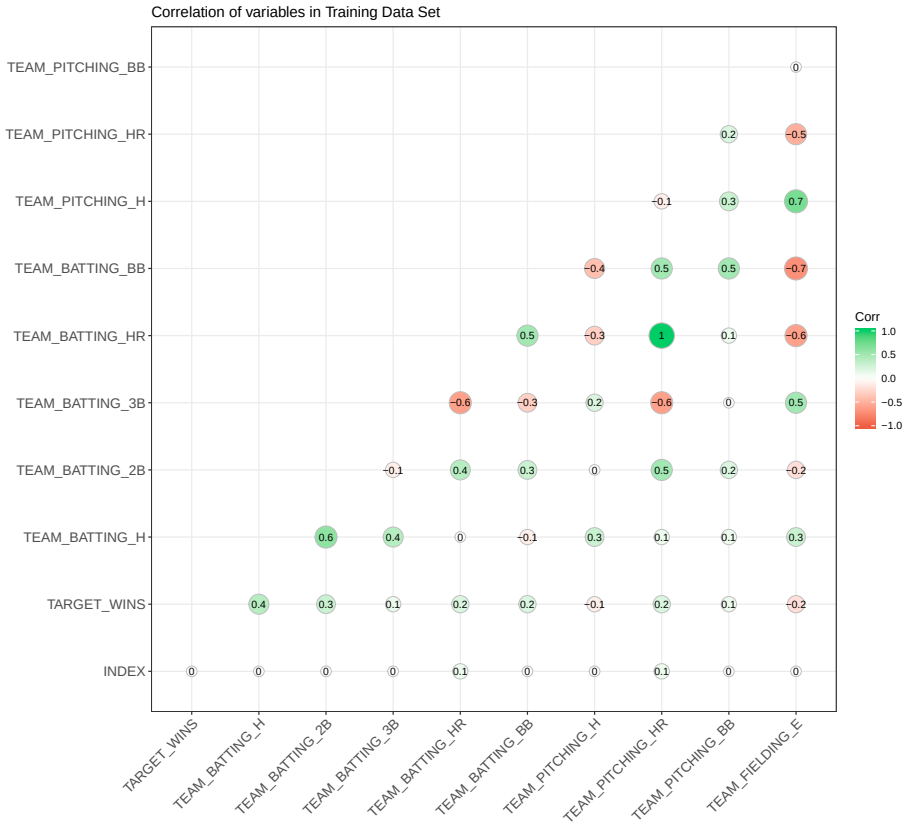
```
# Percentage of missing values
missing_values <- metastats %>%
  filter(pct_missing < 1) %>%
  dplyr::select(STATS, pct_missing) %>%
  arrange(pct_missing)
missing_values
```

STATS	pct_missing
TEAM_BAT'	0.084
TEAM_BASERUN_CS	0.661
TEAM_FIEL	0.874
TEAM_BASERUN_SB	0.942
TEAM_BAT'	0.955
TEAM_PITCHING_SO	0.955

When we look at the missing values within the training data set, we see that proportionaly against the total observations, TEAM_BATTING_HBP and TEAM_BESARUN_CS variables have the most missing values. We will be handling these missing values in our Data Preperation section.

Correlation and Distribution

```
# Look at correlation between variables

corr <- round(cor(baseball_train), 1)

ggcorrplot(corr,
           type="lower",
           lab=TRUE,
           lab_size=3,
           method="circle",
           colors=c("tomato2", "white", "springgreen3"),
           title="Correlation of variables in Training Data Set",
           ggtheme=theme_bw)
```



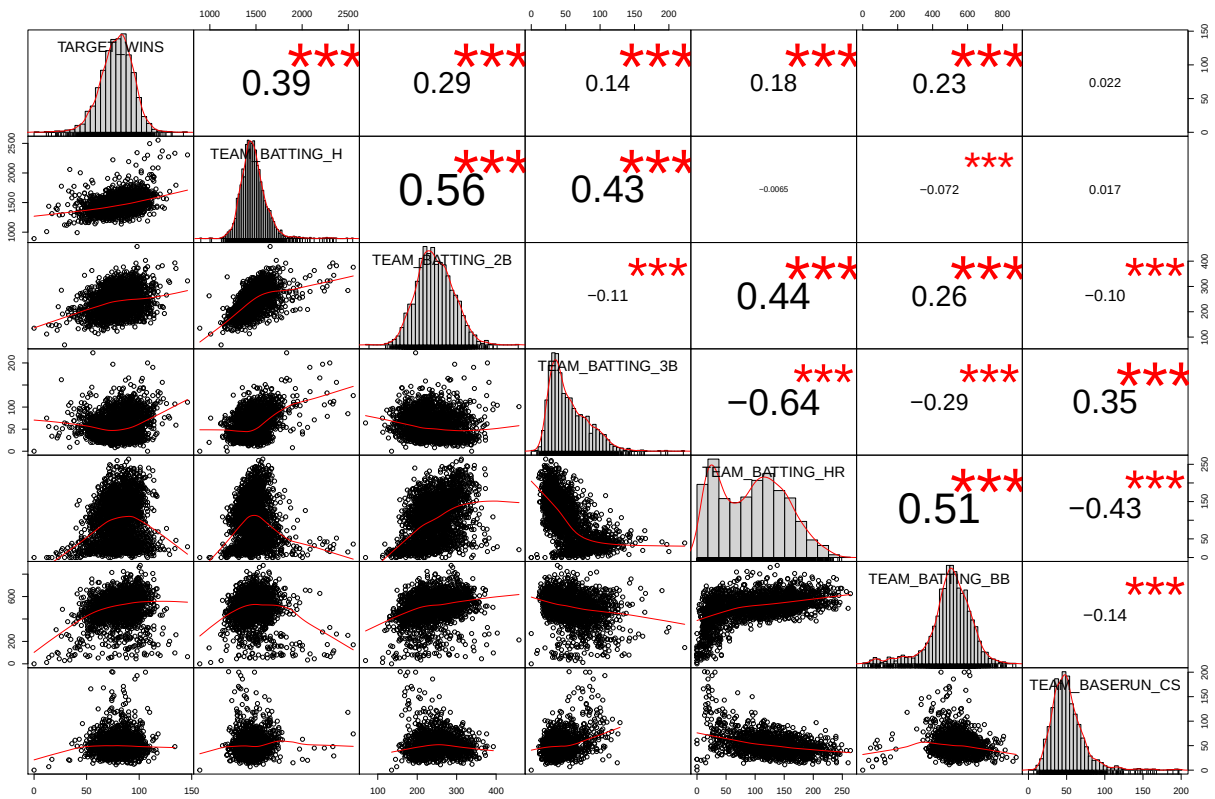
Team_Batting_H and Team_Batting_2B have the strongest positive correlation with Target_Wins. We also see that, there is a strong correlation between Team_Batting_H and Team_Batting_2B, Team_Pitching_B and TEAM_FIELDING_E. We will consider these findings on model creation as collinearity might complicate model estimation and we want to have explanotry variables to be independent from each other. We will try to avoid adding explanotry variables that are correlated to each other.

Let's look at the correlations and distribution of the variables in more detail.

```
# Look at correlation from batting, baserunning, pitching and fielding perspective
Batting_df <- baseball_train[c(2:7, 10)]
BaseRunning_df <- baseball_train[c(8:9)]
Pitching_df <- baseball_train[c(11:14)]
Fielding_df <- baseball_train[c(15:16)]
```

```
# Batting Correlations
chart.Correlation(Batting_df, histogram=TRUE, pch=19)
```

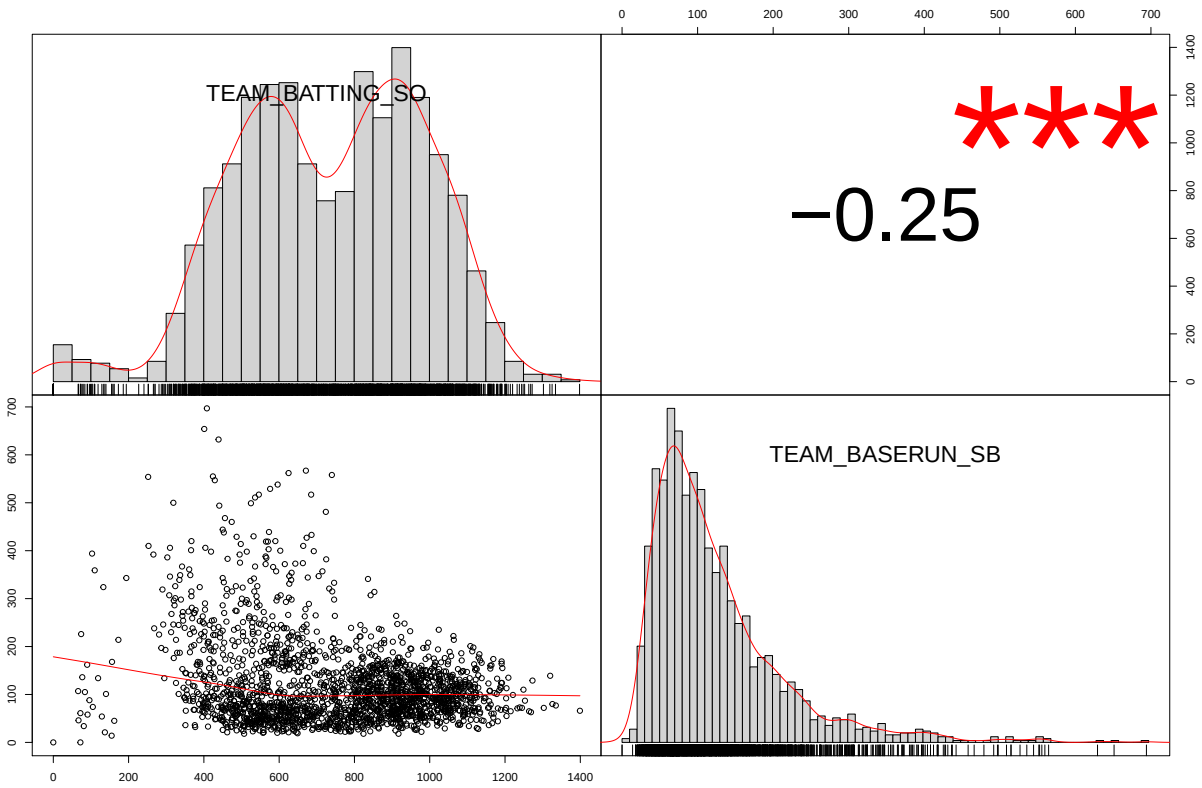
Batting



We can see that our response variable TARGET_WINS, TEAM_BATTING_H, TEAM_BATTING_2B, TEAM_BATTING_BB and TEAM_BASERUN_CS are normaly distributed. TEAM_BATTING_HR on the other hand is bimodal.

```
# baserunning Correlation

chart.Correlation(BaseRunning_df, histogram=TRUE, pch=19)
```

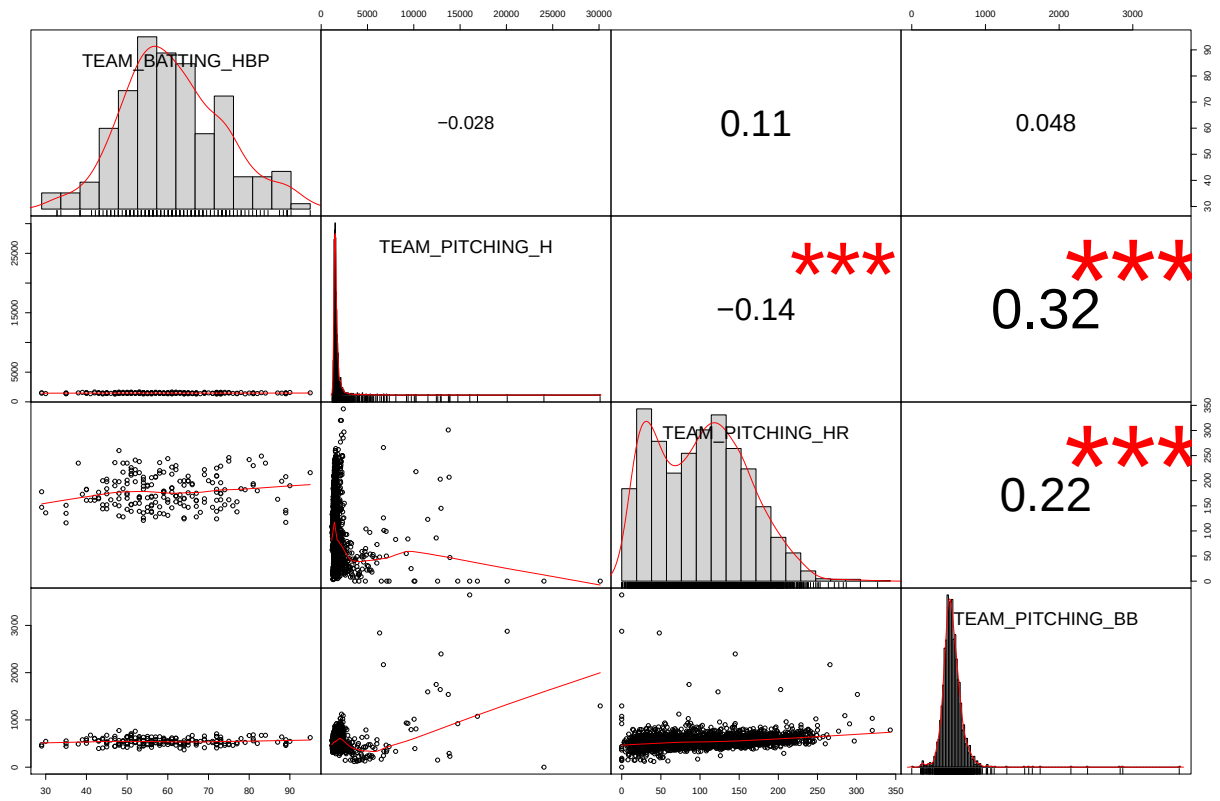
Baserunning



TEAM_BASERUN_SB is right skewed and TEAM_BATTING_SO is bimodal.

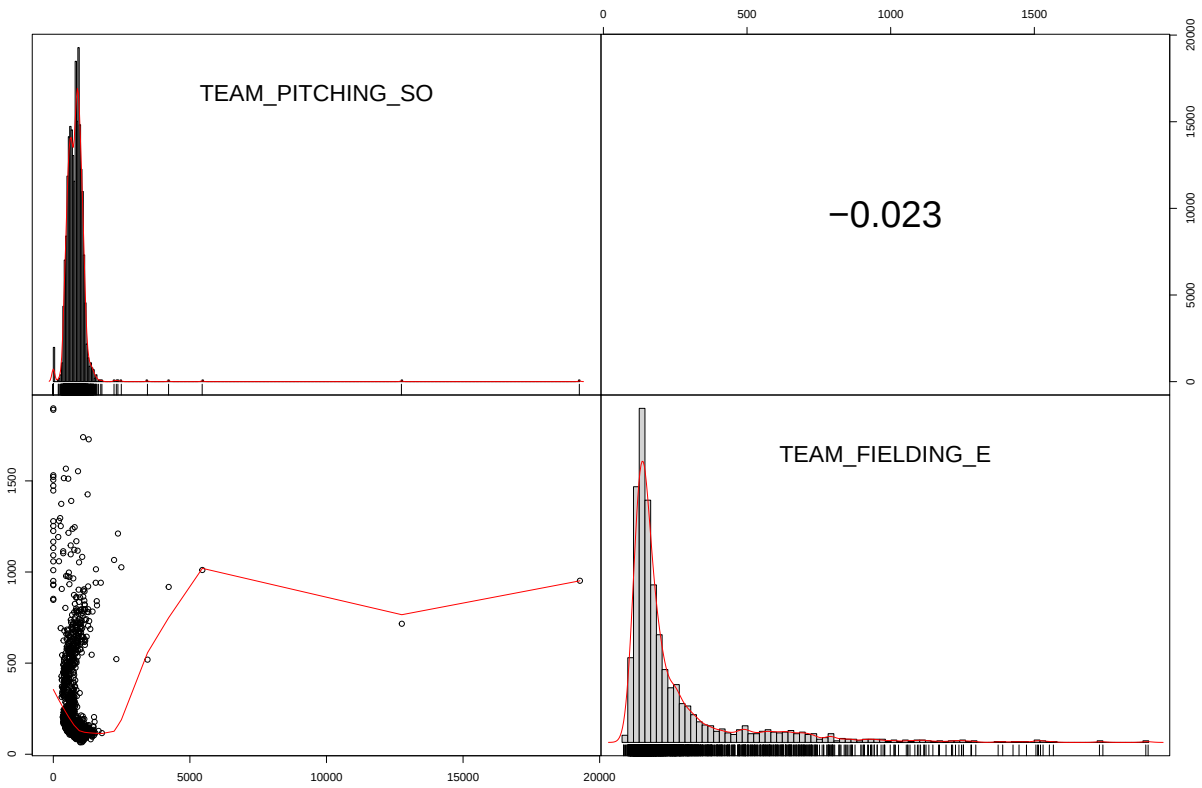
```
#pitching correlations
chart.Correlation(Pitching_df, histogram=TRUE, pch=19)
```

Pitching



TEAM_BATTING_HBP seems to be normally distributed however we shouldnt forget that we have a lot of missing values in this variable.

```
# fielding correlations
chart.Correlation(Fielding_df, histogram=TRUE, pch=19)
```



Let's also look at the outliers and skewness for each varibale.

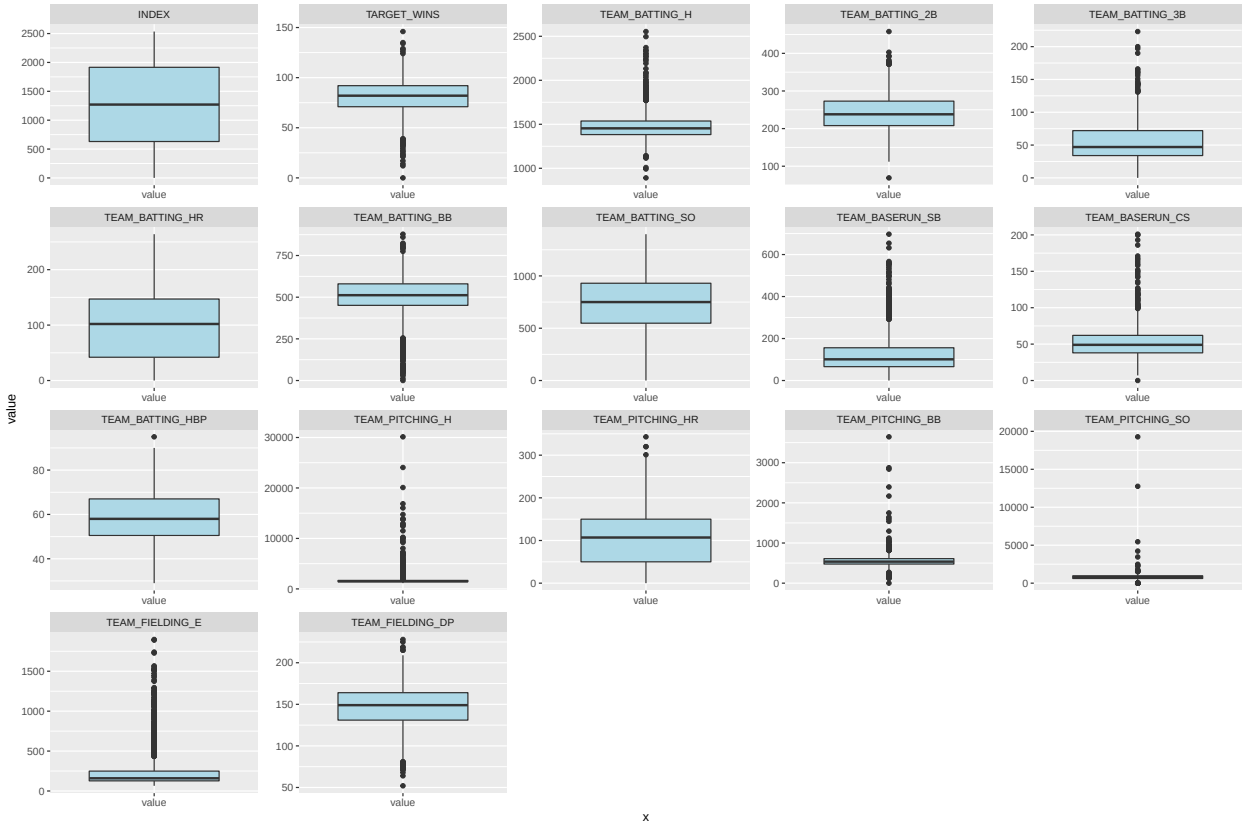
Outliers and Skewness

```
par(mfrow=c(3,3))
datasub_1 <- melt(baseball_train)
```

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

```
suppressWarnings(ggplot(datasub_1, aes(x= "value", y=value)) +
                   geom_boxplot(fill='lightblue') + facet_wrap(~variable, scales = 'free') )
```

```
## Warning: Removed 3478 rows containing non-finite values (stat_boxplot).
```



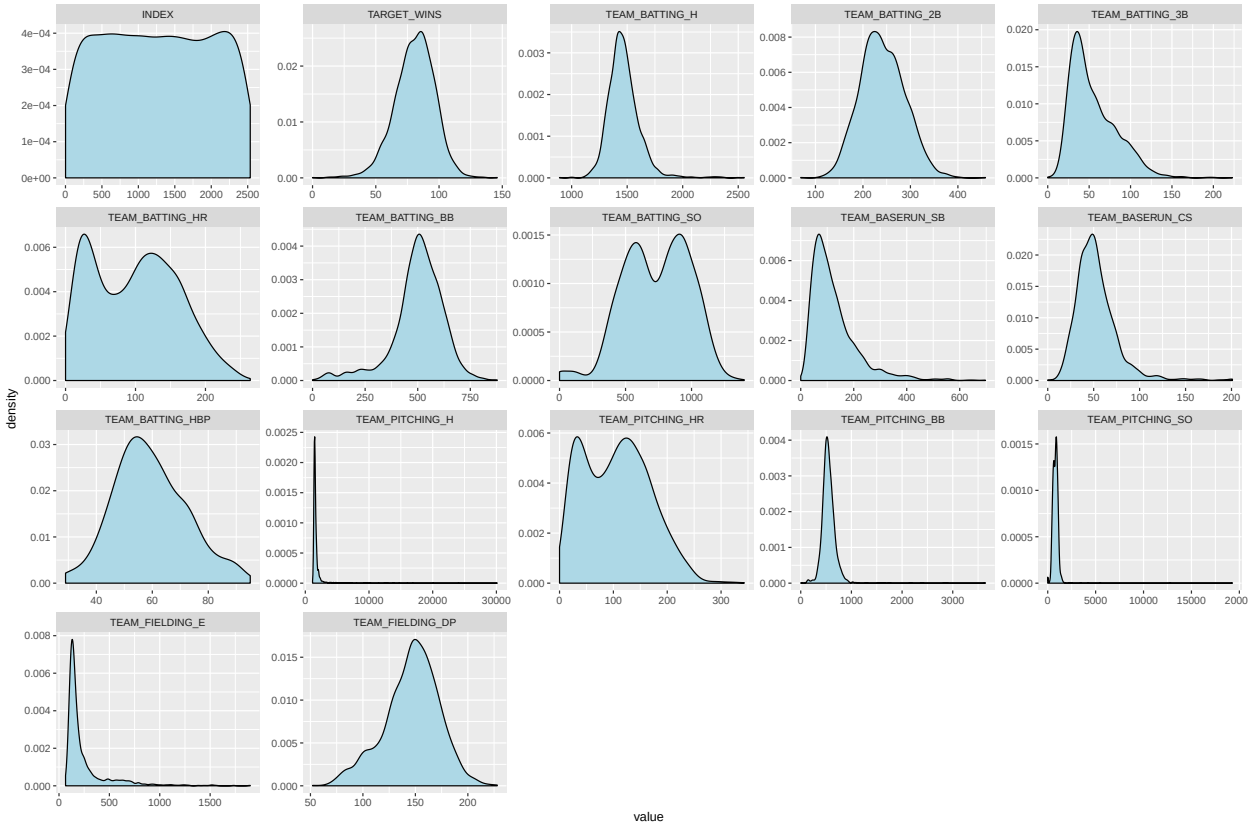
Based on the boxplot we created, TEAM_FIELDING_DP, TEAM_PITCHING_HR, TEAM_BATTING_HR and TEAM_BATTING_SO seem to have the least amount of outliers.

```
par(mfrow = c(3, 3))
datasub = melt(baseball_train)
```

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

```
suppressWarnings(ggplot(datasub, aes(x= value)) +
                   geom_density(fill='lightblue') + facet_wrap(~variable, scales = 'free') )
```

```
## Warning: Removed 3478 rows containing non-finite values (stat_density).
```




```
metastats %>%
  filter(skew > 1) %>%
  dplyr::select(STATS, skew) %>%
  arrange(desc(skew))
```

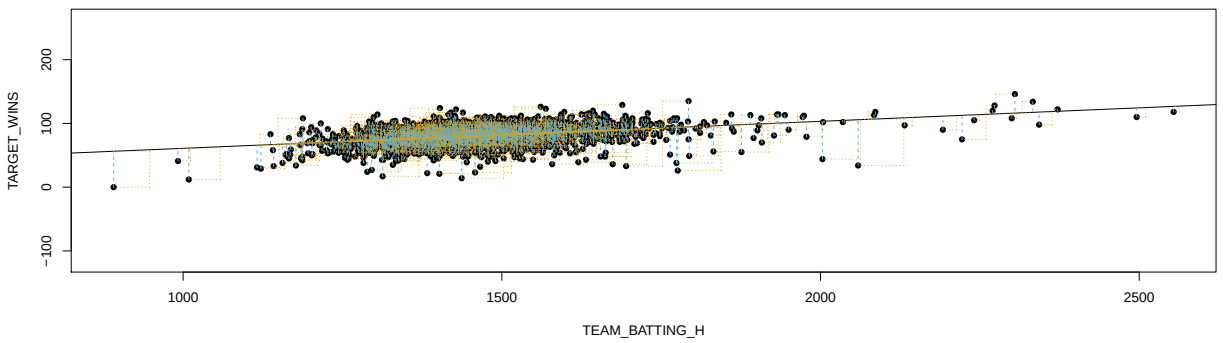
STATS	skew
TEAM_PITCHING	22.2
TEAM_PITCHING_H	10.3
TEAM_PITCHING	6.74
TEAM_FIELDING_E	2.99
TEAM_BASERUN_	1.98
TEAM_BASERUN_SB	1.97
TEAM_BATTING_	1.57
TEAM_BATTING_3B	1.11

We can see that the most skewed variable is TEAM_PITCHING_SO. We will correct the skewed variables in our data preperation section.

When we are creating a linear regression model, we are looking for the fitting line with the least sum of squares, that has the small residuals with minimized squared residuals. From our correlation analysis, we can see that the explatory variable that has the strongest correlation with TARGET_WINS is TEAM_BATTING_H. Let's look at a simple model example to further expand our explaroty analysis.

Simple Model Example

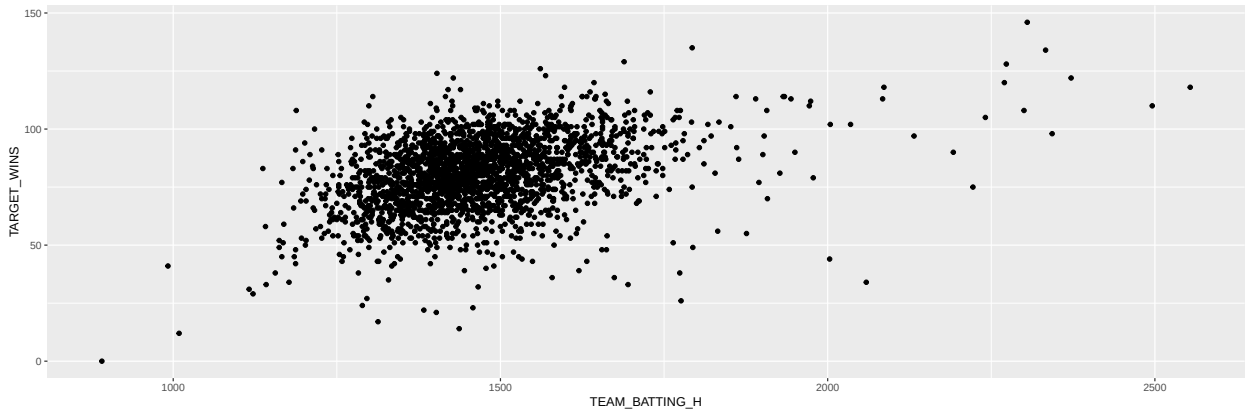
```
# line that follows the best assocation between two variables

plot_ss(x = TEAM_BATTING_H, y = TARGET_WINS, data=baseball_train, showSquares = TRUE, leastSquares = TRU
```




```
##

## Call:
## lm(formula = y ~ x, data = data)
##
## Coefficients:
## (Intercept)            x
##    18.56233      0.04235
##
## Sum of Squares:  479178.4
```

When we are exploring to build a linear regression, one of the first thing we do is to create a scatter plot of the response and explanatory variable.

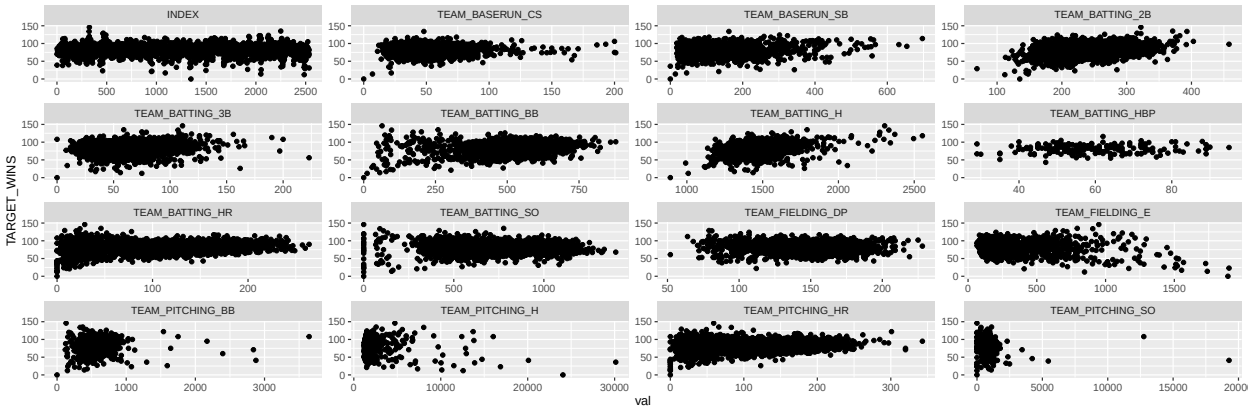
```
# scatter plot between TEAM_BATTING_H and TARGET_WINS

ggplot(baseball_train, aes(x=TEAM_BATTING_H, y=TARGET_WINS))+
  geom_point()
```



One of the conditions for least square lines or linear regression are Linearity. From the scatter plot between TEAM_BATTING_H and TARGET_WINS, we can see this condition is met. We can also create a scatterplot that shows the data points between TARGET_WINS and each variable.

```
baseball_train %>%
  gather(var, val, -TARGET_WINS) %>%
  ggplot(., aes(val, TARGET_WINS))+
  geom_point()+
  facet_wrap(~var, scales="free", ncol=4)
```

```
## Warning: Removed 3478 rows containing missing values (geom_point).
```



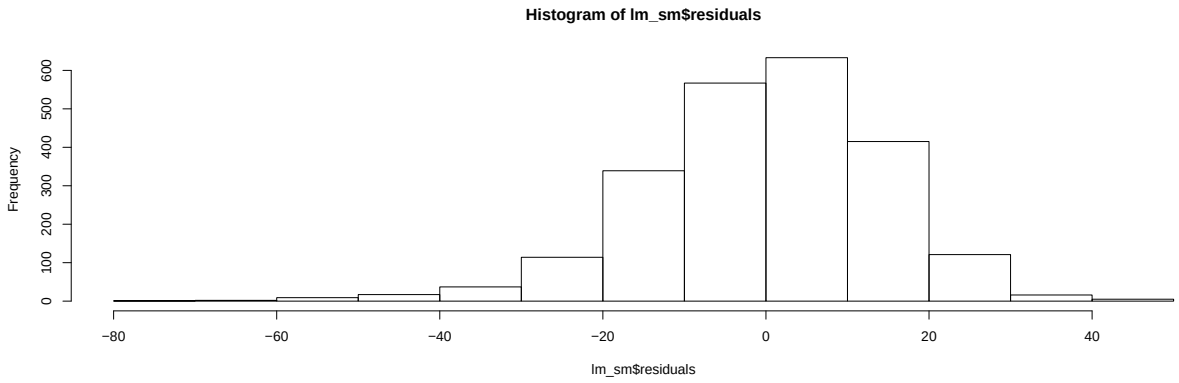
As we displayed earlier, hits walks and home runs have the strongest correlations with TARGET_WINS and also meets the linearity condition.

```
# create a simple example model
lm_sm <- lm(baseball_train$TARGET_WINS ~ baseball_train$TEAM_BATTING_H)
summary(lm_sm)
```

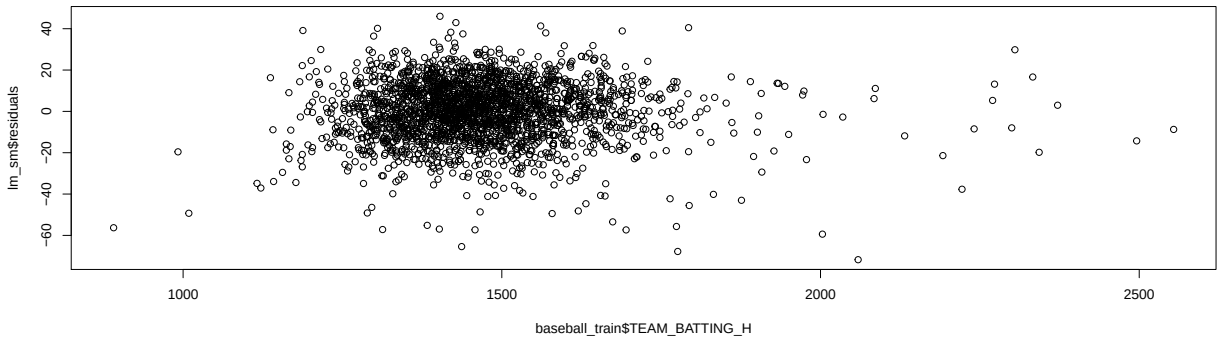
```
##
## Call:
## lm(formula = baseball_train$TARGET_WINS ~ baseball_train$TEAM_BATTING_H)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -71.768  -8.757   0.856   9.762  46.016
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  18.562326   3.107523   5.973 2.69e-09 ***
## baseball_train$TEAM_BATTING_H  0.042353   0.002105  20.122  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.52 on 2274 degrees of freedom
## Multiple R-squared:  0.1511, Adjusted R-squared:  0.1508
## F-statistic: 404.9 on 1 and 2274 DF,  p-value: < 2.2e-16
```

TARET_BATTING_H has the strongest correlation with TARGET_WINS response variable, however when we create a simple model just using TARGET_BATTING_H, we can only explain 15% of the variablity. (Adjusted R-squared: 0.1508). The remainder of the varibility can be explained with selected other variables within the training dataset.

```
#histogram of residuals for the simple model
hist(lm_sm$residuals)
```




```
# check for constant variability (honoscedasticity)

plot(lm_sm$residuals ~ baseball_train$TEAM_BATTING_H)
```



We do see that the residuals are distributed normally and variability around the regression line is roughly constant.

Based on our explatory analysis, we were able to see the correlation level between the possible explanatory variables and repsonse variable TARGET_WINS. Some of the variables such as TARGET_BATTING_H has somewhat strong positive correlation, however some of the variables such as TEAM_PITCHING_BB has weak positive relationship with TARGET_WINS. We also found out, hit by the pitcher(TEAM_BATTING_HBP) and caught stealing (TEAM_BASERUN_CS) variables are missing majority of the values. Skewness and distribution analysis gave us the insights that we have some variables that are right-tailed. Considering all of these insights, we will handle missing values, correct skewness and outliers and select our explaratory variables based on correlation in order to create our regression model.

Data Preparation

Objective

In this section, we will prepare the dataset for linear regression modeling. We accomplish this by handling missing values and outliers and by tranforming the data into more normal distributions. This section covers:

Identify and Handle Missing Data Correct Outliers *Adjust Skewed value - Box Cox Transformation

First, we will start by copying the dataset into a new variable, baseball_train_01, and we will remove the Index variable from the new dataset as well. We will now have 16 variables.

```
baseball_train_01 <- baseball_train

baseball_train_01 <-subset(baseball_train_01, select = -c(INDEX))
```

Identify and Handle Missing Data

Removal of Sparsely Populated Variables - MCAR In the Data Exploration section, we identified these variables as having missing data values.The table below lists the variables with missing data. The variable, TEAM_BATTING_HBP, is sparsely populated. Since this data is Missing Completely at Random (MCAR) and is not related to any other variable, it is safe to completely remove the variable from the dataset.

```
missing_values
```

STATS	pct_missing
TEAM_BAT′	0.084
TEAM_BASERUN_CS	0.661
TEAM_FIEL	0.874
TEAM_BASERUN_SB	0.942
TEAM_BAT′	0.955
TEAM_PITCHING_SO	0.955

```
baseball_train_01 <-subset(baseball_train_01, select = -c(TEAM_BATTING_HBP))
```

There are now 15 variables.

```
dim(baseball_train_01)
```

```
## [1] 2276   15
```

Imputation of Missing Values For the remaining variables with missing values, we will impute the mean of the variable. The function, "na_mean" updates all missing values with the mean of the variable.

```
baseball_train_01 <- na_mean(baseball_train_01, option = "mean")
```

Re-running the metastats dataframe on the new baseball_train_01 dataset shows that there are no missing values.

```
# look at descriptive statistics
metastats <- data.frame(describe(baseball_train_01))
metastats <- tibble::rownames_to_column(metastats, "STATS")
metastats["pct_missing"] <- round(metastats["n"]/2276, 3)
```

```
# Percentage of missing values
missing_values2 <- metastats %>%
  filter(pct_missing < 1) %>%
  dplyr::select(STATS, pct_missing) %>%
  arrange(pct_missing)
missing_values2
```

```
## Warning in max(nchar(as.character(col), type = "width")): no non-missing
## arguments to max; returning -Inf

## Warning in max(nchar(as.character(col), type = "width")): no non-missing
## arguments to max; returning -Inf
```

STATS	pct_missing

Correct Outliers

In this section, we created two functions that can identify outliers. The funcion, Identify_Outlier, uses the Turkey method, where outliers are identified by being below Q1-1.5*IQR and above Q3+1.5*IQR. The second function, tag_outlier, returns a binary list of values, "Acceptable" or "Outlier" that will be added to the dataframe.

```
Identify_Outlier <- function(value){

    interquartile_range = IQR(sort(value),na.rm = TRUE)
    q1 = matrix(c(quantile(sort(value),na.rm = TRUE)))[2]
    q3 = matrix(c(quantile(sort(value),na.rm = TRUE)))[4]
    lower = q1-(1.5*interquartile_range)
    upper = q3+(1.5*interquartile_range)

    bound <- c(lower, upper)

    return (bound)
}
```

```
tag_outlier <- function(value) {

   boundaries <- Identify_Outlier(value)
   tags <- c()
   counter = 1
```

```
    for (i in as.numeric(value))
    {

        if (i >= boundaries[1] & i <= boundaries[2]){
          tags[counter] <- "Acceptable"
        } else{
          tags[counter] <- "Outlier"
        }

      counter = counter +1
    }

   return (tags)
}
```

As seen in the box plots from the previous section, "TEAM_BASERUN_SB", "TEAM_BASERUN_CS", "TEAM_PITCHING_H", "TEAM_PITCHING_BB", "TEAM_PITCHING_SO", and "TEAM_FIELDING_E" all have a high number of outliers. We will use the two functions above to tag those rows with extreme outliers.

```
tags<- tag_outlier(baseball_train_01$TEAM_BASERUN_SB)
baseball_train_01$TEAM_BASERUN_SB_Outlier <- tags

tags<- tag_outlier(baseball_train_01$TEAM_BASERUN_CS)
baseball_train_01$TEAM_BASERUN_CS_Outlier <- tags

tags<- tag_outlier(baseball_train_01$TEAM_PITCHING_H)
baseball_train_01$TEAM_PITCHING_H_Outlier <- tags

tags<- tag_outlier(baseball_train_01$TEAM_PITCHING_BB)
baseball_train_01$TEAM_PITCHING_BB_Outlier <- tags

tags<- tag_outlier(baseball_train_01$TEAM_PITCHING_SO)
baseball_train_01$TEAM_PITCHING_SO_Outlier <- tags

tags<- tag_outlier(baseball_train_01$TEAM_FIELDING_E)
baseball_train_01$TEAM_FIELDING_E_Outlier <- tags
```

Below, we filtered out all of the outliers and created a new dataframe, baseball_train_02

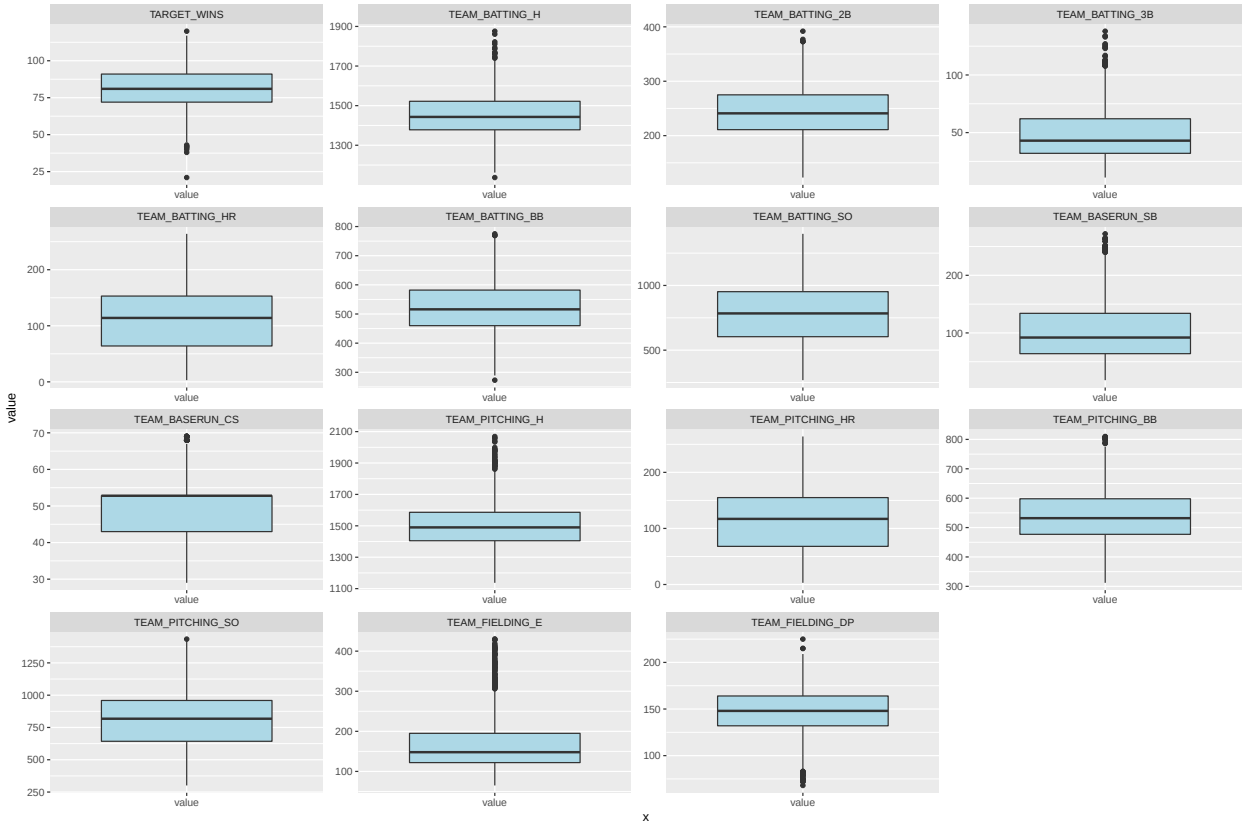
```
baseball_train_02 <- baseball_train_01 %>%
                filter(
                        TEAM_BASERUN_SB_Outlier != "Outlier" &
                        TEAM_BASERUN_CS_Outlier != "Outlier" &
                        TEAM_PITCHING_H_Outlier != "Outlier" &
                        TEAM_PITCHING_BB_Outlier != "Outlier" &
                        TEAM_PITCHING_SO_Outlier != "Outlier" &
                        TEAM_FIELDING_E_Outlier != "Outlier"
                )
```

Re-running the boxplots show data that has a better normal distribution except for the variable, TEAM_FIELDING_E which is still skewed. We will handle this next.

```
par(mfrow=c(3,3))
datasub_1 <- melt(baseball_train_02)
```

```
## Using TEAM_BASERUN_SB_Outlier, TEAM_BASERUN_CS_Outlier, TEAM_PITCHING_H_Outlier, TEAM_PITCHING_BB_Ou
```

```
suppressWarnings(ggplot(datasub_1, aes(x= "value", y=value)) +
                   geom_boxplot(fill='lightblue') + facet_wrap(~variable, scales = 'free') )
```



Adjust Skewed values

Box Cox Transformation Removing the outliers tranformed each variable to a closer to a normal distribution and checking the skewness of the variables confirm this with the exception of TEAM_FIELDING_E. This variable is still skewed and not normal. In this section, we will use the Box Cox tranformation from the MASS library to normalize this variable.

```
metastats_02 <- data.frame(describe(baseball_train_02))
```

```
## Warning in describe(baseball_train_02): NAs introduced by coercion

## Warning in describe(baseball_train_02): NAs introduced by coercion

## Warning in describe(baseball_train_02): NAs introduced by coercion

## Warning in describe(baseball_train_02): NAs introduced by coercion
```

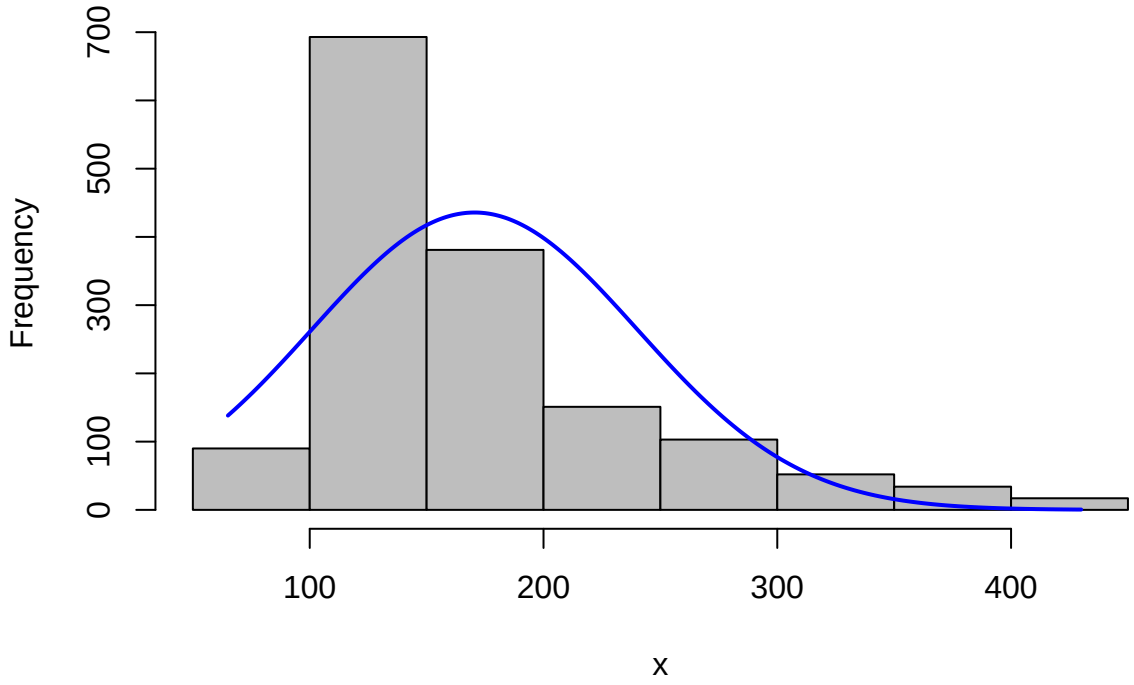
```
## Warning in describe(baseball_train_02): NAs introduced by coercion

## Warning in describe(baseball_train_02): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf
```

```r
metastats_02 <- tibble::rownames_to_column(metastats_02, "STATS")

metastats_02 %>%
 filter(skew > 1 | skew < -1) %>%
  dplyr::select(STATS, skew) %>%
  arrange(desc(skew))
```

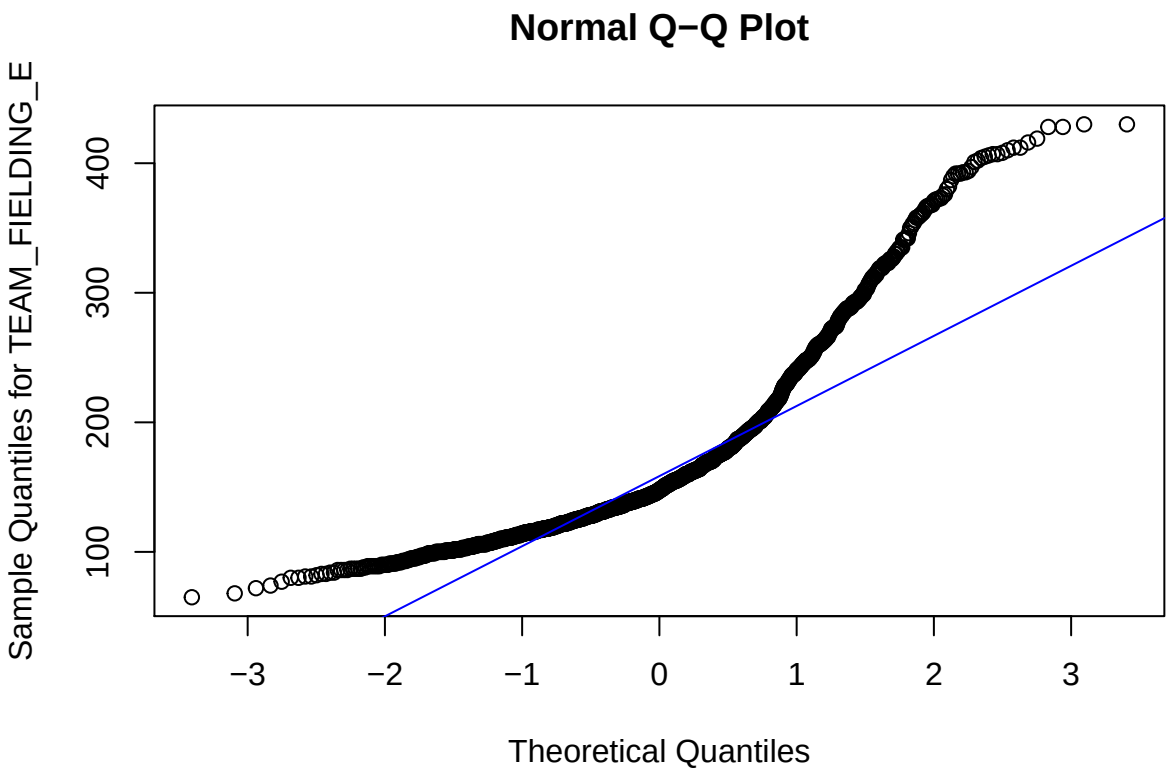
STATS	skew
TEAM_FIELDING_E	1.45

Looking at the histogram and QQ plots we can confirm that the variable, TEAM_FIELDING_E, is not normally distributed. It is skewed to the right.

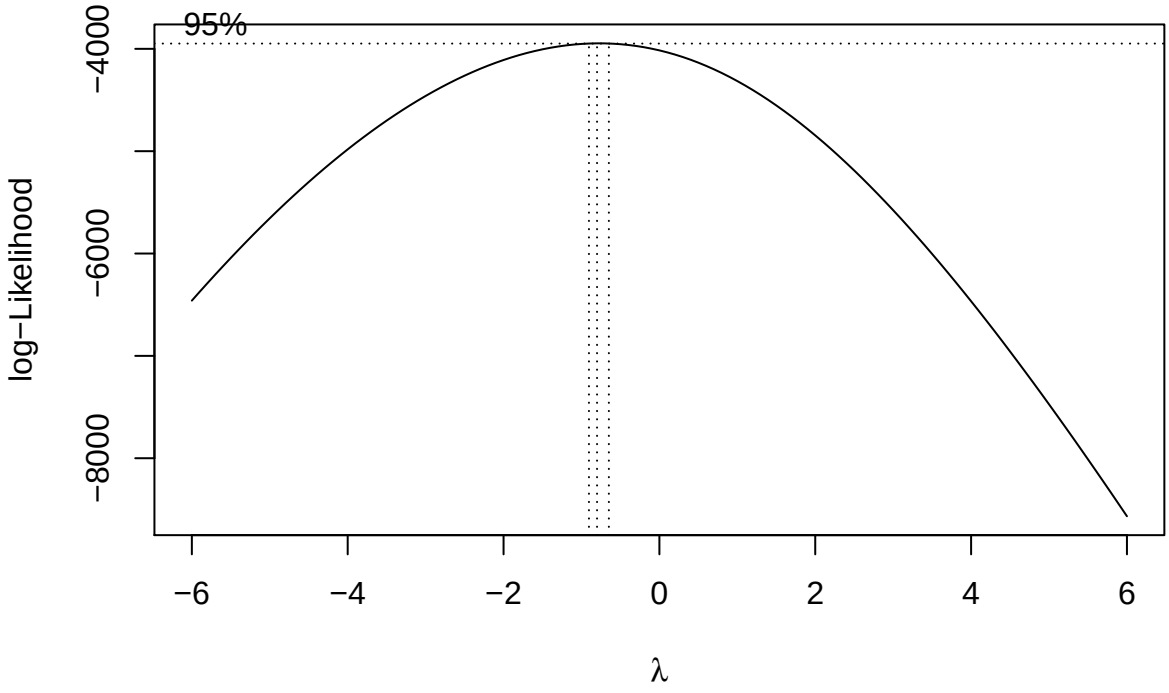
```
plotNormalHistogram(baseball_train_02$TEAM_FIELDING_E)
```




```
qqnorm(baseball_train_02$TEAM_FIELDING_E,
       ylab="Sample Quantiles for TEAM_FIELDING_E")
         qqline(baseball_train_02$TEAM_FIELDING_E,
           col="blue")
```




```
TEAM_FIELDING_E <- as.numeric(dplyr::pull(baseball_train_02, TEAM_FIELDING_E))

#Transforms TEAM_FIELDING_E as a single vector
Box = boxcox(TEAM_FIELDING_E ~ 1, lambda = seq(-6,6,0.1))
```



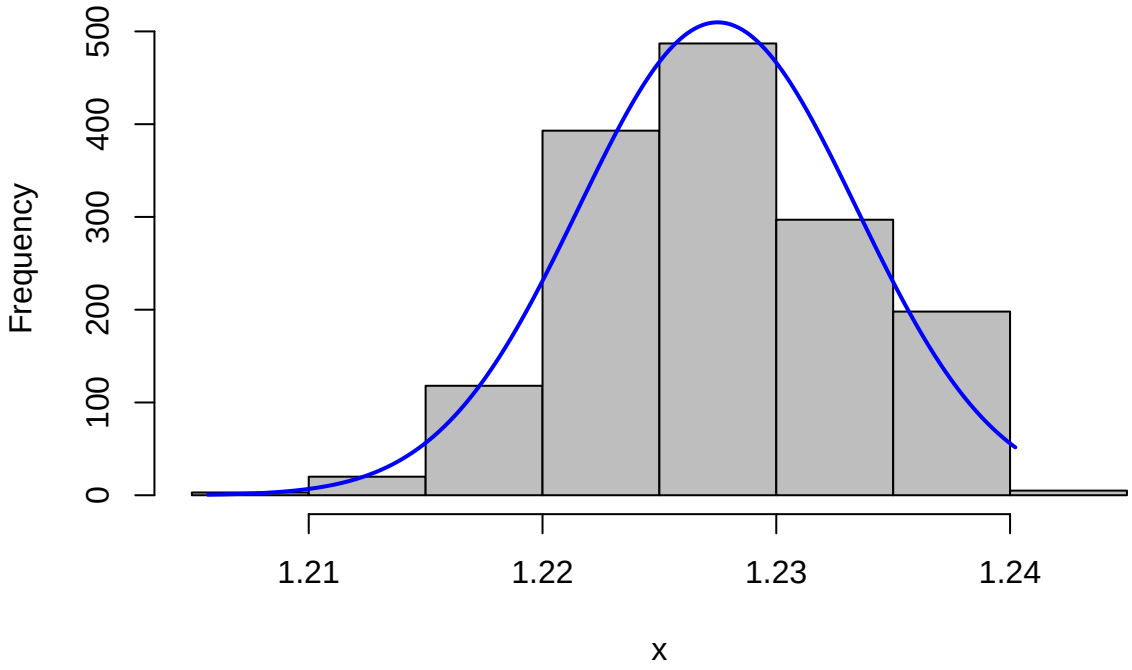

```
#Creates a dataframe with results
Cox = data.frame(Box$x, Box$y)

# Order the new data frame by decreasing y to find the best lambda.Displays the lambda with the greates
Cox2 = Cox[with(Cox, order(-Cox$Box.y)),]
Cox2[1,]
```

Box.x	Box.y
-0.8	-3.95e+03

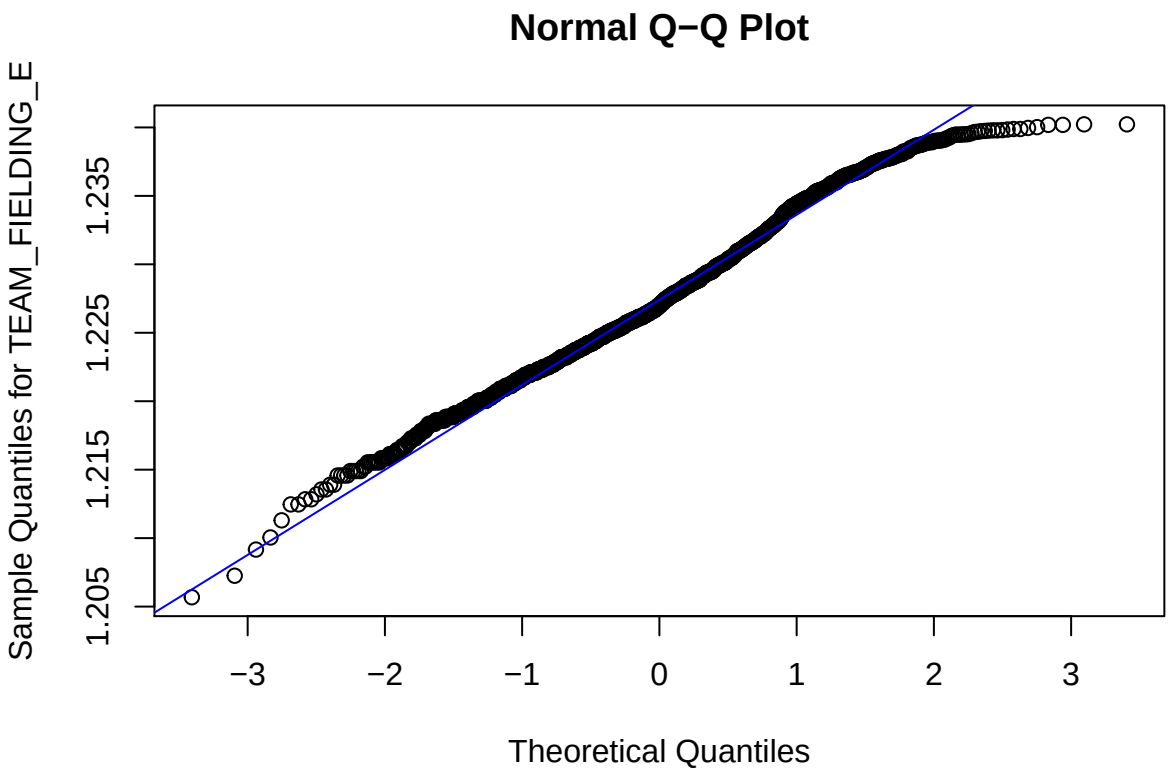
```
#Extract that lambda and Transform the data
lambda = Cox2[1, "Box.x"]
T_box = (TEAM_FIELDING_E ^ lambda - 1)/lambda
```

We can now see that TEAM_FIELDING_E has a normal distribution.

```
plotNormalHistogram(T_box)
```



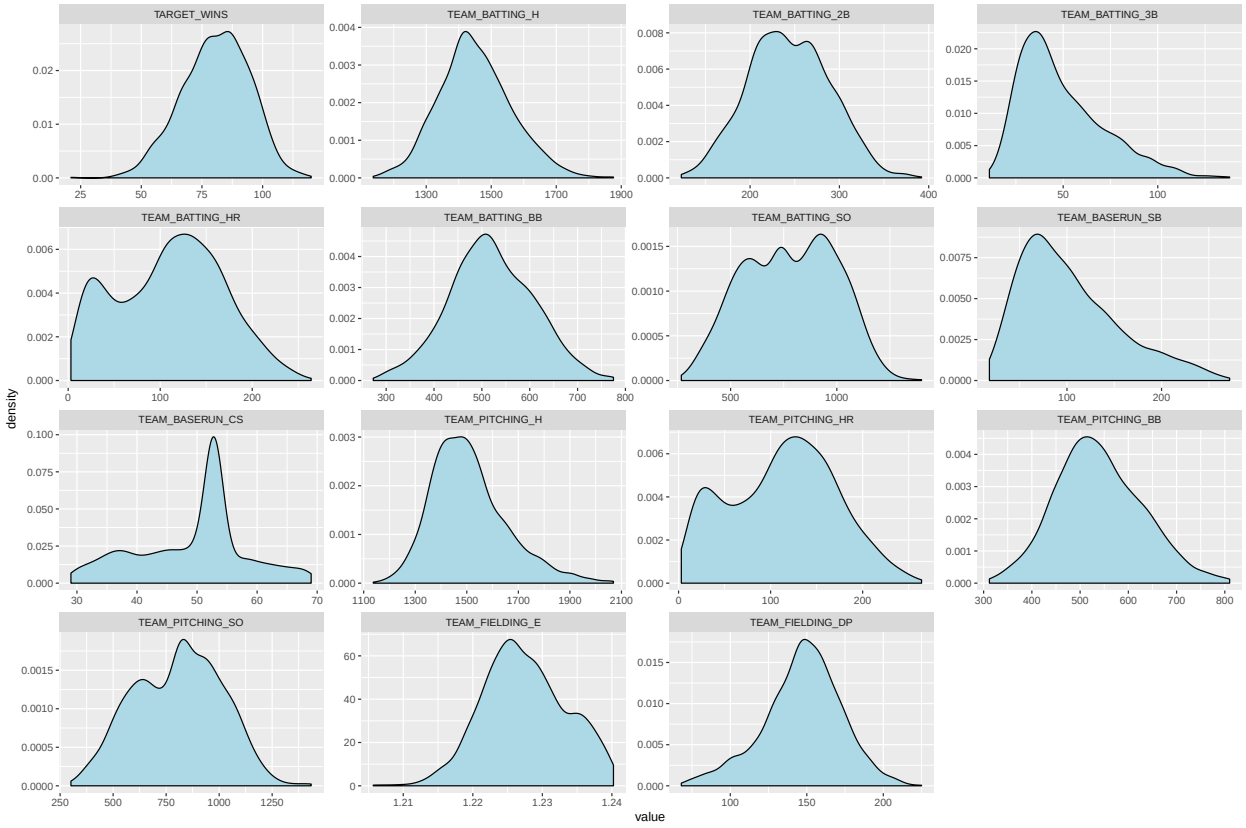

```
qqnorm(T_box, ylab="Sample Quantiles for TEAM_FIELDING_E")
qqline(T_box,
       col="blue")
```




```
baseball_train_02$TEAM_FIELDING_E <- T_box
```

The density plots below show that all of the variables for the dataset baseball_train_02 are now normally distributed. In the next section, we will use this dataset to build the models and discuss the coefficients of the models.

```
par(mfrow = c(3, 3))
datasub = melt(baseball_train_02)
suppressWarnings(ggplot(datasub, aes(x= value)) +
                   geom_density(fill='lightblue') + facet_wrap(~variable, scales = 'free') )
```



Viewing the dataframe shows that the dataset contains characters resulting from the transfromation of the outliers. These non numeric characters will impact our models especially if we build the intial baseline model with all the variables. We will need one more step to have our data ready for the models.

```
str(baseball_train_02)
```

```
## 'data.frame':    1521 obs. of  21 variables:
##  $ TARGET_WINS             : int  70 82 75 80 85 76 78 87 88 66 ...
##  $ TEAM_BATTING_H          : int  1387 1297 1279 1244 1273 1271 1305 1417 1563 1460 ...
##  $ TEAM_BATTING_2B         : int  209 186 200 179 171 213 179 226 242 239 ...
##  $ TEAM_BATTING_3B         : int  38 27 36 54 37 18 27 28 43 32 ...
##  $ TEAM_BATTING_HR         : int  96 102 92 122 115 96 82 108 164 107 ...
##  $ TEAM_BATTING_BB         : int  451 472 443 525 456 441 374 539 589 546 ...
##  $ TEAM_BATTING_SO         : num  922 920 973 1062 1027 ...
##  $ TEAM_BASERUN_SB         : num  43 49 107 80 40 72 60 86 100 92 ...
##  $ TEAM_BASERUN_CS         : num  30 39 59 54 36 34 39 69 53 64 ...
##  $ TEAM_PITCHING_H         : int  1396 1297 1279 1244 1281 1271 1364 1417 1563 1478 ...
##  $ TEAM_PITCHING_HR        : int  97 102 92 122 116 96 86 108 164 108 ...
##  $ TEAM_PITCHING_BB        : int  454 472 443 525 459 441 391 539 589 553 ...
##  $ TEAM_PITCHING_SO        : num  928 920 973 1062 1033 ...
##  $ TEAM_FIELDING_E         : num  1.23 1.23 1.22 1.23 1.22 ...
##  $ TEAM_FIELDING_DP        : num  156 168 149 186 136 159 141 136 172 146 ...
##  $ TEAM_BASERUN_SB_Outlier : chr  "Acceptable" "Acceptable" "Acceptable" "Acceptable" ...
##  $ TEAM_BASERUN_CS_Outlier : chr  "Acceptable" "Acceptable" "Acceptable" "Acceptable" ...
##  $ TEAM_PITCHING_H_Outlier : chr  "Acceptable" "Acceptable" "Acceptable" "Acceptable" ...
##  $ TEAM_PITCHING_BB_Outlier: chr  "Acceptable" "Acceptable" "Acceptable" "Acceptable" ...
```

```
##  $ TEAM_PITCHING_SO_Outlier: chr  "Acceptable" "Acceptable" "Acceptable" "Acceptable" ...
##  $ TEAM_FIELDING_E_Outlier : chr  "Acceptable" "Acceptable" "Acceptable" "Acceptable" ...
```

Subsetting - The code below will subset the data to have only numeric or integer values that will be used for our models. This will create baseball_train_03 dataframe.

```
baseball_train_03 <- baseball_train_02[c(1:15) ]
str(baseball_train_03)
```

```
## 'data.frame':    1521 obs. of  15 variables:
##  $ TARGET_WINS     : int  70 82 75 80 85 76 78 87 88 66 ...
##  $ TEAM_BATTING_H  : int  1387 1297 1279 1244 1273 1271 1305 1417 1563 1460 ...
##  $ TEAM_BATTING_2B : int  209 186 200 179 171 213 179 226 242 239 ...
##  $ TEAM_BATTING_3B : int  38 27 36 54 37 18 27 28 43 32 ...
##  $ TEAM_BATTING_HR : int  96 102 92 122 115 96 82 108 164 107 ...
##  $ TEAM_BATTING_BB : int  451 472 443 525 456 441 374 539 589 546 ...
##  $ TEAM_BATTING_SO : num  922 920 973 1062 1027 ...
##  $ TEAM_BASERUN_SB : num  43 49 107 80 40 72 60 86 100 92 ...
##  $ TEAM_BASERUN_CS : num  30 39 59 54 36 34 39 69 53 64 ...
##  $ TEAM_PITCHING_H : int  1396 1297 1279 1244 1281 1271 1364 1417 1563 1478 ...
##  $ TEAM_PITCHING_HR: int  97 102 92 122 116 96 86 108 164 108 ...
##  $ TEAM_PITCHING_BB: int  454 472 443 525 459 441 391 539 589 553 ...
##  $ TEAM_PITCHING_SO: num  928 920 973 1062 1033 ...
##  $ TEAM_FIELDING_E : num  1.23 1.23 1.22 1.23 1.22 ...
##  $ TEAM_FIELDING_DP: num  156 168 149 186 136 159 141 136 172 146 ...
```

Build Models

The first Model is using stepwise in Backward direction to eliminate variables, this is an automated process which is different from the manual variable selction process. We will not pay much attention to this process as the focus of the project is to manually identify and select those significant variables that will predict TARGET WINS.

```
Model <- step(lm(TARGET_WINS ~ ., data=baseball_train_03), direction = "backward")
```

```
## Start:  AIC=7313.35
## TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
##     TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB +
##     TEAM_BASERUN_CS + TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_BB +
##     TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP
##
##                     Df Sum of Sq    RSS    AIC
## - TEAM_PITCHING_H    1       0.7 182710 7311.4
## - TEAM_PITCHING_HR   1     162.2 182872 7312.7
## - TEAM_BATTING_H     1     216.2 182926 7313.2
## <none>                           182709 7313.4
## - TEAM_BASERUN_CS    1     330.3 183040 7314.1
## - TEAM_BATTING_HR    1     338.0 183047 7314.2
## - TEAM_PITCHING_BB   1     363.7 183073 7314.4
## - TEAM_BATTING_BB    1     629.6 183339 7316.6
## - TEAM_PITCHING_SO   1    1242.9 183952 7321.7
## - TEAM_BATTING_SO    1    1857.6 184567 7326.7
```

```
## - TEAM_BATTING_2B   1    1864.9 184574 7326.8
## - TEAM_FIELDING_DP  1    6690.2 189400 7366.1
## - TEAM_BATTING_3B   1    7536.4 190246 7372.8
## - TEAM_BASERUN_SB   1    8080.4 190790 7377.2
## - TEAM_FIELDING_E   1   18743.7 201453 7459.9
##
## Step:  AIC=7311.36
## TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
##     TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB +
##     TEAM_BASERUN_CS + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_PITCHING_SO +
##     TEAM_FIELDING_E + TEAM_FIELDING_DP
##
##                     Df Sum of Sq    RSS    AIC
## - TEAM_PITCHING_HR  1     173.3 182883 7310.8
## <none>                          182710 7311.4
## - TEAM_BASERUN_CS   1     331.1 183041 7312.1
## - TEAM_BATTING_HR   1     358.9 183069 7312.3
## - TEAM_PITCHING_SO  1    1259.1 183969 7319.8
## - TEAM_PITCHING_BB  1    1509.7 184220 7321.9
## - TEAM_BATTING_2B   1    1876.6 184587 7324.9
## - TEAM_BATTING_SO   1    1880.0 184590 7324.9
## - TEAM_BATTING_BB   1    2658.3 185368 7331.3
## - TEAM_BATTING_H    1    4833.0 187543 7349.1
## - TEAM_FIELDING_DP  1    6705.4 189416 7364.2
## - TEAM_BATTING_3B   1    7548.6 190259 7370.9
## - TEAM_BASERUN_SB   1    8142.6 190853 7375.7
## - TEAM_FIELDING_E   1   18841.1 201551 7458.6
##
## Step:  AIC=7310.8
## TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
##     TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB +
##     TEAM_BASERUN_CS + TEAM_PITCHING_BB + TEAM_PITCHING_SO + TEAM_FIELDING_E +
##     TEAM_FIELDING_DP
##
##                     Df Sum of Sq    RSS    AIC
## <none>                          182883 7310.8
## - TEAM_BASERUN_CS   1     418.7 183302 7312.3
## - TEAM_PITCHING_SO  1    1202.0 184085 7318.8
## - TEAM_PITCHING_BB  1    1537.3 184421 7321.5
## - TEAM_BATTING_2B   1    1928.3 184812 7324.8
## - TEAM_BATTING_SO   1    1977.9 184861 7325.2
## - TEAM_BATTING_BB   1    2678.0 185561 7330.9
## - TEAM_BATTING_H    1    4860.2 187744 7348.7
## - TEAM_BATTING_HR   1    5541.1 188424 7354.2
## - TEAM_BATTING_3B   1    7420.8 190304 7369.3
## - TEAM_FIELDING_DP  1    7423.8 190307 7369.3
## - TEAM_BASERUN_SB   1    9570.9 192454 7386.4
## - TEAM_FIELDING_E   1   18687.6 201571 7456.8
```

```
summary(Model)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
```

```
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
##     TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_PITCHING_BB + TEAM_PITCHING_SO +
##     TEAM_FIELDING_E + TEAM_FIELDING_DP, data = baseball_train_03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.468  -6.985  -0.128   7.454  34.637
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.368e+03  1.084e+02  12.624  < 2e-16 ***
## TEAM_BATTING_H    3.169e-02  5.006e-03   6.331 3.22e-10 ***
## TEAM_BATTING_2B  -4.198e-02  1.053e-02  -3.987 7.00e-05 ***
## TEAM_BATTING_3B   1.756e-01  2.244e-02   7.822 9.68e-15 ***
## TEAM_BATTING_HR   7.519e-02  1.112e-02   6.759 1.97e-11 ***
## TEAM_BATTING_BB   1.564e-01  3.328e-02   4.699 2.85e-06 ***
## TEAM_BATTING_SO  -8.768e-02  2.171e-02  -4.038 5.65e-05 ***
## TEAM_BASERUN_SB   6.625e-02  7.458e-03   8.884  < 2e-16 ***
## TEAM_BASERUN_CS  -6.510e-02  3.503e-02  -1.858 0.063350 .
## TEAM_PITCHING_BB -1.130e-01  3.175e-02  -3.560 0.000382 ***
## TEAM_PITCHING_SO  6.484e-02  2.059e-02   3.148 0.001675 **
## TEAM_FIELDING_E  -1.085e+03  8.739e+01 -12.413  < 2e-16 ***
## TEAM_FIELDING_DP -1.121e-01  1.433e-02  -7.824 9.56e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.01 on 1508 degrees of freedom
## Multiple R-squared:  0.3725, Adjusted R-squared:  0.3675
## F-statistic: 74.59 on 12 and 1508 DF,  p-value: < 2.2e-16
```

The step backward variable selection process identified eleven significant variables with an R-squared of 37%, Residual Error of 11.01 and F-Statistic of 74.59. Notice that some of the coefficients are negative which means these Team will most likely result in negative wins. We will explore these coefficient a little further in this analysis.

OLS- MODEL 1

Using all the 15 Variables

```
Model1 <-lm(TARGET_WINS ~ ., data=baseball_train_03)
summary(Model1)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = baseball_train_03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.067  -7.014  -0.101   7.499  34.361
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.376e+03  1.089e+02  12.634  < 2e-16 ***
```

```
## TEAM_BATTING_H    2.992e-02  2.242e-02   1.335   0.1821
## TEAM_BATTING_2B  -4.140e-02  1.056e-02  -3.921 9.23e-05 ***
## TEAM_BATTING_3B   1.775e-01  2.252e-02   7.882 6.15e-15 ***
## TEAM_BATTING_HR   2.424e-01  1.452e-01   1.669   0.0953 .
## TEAM_BATTING_BB   1.606e-01  7.051e-02   2.278   0.0229 *
## TEAM_BATTING_SO  -1.072e-01  2.741e-02  -3.913 9.52e-05 ***
## TEAM_BASERUN_SB   6.361e-02  7.794e-03   8.161 6.94e-16 ***
## TEAM_BASERUN_CS  -5.853e-02  3.547e-02  -1.650   0.0992 .
## TEAM_PITCHING_H   1.587e-03  2.058e-02   0.077   0.9386
## TEAM_PITCHING_HR -1.617e-01  1.398e-01  -1.156   0.2478
## TEAM_PITCHING_BB -1.166e-01  6.735e-02  -1.732   0.0836 .
## TEAM_PITCHING_SO  8.366e-02  2.614e-02   3.201   0.0014 **
## TEAM_FIELDING_E  -1.092e+03  8.785e+01 -12.430  < 2e-16 ***
## TEAM_FIELDING_DP -1.086e-01  1.463e-02  -7.426 1.87e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.01 on 1506 degrees of freedom
## Multiple R-squared:  0.3731, Adjusted R-squared:  0.3672
## F-statistic: 64.01 on 14 and 1506 DF,  p-value: < 2.2e-16
```

This Model identified seven significant variables at $\alpha = 0.05$ with an R-squared of 37%, Residual Error of 11.01 and F-Statistic of 64.01. Although the F-Statistic reduced, this model does not improve significantly from the previous model.

```
Metrics1 <- data.frame(
  R2 = rsquare(Model1, data = baseball_train_03),
  RMSE = rmse(Model1, data = baseball_train_03),
  MAE = mae(Model1, data = baseball_train_03)
)
print(Metrics1)
```

```
##          R2     RMSE      MAE
## 1 0.3730717 10.96013 8.741105
```

OLS- MODEL 2

Using all the seven (7) significant variables from Model 1

```
Model2 <- lm(TARGET_WINS~TEAM_FIELDING_E + TEAM_BASERUN_SB + TEAM_BATTING_3B + TEAM_FIELDING_DP + TEAM_
summary(Model2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_FIELDING_E + TEAM_BASERUN_SB +
##     TEAM_BATTING_3B + TEAM_FIELDING_DP + TEAM_PITCHING_SO + TEAM_BATTING_SO +
##     TEAM_BATTING_2B, data = baseball_train_03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.799  -8.299  -0.053   8.472  39.785
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.808e+03  1.158e+02  15.607  < 2e-16 ***
## TEAM_FIELDING_E  -1.410e+03  9.313e+01 -15.141  < 2e-16 ***
## TEAM_BASERUN_SB   5.429e-02  7.497e-03   7.242 7.02e-13 ***
## TEAM_BATTING_3B   1.788e-01  2.289e-02   7.808 1.08e-14 ***
## TEAM_FIELDING_DP -5.319e-02  1.525e-02  -3.488 0.000501 ***
## TEAM_PITCHING_SO -8.587e-03  9.176e-03  -0.936 0.349497
## TEAM_BATTING_SO  -8.186e-03  9.308e-03  -0.880 0.379267
## TEAM_BATTING_2B   4.475e-02  7.971e-03   5.614 2.35e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.19 on 1513 degrees of freedom
## Multiple R-squared:  0.2288, Adjusted R-squared:  0.2253
## F-statistic: 64.14 on 7 and 1513 DF,  p-value: < 2.2e-16
```

This Model identified five significant variables at $\alpha = 0.05$ with an R-squared of 22%, Residual Error of 12.19 and F-Statistic of 64.14. The R-Squared decreased and the Error increased slightly.

```
Metrics2 <- data.frame(
  R2 = rsquare(Model2, data = baseball_train_03),
  RMSE = rmse(Model2, data = baseball_train_03),
  MAE = mae(Model2, data = baseball_train_03)
)
print(Metrics2)
```

```
##          R2     RMSE      MAE
## 1 0.2288348 12.15572 9.731629
```

OLS- MODEL 3

All offensive categories which include hitting and base running

```
Model3 <-lm(TARGET_WINS~TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BATTING_HR + TEAM_BATTING_2B + TEAM_BAT
summary(Model3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB +
##     TEAM_BATTING_HR + TEAM_BATTING_2B + TEAM_BATTING_SO + TEAM_BASERUN_CS +
##     TEAM_BATTING_3B + TEAM_BASERUN_SB, data = baseball_train_03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.812  -7.822   0.247   8.166  35.877
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     15.763131   7.024077   2.244   0.0250 *
## TEAM_BATTING_H   0.024765   0.005285   4.686 3.03e-06 ***
## TEAM_BATTING_BB  0.037681   0.003994   9.435  < 2e-16 ***
```

```
## TEAM_BATTING_HR  0.099319   0.011448   8.676  < 2e-16 ***
## TEAM_BATTING_2B -0.013435   0.010919  -1.230   0.2187
## TEAM_BATTING_SO -0.010801   0.002767  -3.904 9.88e-05 ***
## TEAM_BASERUN_CS -0.068614   0.037166  -1.846   0.0651 .
## TEAM_BATTING_3B  0.115379   0.022950   5.027 5.57e-07 ***
## TEAM_BASERUN_SB  0.076701   0.007431  10.321  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.73 on 1512 degrees of freedom
## Multiple R-squared:  0.2857, Adjusted R-squared:  0.2819
## F-statistic: 75.58 on 8 and 1512 DF,  p-value: < 2.2e-16
```

This Model identified five significant variables at $\alpha = 0.05$ with an R-squared of 28%, Residual Error of 11.73 and F-Statistic of 75.58. Although the R-squared is not that great, the standard errors are more reasonable. We will hold onto this Model as performing better than the previous models for now.

```
Metrics3 <- data.frame(
  R2 = rsquare(Model3, data = baseball_train_03),
  RMSE = rmse(Model3, data = baseball_train_03),
  MAE = mae(Model3, data = baseball_train_03)
)
print(Metrics3)
```

```
##          R2     RMSE      MAE
## 1 0.2856527 11.69934 9.330048
```

OLS- MODEL 4

All defensive categories which include fielding and pitching

```
Model4 <- lm(TARGET_WINS~TEAM_PITCHING_H + TEAM_PITCHING_BB + TEAM_PITCHING_HR + TEAM_PITCHING_SO + TEA
summary(Model4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_PITCHING_H + TEAM_PITCHING_BB +
##     TEAM_PITCHING_HR + TEAM_PITCHING_SO + TEAM_FIELDING_E, data = baseball_train_03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.818  -8.397   0.393   8.617  42.600
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.915e+02  1.079e+02   7.335 3.61e-13 ***
## TEAM_PITCHING_H   2.420e-02  2.705e-03   8.947  < 2e-16 ***
## TEAM_PITCHING_BB  2.542e-02  3.943e-03   6.448 1.52e-10 ***
## TEAM_PITCHING_HR  1.503e-02  9.799e-03   1.533 0.125415
## TEAM_PITCHING_SO -9.205e-03  2.369e-03  -3.886 0.000106 ***
## TEAM_FIELDING_E  -6.153e+02  8.770e+01  -7.016 3.44e-12 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.46 on 1515 degrees of freedom
## Multiple R-squared:  0.1932, Adjusted R-squared:  0.1905
## F-statistic: 72.56 on 5 and 1515 DF,  p-value: < 2.2e-16
```

This Model identified five significant variables at $\alpha = 0.05$ with an R-squared of 19%, Residual Error of 12.46 and F-Statistic of 75.56.There is no significant improvement with this model.

```
Metrics4 <- data.frame(
  R2 = rsquare(Model4, data = baseball_train_03),
  RMSE = rmse(Model4, data = baseball_train_03),
  MAE = mae(Model4, data = baseball_train_03)
)
print(Metrics4)
```

```
##          R2     RMSE      MAE
## 1 0.1932003 12.43339 9.945165
```

OLS- MODEL 5

Using only the significant variables from Model 3

```
Model5 <- lm(TARGET_WINS~TEAM_PITCHING_H + TEAM_PITCHING_BB + TEAM_PITCHING_HR + TEAM_PITCHING_SO + TEAI
summary(Model5)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_PITCHING_H + TEAM_PITCHING_BB +
##     TEAM_PITCHING_HR + TEAM_PITCHING_SO + TEAM_BATTING_3B + TEAM_BASERUN_SB,
##     data = baseball_train_03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.002  -7.725   0.407   8.171  36.647
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      44.113607   4.898047   9.006  < 2e-16 ***
## TEAM_PITCHING_H   0.002544   0.002951   0.862    0.389
## TEAM_PITCHING_BB  0.029880   0.003774   7.917 4.66e-15 ***
## TEAM_PITCHING_HR  0.134928   0.009503  14.198  < 2e-16 ***
## TEAM_PITCHING_SO -0.016789   0.002522  -6.657 3.91e-11 ***
## TEAM_BATTING_3B   0.141192   0.023104   6.111 1.26e-09 ***
## TEAM_BASERUN_SB   0.077656   0.007190  10.800  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.93 on 1514 degrees of freedom
## Multiple R-squared:  0.2606, Adjusted R-squared:  0.2576
## F-statistic: 88.92 on 6 and 1514 DF,  p-value: < 2.2e-16
```

This Model identified five significant variables at $\alpha = 0.05$ with an R-squared of 26%, Residual Error of 11.93 and F-Statistic of 88.92. Although the R-squared is not better than than Model3, the F-statistic improved with smaller Standard Error.

```
Metrics5 <- data.frame(
  R2 = rsquare(Model5, data = baseball_train_03),
  RMSE = rmse(Model5, data = baseball_train_03),
  MAE = mae(Model5, data = baseball_train_03)
)
print(Metrics5)
```

```
##          R2     RMSE      MAE
## 1 0.2605772 11.90291 9.506094
```

Compare OLS Model Quality

```
anova(Model, Model1, Model2, Model3, Model4, Model5)
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1.51e+03	1.83e+05				
1.51e+03	1.83e+05	2	174	0.717	0.488
1.51e+03	2.25e+05	-7	-4.2e+04	49.5	1.39e-63
1.51e+03	2.08e+05	1	1.66e+04	136	3.01e-30
1.52e+03	2.35e+05	-3	-2.69e+04	74	1.15e-44
1.51e+03	2.15e+05	1	1.96e+04	162	2.72e-35

```
tab_model(Model, Model1, Model2, Model3, Model4, Model5)
```

TARGET WINS

TARGET WINS

TARGET WINS

TARGET WINS

TARGET WINS

TARGET WINS

Predictors

Estimates

CI

p

Estimates

CI

p

Estimates

CI

p

Estimates

CI

p

Estimates

CI

p

Estimates

CI

p

(Intercept)

1368.26

1155.66 – 1580.87

<0.001

1376.10

1162.45 – 1589.76

<0.001

1807.57

1580.39 – 2034.75

<0.001

15.76

1.99 – 29.54

0.025

791.49

579.82 – 1003.16

<0.001

44.11

34.51 – 53.72

<0.001

TEAM_BATTING_H

0.03

0.02 – 0.04

<0.001

0.03

-0.01 – 0.07

0.182

0.02

0.01 – 0.04

<0.001

TEAM_BATTING_2B

-0.04

-0.06 – -0.02

<0.001

-0.04

-0.06 – -0.02

<0.001

0.04

0.03 – 0.06

<0.001

-0.01

-0.03 – 0.01

0.219

TEAM_BATTING_3B

0.18

0.13 – 0.22

<0.001

0.18

0.13 – 0.22

<0.001

0.18

0.13 – 0.22

<0.001

0.12

0.07 – 0.16

<0.001

0.14

0.10 – 0.19

<0.001

TEAM_BATTING_HR

0.08

0.05 – 0.10

<0.001

0.24

-0.04 – 0.53

0.095

0.10

0.08 – 0.12

<0.001

TEAM_BATTING_BB

0.16

0.09 – 0.22

<0.001

0.16

0.02 – 0.30

0.023

0.04

0.03 – 0.05

<0.001

TEAM_BATTING_SO

-0.09

-0.13 – -0.05

<0.001

-0.11

-0.16 – -0.05

<0.001

-0.01

-0.03 – 0.01

0.379

-0.01

-0.02 – -0.01

<0.001

TEAM_BASERUN_SB

0.07

0.05 – 0.08

<0.001

0.06

0.05 – 0.08

<0.001

0.05

0.04 – 0.07

<0.001

0.08

0.06 – 0.09

<0.001

0.08

0.06 – 0.09

<0.001

TEAM_BASERUN_CS

-0.07

-0.13 – 0.00

0.063

-0.06

-0.13 – 0.01

0.099

-0.07

-0.14 – 0.00

0.065

TEAM_PITCHING_BB

-0.11

-0.18 – -0.05

<0.001

-0.12

-0.25 – 0.02

0.084

0.03

0.02 – 0.03

<0.001

0.03

0.02 – 0.04

<0.001

TEAM_PITCHING_SO

0.06

0.02 – 0.11

0.002

0.08

0.03 – 0.13

0.001

-0.01

-0.03 – 0.01

0.349

-0.01

-0.01 – -0.00

<0.001

-0.02

-0.02 – -0.01

<0.001

TEAM_FIELDING_E

-1084.76

-1256.17 – -913.35

<0.001

-1091.94

-1264.26 – -919.62

<0.001

-1410.16

-1592.85 – -1227.48

<0.001

-615.31

-787.34 – -443.27

<0.001

TEAM_FIELDING_DP

-0.11

-0.14 – -0.08

<0.001

-0.11

-0.14 – -0.08

<0.001

-0.05

-0.08 – -0.02

0.001

TEAM_PITCHING_H

0.00

-0.04 – 0.04

0.939

0.02

0.02 – 0.03

<0.001

0.00

-0.00 – 0.01

0.389

TEAM_PITCHING_HR

-0.16

-0.44 – 0.11

0.248

0.02

-0.00 – 0.03

0.125

0.13

0.12 – 0.15

<0.001

Observations

1521

1521

1521

1521

1521

1521

R2 / R2 adjusted

0.372 / 0.367

0.373 / 0.367

0.229 / 0.225

0.286 / 0.282

0.193 / 0.191

0.261 / 0.258

RIDGE Regression- MODEL 6

The Ridge regression is an extension of linear regression where the loss function is modified to minimize the complexity of the model. This modification is done by adding a penalty parameter that is equivalent to the square of the magnitude of the coefficients.

Before implementing the RIDGE model, we will split the training dataset into 2 parts that is - training set within the training set and a test set that can be used for evaluation. By enforcing stratified sampling both our training and testing sets have approximately equal response "TARGET_WINS" distributions.

Transforming the variables into the form of a matrix will enable us to penalize the model using the 'glmnet' method in glmnet package.

```
#Split the data into Training and Test Set
baseball_train_set<- initial_split(baseball_train_03, prop = 0.7, strata = "TARGET_WINS")
train_baseball  <- training(baseball_train_set)
test_baseball   <- testing(baseball_train_set)

train_Ind<- as.matrix(train_baseball)
train_Dep<- as.matrix(train_baseball$TARGET_WINS)

test_Ind<- as.matrix(test_baseball)
test_Dep<- as.matrix(test_baseball$TARGET_WINS)
```

For the avoidance of multicollinearity, avoiding overfitting and predicting better, implementing RIDGE regression will become useful.

```
lambdas <- 10^seq(2, -3, by = -.1)
Model6 <- glmnet(train_Ind,train_Dep, nlambda = 25, alpha = 0, family = 'gaussian', lambda = lambdas)
summary(Model6)
```

```
##           Length Class     Mode
## a0         51    -none-    numeric
## beta      765    dgCMatrix S4
## df         51    -none-    numeric
## dim         2    -none-    numeric
## lambda     51    -none-    numeric
## dev.ratio  51    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        7    -none-    call
## nobs        1    -none-    numeric
```

```
print(Model6, digits = max(3, getOption("digits") - 3),
          signif.stars = getOption("show.signif.stars"))
```

```
##
## Call:  glmnet(x = train_Ind, y = train_Dep, family = "gaussian", alpha = 0,      nlambda = 25, lambda
##
##    Df   %Dev  Lambda
## 1  15 0.2972 100.000
```

```
## 2  15 0.3464  79.430
## 3  15 0.3993  63.100
## 4  15 0.4554  50.120
## 5  15 0.5135  39.810
## 6  15 0.5724  31.620
## 7  15 0.6308  25.120
## 8  15 0.6873  19.950
## 9  15 0.7405  15.850
## 10 15 0.7892  12.590
## 11 15 0.8325  10.000
## 12 15 0.8697   7.943
## 13 15 0.9009   6.310
## 14 15 0.9261   5.012
## 15 15 0.9459   3.981
## 16 15 0.9612   3.162
## 17 15 0.9726   2.512
## 18 15 0.9809   1.995
## 19 15 0.9869   1.585
## 20 15 0.9911   1.259
## 21 15 0.9940   1.000
## 22 15 0.9960   0.794
## 23 15 0.9974   0.631
## 24 15 0.9983   0.501
## 25 15 0.9989   0.398
## 26 15 0.9993   0.316
## 27 15 0.9995   0.251
## 28 15 0.9997   0.200
## 29 15 0.9998   0.158
## 30 15 0.9999   0.126
## 31 15 0.9999   0.100
## 32 15 0.9999   0.079
## 33 15 1.0000   0.063
## 34 15 1.0000   0.050
## 35 15 1.0000   0.040
## 36 15 1.0000   0.032
## 37 15 1.0000   0.025
## 38 15 1.0000   0.020
## 39 15 1.0000   0.016
## 40 15 1.0000   0.013
## 41 15 1.0000   0.010
## 42 15 1.0000   0.008
## 43 15 1.0000   0.006
## 44 15 1.0000   0.005
## 45 15 1.0000   0.004
## 46 15 1.0000   0.003
## 47 15 1.0000   0.003
## 48 15 1.0000   0.002
## 49 15 1.0000   0.002
## 50 15 1.0000   0.001
## 51 15 1.0000   0.001
```

The significant difference between the OLS and the Ridge Regresion is the hyperparameter tuning using lambda. The Ridge regression does not perform Feature Selection, but it predicts better and solve overfitting. Cross Validating the Ridge Regression will help us to identify the optimal lambda to penalize the model and

enhance the predictability.

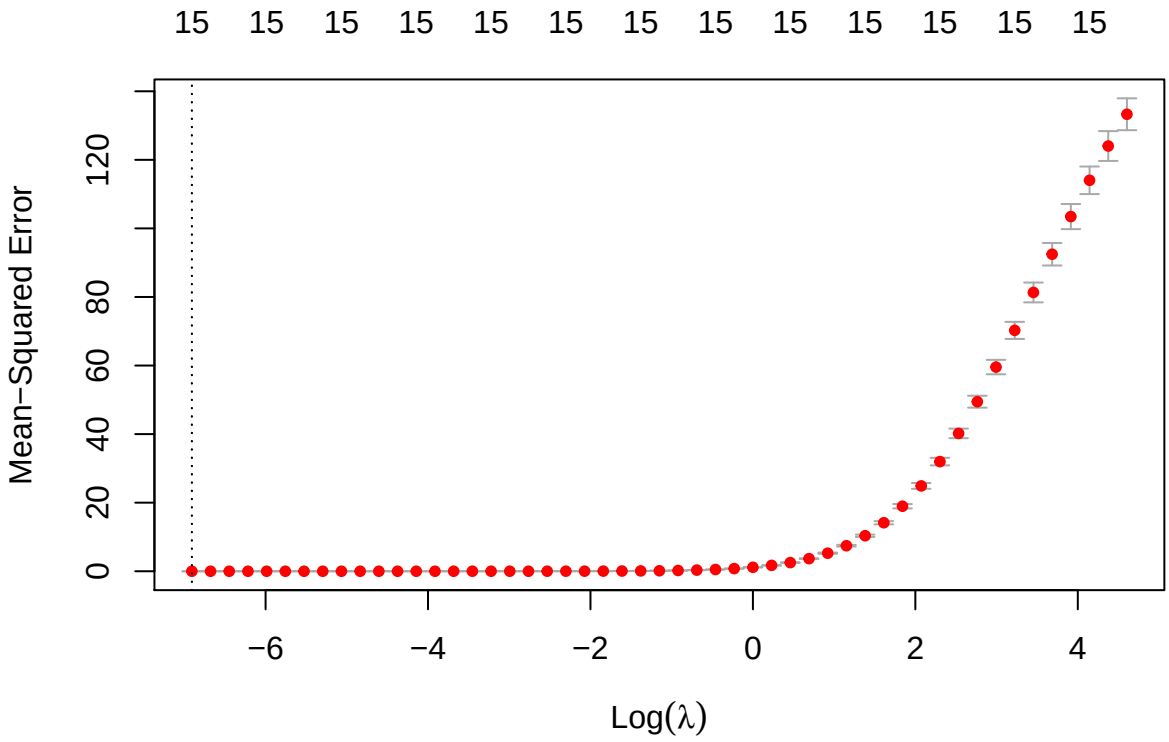
```
CrossVal_ridge <- cv.glmnet(train_Ind,train_Dep, alpha = 0, lambda = lambdas)
optimal_lambda <- CrossVal_ridge$lambda.min
optimal_lambda #The optimal lambda is 0.001 which we will use to penelize the Ridge Regression model.
```

```
## [1] 0.001
```

```
coef(CrossVal_ridge) # Shows the coefficients
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)       2.454750e-01
## TARGET_WINS       9.998795e-01
## TEAM_BATTING_H    2.782574e-05
## TEAM_BATTING_2B  -4.685653e-06
## TEAM_BATTING_3B   1.763536e-05
## TEAM_BATTING_HR  -4.037589e-04
## TEAM_BATTING_BB  -1.118614e-05
## TEAM_BATTING_SO  -7.425840e-06
## TEAM_BASERUN_SB   1.709784e-05
## TEAM_BASERUN_CS  -2.475998e-05
## TEAM_PITCHING_H  -2.414711e-05
## TEAM_PITCHING_HR  4.015911e-04
## TEAM_PITCHING_BB  1.179415e-05
## TEAM_PITCHING_SO  3.615699e-06
## TEAM_FIELDING_E  -1.926279e-01
## TEAM_FIELDING_DP -1.348863e-05
```

```
plot(CrossVal_ridge)
```



The plot shows that the errors increases as the magnitude of lambda increases, previously, we identified that the optimal lambda is 0.001 which is very obvious from the plot above. The coefficients are restricted to be small but not quite zero as Ridge Regression does not force the coefficient to zero. This indicates that the model is performing well so far. But let's make it better using the optimal labmda.

```
eval_results <- function(true, predicted, df){
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  R_square <- 1 - SSE / SST
  RMSE = sqrt(SSE/nrow(df))
data.frame(
  RMSE = RMSE,
  Rsquare = R_square
)

}
# Prediction and evaluation on train data
predictions_train <- predict(Model6, s = optimal_lambda, newx = train_Ind)
eval_results(train_Dep, predictions_train, train_baseball)
```

RMSE	Rsquare
0.00171	1

We should be a little concern about the 100% R-squared performance for this Model. Although the Ridge Regression forces the coefficients towards zero to improve the Model performance and enhance the pre-

dictability, the very high peformance may require further investigation. Lets improve the model using a more reason lambda because optimal might not always be the best.

The Improved Ridge Regression

```
Model6_Improved <- glmnet(train_Ind,train_Dep, nlambda = 25, alpha = 0, family = 'gaussian', lambda = 6
summary(Model6_Improved)
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      15     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       7     -none-    call
## nobs       1     -none-    numeric
```

```
coef(Model6_Improved)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                               s0
## (Intercept)       1.899746e+02
## TARGET_WINS       6.287136e-01
## TEAM_BATTING_H    6.263412e-03
## TEAM_BATTING_2B  -4.381383e-04
## TEAM_BATTING_3B   2.727159e-02
## TEAM_BATTING_HR   7.903321e-03
## TEAM_BATTING_BB   5.892008e-03
## TEAM_BATTING_SO  -1.088080e-03
## TEAM_BASERUN_SB   1.258018e-02
## TEAM_BASERUN_CS  -1.173467e-02
## TEAM_PITCHING_H   2.422667e-03
## TEAM_PITCHING_HR  7.713178e-03
## TEAM_PITCHING_BB  4.553239e-03
## TEAM_PITCHING_SO -1.405703e-03
## TEAM_FIELDING_E  -1.446905e+02
## TEAM_FIELDING_DP -1.564406e-02
```

Let's compute the Model's Performance Metric to see how this model is doing.

```
eval_results <- function(true, predicted, df){
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  R_square <- 1 - SSE / SST
  RMSE = sqrt(SSE/nrow(df))
```

```
data.frame(
  RMSE = RMSE,
  Rsquare = R_square
)

}

# Prediction and evaluation on train data
predictions_train <- predict(Model6_Improved, s = lambda, newx = train_Ind)
eval_results(train_Dep, predictions_train, train_baseball)
```

RMSE	Rsquare
4.33	0.901

```
# Prediction and evaluation on test data
predictions_test <- predict(Model6_Improved, s = lambda, newx = test_Ind)
eval_results(test_Dep, predictions_test, test_baseball)
```

RMSE	Rsquare
4.29	0.907

The improved Model6 output shows that the RMSE and R-squared values for the Ridge Regression model on the training and test data are significantly improved. The Loss Function (RMSE) are severely reduced compared to the OLS models which indicates that the Ridge Regression is not overfitting. These performance is significantly improved compared to the OLS Models 1 to 5.

Model Performance Comparison

```
ModelName <- c("Model", "Model1","Model2","Model3","Model4","Model5","Model6")
Model_RSquared <- c("37%", "37%", "22%", "28%", "19%", "26% ", "90%")
Model_RMSE <- c("11.01", "10.96", "12.15", "11.69", "12.43", "11.93 ", "4.33")
Model_FStatistic <- c("74.59", "64.01", "64.14", "75.58", "72.56", "88.92 ", "NA")
Model_Performance <- data.frame(ModelName,Model_RSquared,Model_RMSE,Model_FStatistic)
Model_Performance
```

ModelName	Model_RSquared	Model_RMSE	Model_FStatistic
Model	37%	11.01	74.59
Model1	37%	10.96	64.01
Model2	22%	12.15	64.14
Model3	28%	11.69	75.58
Model4	19%	12.43	72.56
Model5	26%	11.93	88.92
Model6	90%	4.33	NA

Model Prediction

Based on the Model metrics above, we're ready to make prediction and we will select our acceptable OLS Model3 and Model5 which has better F-Statistic, smaller standard errors and less negative coefficient as our

best OLS models. We will also compare the prediction accuracy of these models to that of the improved Ridge Regression Model which is our champion Model for this exercise based on the very small RMSE and the highest R-squared of over 90%.

```
predicted <- predict(Model3, newx = test_baseball)# predict on test data
predicted_values <- cbind (actual=test_baseball$TARGET_WINS, predicted)  # combine
```

```
## Warning in cbind(actual = test_baseball$TARGET_WINS, predicted): number of
## rows of result is not a multiple of vector length (arg 1)
```

```
predicted_values
```

```
##      actual predicted
## 1        75  69.49875
## 2        85  67.56125
## 3        76  68.38378
## 4        81  73.25628
## 5        92  67.37030
## 6        80  66.86382
## 7        85  63.36361
## 8        92  76.58169
## 9       100  89.59170
## 10       85  76.54657
## 11       98  86.70081
## 12       53  77.31870
## 13       63  78.60647
## 14       56  84.73255
## 15       47  89.15530
## 16       66  86.01142
## 17       53  76.15138
## 18       74  76.18203
## 19       67  79.02243
## 20       40  72.03653
## 21       96  89.30377
## 22       80  84.16328
## 23       94  79.30964
## 24       84  76.00667
## 25       88  93.36918
## 26       86  79.72780
## 27       88  83.96472
## 28       67  81.83322
## 29       70  86.75976
## 30       67  83.85432
## 31       66  75.24767
## 32      103  96.12431
## 33       76  85.18416
## 34       69  87.23777
## 35       48  86.66397
## 36       83  80.90673
## 37       77  71.73633
## 38       78  81.22686
## 39       86  90.64517
## 40       59  94.18601
```

```
## 41       45  95.53100
## 42       77  80.08285
## 43       73  75.86419
## 44       86  66.74352
## 45       98  64.57578
## 46      108  60.91243
## 47      104  72.28834
## 48       97  68.48019
## 49       92  65.67250
## 50       88  72.48471
## 51      104  86.44011
## 52      100  80.65902
## 53       94  82.06254
## 54       85  78.70595
## 55       84  77.58369
## 56       73  72.97126
## 57       67  78.18561
## 58       88  72.24895
## 59       67  76.25716
## 60       78  77.02116
## 61      104  75.30225
## 62       88  76.46272
## 63       49  78.36297
## 64       94  68.74638
## 65       79  70.51741
## 66       85  66.00240
## 67       78  74.32965
## 68       79  68.02212
## 69       75  71.72855
## 70       68  69.41157
## 71       83  69.61566
## 72       84  70.79319
## 73       73  69.48860
## 74       76  65.80599
## 75       97  70.70940
## 76       85  82.73716
## 77       78  77.23526
## 78       92  81.04205
## 79       80  86.68577
## 80       98  81.16218
## 81      104  88.42305
## 82       98  85.99275
## 83       93  73.58515
## 84      104  82.22049
## 85       95  85.06236
## 86       95  86.09993
## 87       87  76.29960
## 88       67  79.91231
## 89       76  77.53938
## 90       88  84.34369
## 91       92  74.79848
## 92       75  70.02759
## 93       79  82.39423
## 94       76  79.28918
```

```
## 95       77  91.97048
## 96       93  78.96951
## 97       66  74.19733
## 98       92  72.42719
## 99       71  77.14336
## 100      98  76.15731
## 101      94  77.37089
## 102      94  72.05640
## 103     105  76.89373
## 104      79  75.92414
## 105      87  85.81217
## 106      71  75.59281
## 107      89  80.60366
## 108      77  85.62160
## 109      96  82.00590
## 110      86  84.74713
## 111      98  84.88357
## 112      56  86.74637
## 113      81  89.85103
## 114      87  88.78040
## 115      99  78.46174
## 116      72  84.86044
## 117      77  83.42672
## 118      94  82.79613
## 119      77  82.72617
## 120      80  66.91725
## 121      83  69.51332
## 122      90  78.66910
## 123      69  72.33942
## 124      70  67.50771
## 125      68  64.77165
## 126      82  69.79720
## 127     111  71.15595
## 128      79  77.34661
## 129      56  77.95811
## 130      94  86.54917
## 131      73  73.89007
## 132      80  79.45996
## 133      77  76.41158
## 134     102  81.88671
## 135      93  84.23277
## 136      61  81.62631
## 137      84  78.66292
## 138      85  78.01240
## 139      78  82.19463
## 140      80  84.27986
## 141      85  73.82318
## 142      92  75.50558
## 143      93  72.20240
## 144      71  71.73169
## 145      82  78.70550
## 146      80  82.60550
## 147      79  79.89101
## 148      81  80.98410
```

```
## 149      60  73.62353
## 150      75  74.78823
## 151      77  70.09769
## 152      73  69.79510
## 153      77  73.21664
## 154      80  76.42604
## 155      21  70.00951
## 156      87  79.24643
## 157      82  76.55529
## 158      73  77.45710
## 159      74  77.29383
## 160      67  74.83120
## 161      76  80.57564
## 162      66  75.27052
## 163      95  72.96984
## 164      92  88.01300
## 165      93  89.33125
## 166      64  86.25568
## 167      87  69.87604
## 168      94  89.64940
## 169      81  80.48480
## 170     101  81.38282
## 171      79  78.12840
## 172      88  80.46360
## 173      79  77.90473
## 174      74  86.93266
## 175      86  88.60549
## 176      83  86.22276
## 177      85  85.13130
## 178      59  80.44713
## 179      85  88.03475
## 180      68  80.82213
## 181      66  83.47961
## 182      43  72.80089
## 183      72  80.37391
## 184      64  79.72336
## 185      76  78.69135
## 186      80  85.57211
## 187      92  83.20429
## 188      65  89.48194
## 189      72  96.68228
## 190      81  87.75298
## 191      83  91.87313
## 192      76  86.26651
## 193      81  77.82142
## 194      89  76.04487
## 195      82  84.88891
## 196      65  81.55068
## 197      88  83.70493
## 198      79  80.38773
## 199      67 101.92399
## 200      62  81.10336
## 201      72  73.46680
## 202      60  93.36972
```

```
## 203      94  86.05010
## 204      79  85.59259
## 205      97  73.88114
## 206      75  74.83380
## 207      84  64.08681
## 208      85  65.84094
## 209      74  71.98560
## 210      93  79.13560
## 211     105  85.15069
## 212      86  84.82851
## 213      88  88.64454
## 214      94  87.36139
## 215     110  73.46671
## 216      97  73.32881
## 217     104  76.75376
## 218      98  80.18206
## 219      88  75.03975
## 220      86  75.66433
## 221      85  70.51948
## 222      88  74.54246
## 223      88  72.27715
## 224      92  70.35686
## 225      93  72.18585
## 226      90  79.91554
## 227      83  88.74324
## 228      80  87.01201
## 229      74  85.96054
## 230      66  89.84882
## 231      86  93.28619
## 232      87  91.12115
## 233      87  94.67752
## 234      68  73.79792
## 235      68  80.75316
## 236      69  76.99427
## 237      71  87.39336
## 238      80  77.79257
## 239      91  75.87232
## 240      97  87.87867
## 241      77  79.02455
## 242      69  79.53593
## 243      74  77.39581
## 244      89  71.37449
## 245      67  85.39949
## 246      92  78.33072
## 247      53  86.04030
## 248      56  79.55776
## 249      62  82.75494
## 250      64  86.19625
## 251      66  87.67022
## 252      91  85.00793
## 253      85  80.45297
## 254      61  87.60385
## 255      60  82.44495
## 256      91  77.50487
```

```
## 257      69  79.20244
## 258      85  69.31674
## 259      85  74.80593
## 260      51  83.14847
## 261      50  91.63208
## 262      66  93.48635
## 263      90  84.53632
## 264      98  86.35405
## 265      92  86.94776
## 266     101  82.27458
## 267      91  80.34078
## 268      75  97.84021
## 269      66  90.77786
## 270      83  86.29480
## 271      59  89.59173
## 272      99  81.45834
## 273      54  79.98754
## 274      84  76.10745
## 275      75  78.12297
## 276     104  76.91669
## 277      99  74.90087
## 278      92  87.24509
## 279      93  95.14666
## 280      80  90.93669
## 281      89  90.70110
## 282      84  81.29394
## 283      99  72.64682
## 284     103  82.98819
## 285      91  75.16897
## 286      87  78.10723
## 287      89  82.14880
## 288      86  84.91555
## 289     114  97.32176
## 290     104 101.10846
## 291      90  84.72247
## 292      87  75.83547
## 293      87  76.87742
## 294     108  84.10901
## 295      74  89.08411
## 296     112  81.50902
## 297     107  81.45233
## 298      63  78.28904
## 299      57  74.25636
## 300      52  75.53547
## 301      88  78.88184
## 302      74  77.90593
## 303      54  81.68807
## 304      77  76.22767
## 305      75  71.70319
## 306     102  74.06627
## 307      77  74.02593
## 308      99  79.72855
## 309      96  78.32084
## 310      85  75.75259
```

```
## 311      88  83.80749
## 312      96  79.14016
## 313      59  77.77473
## 314      83  77.15735
## 315      84  76.99388
## 316      94  68.85761
## 317      78  74.32405
## 318      55  76.06391
## 319      65  76.38482
## 320      53  77.40995
## 321      45  77.07957
## 322      67  81.59777
## 323      65  73.66385
## 324      67  79.18950
## 325      49  77.63615
## 326      92  79.20131
## 327      63  79.58513
## 328      67  78.45513
## 329      61  85.18691
## 330      71  87.36482
## 331      86  81.06456
## 332      81  86.55401
## 333      75  76.85508
## 334      78  80.08373
## 335      70  80.76698
## 336      67  78.96028
## 337      75  81.18968
## 338      77  84.35796
## 339      74  78.91002
## 340      92  75.96436
## 341     105  85.42552
## 342      92  80.55205
## 343     103  84.56468
## 344      85  82.79470
## 345      68  76.50017
## 346      84  78.80238
## 347     101  81.74452
## 348      84  81.81584
## 349      79  83.28464
## 350      82  75.40413
## 351      73  74.86536
## 352      65  75.32920
## 353      87  70.05289
## 354      75  72.91093
## 355      53  73.69728
## 356      81  84.47503
## 357      97  72.99975
## 358     100  83.43029
## 359      96  87.95245
## 360      72  78.31703
## 361      67  78.64492
## 362      52  81.71404
## 363      61  80.02600
## 364      71  77.59988
```

```
## 365      69  79.43621
## 366      81  79.28112
## 367      89  83.90819
## 368      75  93.54258
## 369      91  86.40102
## 370      76  82.58538
## 371      98  85.89380
## 372      66  79.81702
## 373      64  76.95876
## 374      57  70.17323
## 375      68  74.13669
## 376      82  72.30131
## 377      93  75.25157
## 378      69  76.54545
## 379      78  78.41872
## 380      92  81.53011
## 381      92  85.93305
## 382      97  83.95041
## 383      91  88.37161
## 384      58  86.31066
## 385      64  82.44512
## 386      85  83.91708
## 387      82  86.67909
## 388      77  87.44296
## 389      70  82.21262
## 390      84  76.52642
## 391      87  75.16841
## 392      88  64.24604
## 393      90  72.58621
## 394      72  77.46501
## 395      74  72.80068
## 396      87  78.05726
## 397      85  80.25288
## 398      77  76.11026
## 399      68  90.32384
## 400      90  74.10965
## 401      88  79.40059
## 402      86  70.55224
## 403      90  86.59573
## 404     101  87.70315
## 405      75  83.05623
## 406      56  78.62248
## 407      52  75.43148
## 408      84  83.51626
## 409      82  74.53313
## 410     100  75.06487
## 411     101  86.07172
## 412      92  87.84534
## 413      76  94.14527
## 414      89  96.04200
## 415     112  90.90022
## 416     102  86.34220
## 417      85  94.79885
## 418      80  86.76487
```

```
## 419      75  87.56303
## 420      93  85.09908
## 421      93  88.22749
## 422      87  93.78125
## 423      76  90.14901
## 424      90  97.39963
## 425      78  88.67552
## 426      72  87.20702
## 427      70  80.27788
## 428      73  85.93653
## 429      95  74.39622
## 430      85  74.75851
## 431      69  72.61288
## 432      69  90.56274
## 433      63  89.47333
## 434      70  78.69775
## 435      62  77.35048
## 436      86  79.53762
## 437      64  83.39287
## 438      79  83.62948
## 439      70  76.57712
## 440      77  77.10248
## 441      83  83.34904
## 442      77  78.56847
## 443      53  69.57909
## 444      86  73.96331
## 445      96  63.06532
## 446      96  69.99296
## 447      76  76.64262
## 448      84  79.98629
## 449      87  73.50146
## 450      55  77.92409
## 451      96  82.13105
## 452      88  77.90857
## 453      68  72.95778
## 454      68  69.73774
## 455      83  69.46233
## 456      75  67.52634
## 457      85  72.19872
## 458      76  78.59388
## 459      81  73.35113
## 460      92  71.86038
## 461      80  75.50214
## 462      85  70.66007
## 463      92  73.52882
## 464     100  88.14247
## 465      85  88.25647
## 466      98  81.56281
## 467      53  85.11315
## 468      63  80.24222
## 469      56  81.24244
## 470      47  82.69428
## 471      66  73.51443
## 472      53  75.48643
```

```
## 473      74  88.59721
## 474      67  74.75932
## 475      40  69.56123
## 476      96  88.86014
## 477      80  71.28452
## 478      94  86.22500
## 479      84  88.95228
## 480      88  92.69140
## 481      86 101.10893
## 482      88  93.85807
## 483      67  84.70288
## 484      70  82.88195
## 485      67  84.57159
## 486      66  73.01808
## 487     103  78.50309
## 488      76  87.77309
## 489      69  90.41206
## 490      48  81.01700
## 491      83  81.18505
## 492      77  78.76486
## 493      78  79.22418
## 494      86  89.33282
## 495      59  81.16533
## 496      45  79.74078
## 497      77  92.39093
## 498      73  89.28796
## 499      86  77.12440
## 500      98  79.77105
## 501     108  75.68524
## 502     104  87.43997
## 503      97  76.52938
## 504      92  84.78700
## 505      88  80.50877
## 506     104  77.35972
## 507     100  72.72118
## 508      94  80.34685
## 509      85  72.55936
## 510      84  70.06595
## 511      73  83.25037
## 512      67  80.59516
## 513      88  81.71215
## 514      67  78.13880
## 515      78  82.04003
## 516     104  88.64843
## 517      88  84.59548
## 518      49  90.67361
## 519      94  84.94869
## 520      79  82.43145
## 521      85  77.98885
## 522      78  72.89823
## 523      79  83.43799
## 524      75  86.92602
## 525      68  87.18417
## 526      83  77.09404
```

```
## 527      84  89.40353
## 528      73  93.34790
## 529      76  86.96778
## 530      97  82.14507
## 531      85  83.43017
## 532      78  75.10548
## 533      92  74.03722
## 534      80  74.16809
## 535      98  94.69567
## 536     104  85.56016
## 537      98  93.39981
## 538      93  85.25712
## 539     104  87.29422
## 540      95  87.46496
## 541      95  83.61357
## 542      87  79.45141
## 543      67  79.45387
## 544      76  78.43478
## 545      88  73.38522
## 546      92  76.03883
## 547      75  72.47713
## 548      79  66.23869
## 549      76  67.67506
## 550      77  74.29160
## 551      93  76.15938
## 552      66  68.53339
## 553      92  64.18492
## 554      71  76.55550
## 555      98  72.31920
## 556      94  75.70448
## 557      94  86.25488
## 558     105  87.13625
## 559      79  85.66963
## 560      87  75.01403
## 561      71  72.48383
## 562      89  76.99319
## 563      77  67.65273
## 564      96  78.88975
## 565      86  85.81330
## 566      98  99.38599
## 567      56 102.30433
## 568      81  93.87881
## 569      87 100.37487
## 570      99  98.32826
## 571      72  89.11290
## 572      77  78.88624
## 573      94  73.97117
## 574      77  86.87866
## 575      80  82.38174
## 576      83  82.41774
## 577      90  70.81168
## 578      69  94.27943
## 579      70  96.56763
## 580      68  96.17381
```

```
## 581      82  83.56929
## 582     111  91.26643
## 583      79  93.49057
## 584      56  95.36925
## 585      94  80.85944
## 586      73  84.39534
## 587      80  82.97421
## 588      77  77.45613
## 589     102  82.72740
## 590      93  77.65384
## 591      61  67.80243
## 592      84  66.62810
## 593      85  74.11153
## 594      78  75.35735
## 595      80  91.41911
## 596      85  89.37015
## 597      92  82.88452
## 598      93  82.28763
## 599      71  84.44431
## 600      82  77.11652
## 601      80  89.87563
## 602      79  87.62389
## 603      81  93.76432
## 604      60  87.78726
## 605      75  93.25463
## 606      77  97.35486
## 607      73  92.79773
## 608      77  98.47135
## 609      80  94.72234
## 610      21  93.80173
## 611      87  96.01043
## 612      82  82.86228
## 613      73  76.13348
## 614      74  77.17624
## 615      67  77.23491
## 616      76  84.72118
## 617      66  84.54481
## 618      95  91.45792
## 619      92  81.48643
## 620      93  76.05063
## 621      64  75.66432
## 622      87  77.26623
## 623      94  77.11791
## 624      81  85.23634
## 625     101  94.58567
## 626      79  82.27350
## 627      88  76.21672
## 628      79  80.88749
## 629      74  77.09902
## 630      86  79.78842
## 631      83  80.55472
## 632      85  78.32830
## 633      59  74.66971
## 634      85  70.44247
```

```
## 635      68  75.76856
## 636      66  81.25690
## 637      43  85.21570
## 638      72  87.45149
## 639      64  83.24715
## 640      76  85.12337
## 641      80  90.08131
## 642      92  85.69963
## 643      65  91.72563
## 644      72  95.57921
## 645      81  80.82969
## 646      83  76.76201
## 647      76  84.40290
## 648      81  88.69413
## 649      89  83.20373
## 650      82  92.68599
## 651      65  82.30209
## 652      88  79.07586
## 653      79  86.73771
## 654      67  81.29509
## 655      62  66.71945
## 656      72  73.26650
## 657      60  86.38450
## 658      94  80.74925
## 659      79  72.12318
## 660      97  86.47275
## 661      75  79.04706
## 662      84  82.85552
## 663      85  74.31326
## 664      74  75.84208
## 665      93  84.08627
## 666     105  79.05523
## 667      86  79.31311
## 668      88  80.23662
## 669      94  75.09644
## 670     110  61.44278
## 671      97  60.04693
## 672     104  73.69484
## 673      98  75.61348
## 674      88  75.09419
## 675      86  60.58337
## 676      85  79.31909
## 677      88  85.07393
## 678      88  71.05731
## 679      92  82.34659
## 680      93  75.70126
## 681      90  76.36214
## 682      83  79.97192
## 683      80  79.96235
## 684      74  77.60617
## 685      66  75.63627
## 686      86  72.69408
## 687      87  78.08013
## 688      87  77.26828
```

```
## 689      68  81.21731
## 690      68  75.48051
## 691      69  71.40040
## 692      71  74.64094
## 693      80  78.67222
## 694      91  88.47661
## 695      97  82.65871
## 696      77  89.85068
## 697      69  97.41705
## 698      74  84.06305
## 699      89  83.32637
## 700      67  79.06699
## 701      92  81.32051
## 702      53  81.98676
## 703      56  72.57010
## 704      62  73.44789
## 705      64  74.81811
## 706      66  74.98640
## 707      91  84.21973
## 708      85  84.56636
## 709      61  82.75465
## 710      60  77.97886
## 711      91  84.76507
## 712      69  79.18286
## 713      85  82.33910
## 714      85  78.44898
## 715      51  81.50181
## 716      50  84.18849
## 717      66  80.75256
## 718      90  87.77617
## 719      98  79.01648
## 720      92  84.53152
## 721     101  73.76380
## 722      91  74.64648
## 723      75  97.05403
## 724      66  95.97254
## 725      83  77.70426
## 726      59  91.18758
## 727      99  75.73231
## 728      54  73.62204
## 729      84  65.45411
## 730      75  78.44786
## 731     104  84.83279
## 732      99  85.78446
## 733      92  82.58456
## 734      93  80.45664
## 735      80  74.78727
## 736      89  70.95567
## 737      84  73.82233
## 738      99  84.23530
## 739     103  79.76443
## 740      91  75.55942
## 741      87  83.90318
## 742      89  78.55669
```

```
## 743      86  80.46496
## 744     114  76.10448
## 745     104  77.89772
## 746      90  71.32829
## 747      87  73.19595
## 748      87  84.23038
## 749     108  80.71645
## 750      74  82.88104
## 751     112  86.58766
## 752     107  77.59884
## 753      63  76.13671
## 754      57  87.43058
## 755      52  86.48412
## 756      88  90.45556
## 757      74  86.15989
## 758      54  97.07526
## 759      77  95.61839
## 760      75  91.30071
## 761     102 102.70882
## 762      77  94.41229
## 763      99  97.02811
## 764      96  89.96791
## 765      85  82.18003
## 766      88  83.80303
## 767      96  85.00570
## 768      59  80.53210
## 769      83  87.92249
## 770      84  94.71954
## 771      94  73.29979
## 772      78  71.60731
## 773      55  67.21755
## 774      65  61.79597
## 775      53  75.69141
## 776      45  84.94305
## 777      67  76.45704
## 778      65  76.87543
## 779      67  79.08576
## 780      49  82.82991
## 781      92  77.53815
## 782      63  88.68762
## 783      67  86.93289
## 784      61  86.40456
## 785      71  85.95538
## 786      86  71.33692
## 787      81  81.90683
## 788      75  74.62795
## 789      78  72.43715
## 790      70  68.46631
## 791      67  80.61536
## 792      75  77.72868
## 793      77  78.76628
## 794      74  79.50862
## 795      92  83.21869
## 796     105  78.19842
```

```
## 797      92  89.64355
## 798     103  89.13728
## 799      85  82.46795
## 800      68  77.71627
## 801      84  68.72122
## 802     101  73.81261
## 803      84  90.70552
## 804      79  88.53050
## 805      82  93.36524
## 806      73  81.87482
## 807      65  69.53519
## 808      87  65.83305
## 809      75  80.91972
## 810      53  75.66819
## 811      81  69.78134
## 812      97  74.65439
## 813     100  86.20933
## 814      96  90.52296
## 815      72  89.88804
## 816      67  91.22499
## 817      52  85.30952
## 818      61  76.50755
## 819      71  77.81370
## 820      69  76.53768
## 821      81  80.85762
## 822      89  83.55077
## 823      75  86.28180
## 824      91  90.36860
## 825      76  76.09565
## 826      98  75.87291
## 827      66  87.02377
## 828      64  78.48445
## 829      57  76.31062
## 830      68  73.67242
## 831      82  82.00670
## 832      93  76.92192
## 833      69  77.12030
## 834      78  75.08059
## 835      92  87.08930
## 836      92  84.10366
## 837      97  69.87058
## 838      91  67.95321
## 839      58  76.27043
## 840      64  77.82358
## 841      85  61.48364
## 842      82  73.93711
## 843      77  80.09102
## 844      70  80.39881
## 845      84  77.94942
## 846      87  79.27261
## 847      88  80.39794
## 848      90  79.57612
## 849      72  88.63689
## 850      74  85.94038
```

```
## 851      87  83.89021
## 852      85  84.32137
## 853      77  88.01176
## 854      68  89.30777
## 855      90  84.21572
## 856      88  89.72324
## 857      86  88.68351
## 858      90  93.65513
## 859     101  86.63118
## 860      75  80.90752
## 861      56  83.25118
## 862      52  82.34711
## 863      84  83.24225
## 864      82  77.00634
## 865     100  79.45941
## 866     101  70.73381
## 867      92  76.64514
## 868      76  80.04985
## 869      89  77.07574
## 870     112  81.67504
## 871     102  74.98745
## 872      85  84.02452
## 873      80  72.21179
## 874      75  77.78425
## 875      93  81.11352
## 876      93  82.50825
## 877      87  76.61177
## 878      76  74.25842
## 879      90  69.83767
## 880      78  78.11892
## 881      72  75.86576
## 882      70  70.01321
## 883      73  82.94723
## 884      95  80.03326
## 885      85  82.57406
## 886      69  87.68125
## 887      69  74.67282
## 888      63  67.15518
## 889      70  76.66567
## 890      62  77.28677
## 891      86  73.35303
## 892      64  85.23277
## 893      79  82.52631
## 894      70  80.73811
## 895      77  74.93185
## 896      83  84.73855
## 897      77  75.75555
## 898      53  81.92132
## 899      86  87.57509
## 900      96  81.64124
## 901      96  78.23981
## 902      76  79.85345
## 903      84  83.07927
## 904      87  77.58223
```

```
## 905      55  83.27506
## 906      96  84.80112
## 907      88  77.35284
## 908      68  83.92642
## 909      68  80.88043
## 910      83  79.19680
## 911      75  61.31942
## 912      85  58.38833
## 913      76  66.35593
## 914      81  60.64071
## 915      92  59.60618
## 916      80  71.52196
## 917      85  82.97232
## 918      92  69.80963
## 919     100  73.80638
## 920      85  71.90841
## 921      98  76.54061
## 922      53  81.68224
## 923      63  85.81771
## 924      56  92.23216
## 925      47  84.05760
## 926      66  79.86707
## 927      53  83.63096
## 928      74  72.09129
## 929      67  76.10990
## 930      40  76.51728
## 931      96  79.08606
## 932      80  75.26965
## 933      94  92.94321
## 934      84  85.87283
## 935      88  73.31166
## 936      86  77.15060
## 937      88  70.75125
## 938      67  83.97144
## 939      70  95.72627
## 940      67  77.69374
## 941      66  79.19794
## 942     103  76.35351
## 943      76  78.32832
## 944      69  75.91017
## 945      48  70.76003
## 946      83  86.96236
## 947      77  78.23717
## 948      78  81.18502
## 949      86  75.78016
## 950      59  75.92134
## 951      45  77.63943
## 952      77  97.27400
## 953      73  85.27491
## 954      86  88.53154
## 955      98  93.95050
## 956     108 104.40963
## 957     104  93.08013
## 958      97  94.25311
```

```
## 959      92  96.39318
## 960      88 102.13149
## 961     104  93.15151
## 962     100  93.26443
## 963      94  88.38960
## 964      85  83.12927
## 965      84  87.16569
## 966      73  86.20118
## 967      67  84.48924
## 968      88  93.72888
## 969      67 100.22576
## 970      78  90.75092
## 971     104  85.92259
## 972      88  90.08918
## 973      49  88.66595
## 974      94  94.18213
## 975      79  85.07843
## 976      85  87.16259
## 977      78  76.98871
## 978      79  66.83786
## 979      75  71.05327
## 980      68  80.22158
## 981      83  77.96724
## 982      84  74.09699
## 983      73  75.16087
## 984      76  75.85082
## 985      97  78.57326
## 986      85  86.20376
## 987      78  90.85388
## 988      92  82.07620
## 989      80  92.37084
## 990      98  84.76236
## 991     104  86.21000
## 992      98  82.30628
## 993      93  93.02867
## 994     104  92.67147
## 995      95  87.26267
## 996      95  85.61029
## 997      87  81.43227
## 998      67  78.74050
## 999      76  85.55504
## 1000     88  85.67697
## 1001     92  89.49447
## 1002     75  89.33847
## 1003     79  97.19323
## 1004     76  94.41711
## 1005     77  90.90383
## 1006     93  89.17507
## 1007     66  90.21799
## 1008     92  93.50255
## 1009     71  93.87559
## 1010     98  95.81350
## 1011     94  90.93701
## 1012     94  92.44312
```

```
## 1013    105  78.34708
## 1014     79  72.71619
## 1015     87  76.46792
## 1016     71  74.59291
## 1017     89  72.01169
## 1018     77  63.68687
## 1019     96  80.15956
## 1020     86  87.11331
## 1021     98  92.71264
## 1022     56  72.71623
## 1023     81  73.56595
## 1024     87  69.61434
## 1025     99  73.36825
## 1026     72  79.86398
## 1027     77  79.38112
## 1028     94  76.32710
## 1029     77  87.36658
## 1030     80  87.46441
## 1031     83  94.51520
## 1032     90  93.38030
## 1033     69  91.96627
## 1034     70  89.23678
## 1035     68  82.81230
## 1036     82  80.88571
## 1037    111  87.73838
## 1038     79  86.40998
## 1039     56  81.97909
## 1040     94  82.52483
## 1041     73  68.20531
## 1042     80  66.22217
## 1043     77  70.54708
## 1044    102  67.31787
## 1045     93  70.66388
## 1046     61  84.08191
## 1047     84  85.42203
## 1048     85  72.46110
## 1049     78  76.83792
## 1050     80  75.06820
## 1051     85  75.85712
## 1052     92  76.59533
## 1053     93  70.17289
## 1054     71  68.75619
## 1055     82  73.16580
## 1056     80  82.69221
## 1057     79  78.45319
## 1058     81  74.46707
## 1059     60  84.53133
## 1060     75  72.54941
## 1061     77  85.11678
## 1062     73  83.35557
## 1063     77  82.59932
## 1064     80  83.53339
## 1065     21  86.85838
## 1066     87  84.46278
```

```
## 1067     82  88.67685
## 1068     73  83.37239
## 1069     74  88.58651
## 1070     67  89.59870
## 1071     76  84.64656
## 1072     66  81.64766
## 1073     95  92.61935
## 1074     92  86.27243
## 1075     93  82.85143
## 1076     64  85.66203
## 1077     87  95.84800
## 1078     94  90.95834
## 1079     81  95.81290
## 1080    101  98.69009
## 1081     79  85.71811
## 1082     88  69.76796
## 1083     79  75.60889
## 1084     74  71.34770
## 1085     86  72.41132
## 1086     83  72.15901
## 1087     85  71.27513
## 1088     59  84.49067
## 1089     85  82.25327
## 1090     68  81.79879
## 1091     66  85.53387
## 1092     43  80.98372
## 1093     72  73.90174
## 1094     64  73.92732
## 1095     76  71.53401
## 1096     80  74.23172
## 1097     92  81.61689
## 1098     65  82.79164
## 1099     72  77.45100
## 1100     81  76.58674
## 1101     83  78.73276
## 1102     76  96.77330
## 1103     81  87.60659
## 1104     89  76.93790
## 1105     82  87.38424
## 1106     65  72.88084
## 1107     88  80.21365
## 1108     79  79.76677
## 1109     67  66.28857
## 1110     62  69.89638
## 1111     72  67.14770
## 1112     60  70.31318
## 1113     94  63.48091
## 1114     79  71.00934
## 1115     97  70.69567
## 1116     75  70.69689
## 1117     84  73.50918
## 1118     85  78.85426
## 1119     74  83.60694
## 1120     93  83.10800
```

```
## 1121    105  80.02899
## 1122     86  71.49187
## 1123     88  66.74572
## 1124     94  73.57565
## 1125    110  80.18532
## 1126     97  72.35786
## 1127    104  72.86273
## 1128     98  71.97395
## 1129     88  65.32023
## 1130     86  73.37091
## 1131     85  70.95516
## 1132     88  67.70542
## 1133     88  71.73850
## 1134     92  76.98511
## 1135     93  84.88781
## 1136     90  94.03477
## 1137     83  83.22815
## 1138     80  84.15055
## 1139     74  89.26872
## 1140     66  79.14093
## 1141     86  84.17269
## 1142     87  84.81615
## 1143     87  72.06287
## 1144     68  77.75736
## 1145     68  78.77255
## 1146     69  72.63020
## 1147     71  78.03885
## 1148     80  90.17251
## 1149     91  78.48975
## 1150     97  79.38583
## 1151     77  77.11963
## 1152     69  90.17322
## 1153     74  81.64387
## 1154     89  83.99443
## 1155     67  85.64188
## 1156     92  82.44665
## 1157     53  95.51429
## 1158     56  77.02829
## 1159     62  89.46990
## 1160     64  94.55289
## 1161     66  95.36864
## 1162     91  91.57210
## 1163     85  80.87037
## 1164     61  82.79777
## 1165     60  75.87495
## 1166     91  80.97836
## 1167     69  84.59811
## 1168     85  81.97436
## 1169     85  94.32579
## 1170     51  85.66276
## 1171     50  74.16725
## 1172     66  77.35892
## 1173     90  71.56327
## 1174     98  89.51002
```

```
## 1175     92  83.62805
## 1176    101  96.01226
## 1177     91  99.63880
## 1178     75  86.66543
## 1179     66  85.08412
## 1180     83  88.86202
## 1181     59  96.16164
## 1182     99  94.35191
## 1183     54  77.78130
## 1184     84  79.97524
## 1185     75  76.64023
## 1186    104  81.35226
## 1187     99  82.58319
## 1188     92  83.45274
## 1189     93  80.12935
## 1190     80  82.74097
## 1191     89  78.38666
## 1192     84  85.76137
## 1193     99  80.03589
## 1194    103  75.26520
## 1195     91  79.94641
## 1196     87  86.44159
## 1197     89  84.42915
## 1198     86  76.77597
## 1199    114  85.49227
## 1200    104  85.43649
## 1201     90  82.01907
## 1202     87  85.49127
## 1203     87  76.48413
## 1204    108  72.47560
## 1205     74  74.71764
## 1206    112  79.17754
## 1207    107  75.89540
## 1208     63  72.52410
## 1209     57  72.99217
## 1210     52  73.95031
## 1211     88  80.28342
## 1212     74  76.84542
## 1213     54  72.66814
## 1214     77  79.43934
## 1215     75  84.21485
## 1216    102  76.13103
## 1217     77  75.85239
## 1218     99  80.34523
## 1219     96  81.40989
## 1220     85  89.32086
## 1221     88  71.65314
## 1222     96  83.47910
## 1223     59  79.31333
## 1224     83  83.99140
## 1225     84  86.04019
## 1226     94  80.58368
## 1227     78  78.55305
## 1228     55  81.19871
```

```
## 1229     65  80.75316
## 1230     53  83.34638
## 1231     45  72.71053
## 1232     67  83.05962
## 1233     65  75.30919
## 1234     67  61.84184
## 1235     49  75.49493
## 1236     92  66.04304
## 1237     63  67.75688
## 1238     67  69.12003
## 1239     61  71.72482
## 1240     71  69.96031
## 1241     86  70.90493
## 1242     81  82.19781
## 1243     75  78.83265
## 1244     78  79.97341
## 1245     70  73.89507
## 1246     67  74.58934
## 1247     75  77.85825
## 1248     77  80.11585
## 1249     74  72.59577
## 1250     92  73.24053
## 1251    105  74.94112
## 1252     92  80.93204
## 1253    103  82.70670
## 1254     85  83.81551
## 1255     68  81.01255
## 1256     84  83.63764
## 1257    101  83.91016
## 1258     84  75.41719
## 1259     79  77.28019
## 1260     82  85.05006
## 1261     73  76.05888
## 1262     65  78.48064
## 1263     87  86.08551
## 1264     75  75.94236
## 1265     53  79.40133
## 1266     81  84.21428
## 1267     97  76.93622
## 1268    100  76.13965
## 1269     96  80.08179
## 1270     72  80.51164
## 1271     67  78.57486
## 1272     52  82.35286
## 1273     61  91.28096
## 1274     71  96.60549
## 1275     69  94.87582
## 1276     81  92.31089
## 1277     89  94.05118
## 1278     75  94.20321
## 1279     91  96.22611
## 1280     76  87.91910
## 1281     98  83.99419
## 1282     66  78.77420
```

```
## 1283     64  75.09218
## 1284     57  82.55089
## 1285     68  71.98056
## 1286     82  69.20131
## 1287     93  77.96775
## 1288     69  87.78023
## 1289     78  86.35932
## 1290     92  82.21422
## 1291     92  79.65022
## 1292     97  75.35074
## 1293     91  81.20123
## 1294     58  90.70037
## 1295     64  84.24705
## 1296     85  78.67993
## 1297     82  87.97745
## 1298     77  87.69785
## 1299     70  92.90616
## 1300     84  96.57243
## 1301     87  85.69885
## 1302     88  78.76782
## 1303     90  73.07410
## 1304     72  77.54960
## 1305     74  83.26963
## 1306     87  78.98654
## 1307     85  79.07793
## 1308     77  80.37722
## 1309     68  79.71387
## 1310     90  78.50491
## 1311     88  76.75080
## 1312     86  80.27969
## 1313     90  73.46866
## 1314    101  79.25616
## 1315     75  82.00955
## 1316     56  79.89839
## 1317     52  91.02800
## 1318     84  88.08337
## 1319     82  88.85369
## 1320    100  94.45356
## 1321    101  84.35462
## 1322     92  78.05740
## 1323     76  83.95127
## 1324     89  86.33623
## 1325    112  82.90238
## 1326    102  81.67246
## 1327     85  85.51611
## 1328     80  80.18255
## 1329     75  79.54560
## 1330     93  75.54402
## 1331     93  73.65885
## 1332     87  71.01422
## 1333     76  81.45969
## 1334     90  82.86082
## 1335     78  80.62995
## 1336     72  88.25696
```

```
## 1337     70  76.46766
## 1338     73  80.34835
## 1339     95  72.91620
## 1340     85  76.90302
## 1341     69  83.18826
## 1342     69  70.94189
## 1343     63  76.87526
## 1344     70  80.45797
## 1345     62  69.39293
## 1346     86  84.77448
## 1347     64  85.62956
## 1348     79  83.98628
## 1349     70  80.12358
## 1350     77  86.59676
## 1351     83  87.14948
## 1352     77  89.34894
## 1353     53  96.16870
## 1354     86  89.87599
## 1355     96  82.54057
## 1356     96  72.35155
## 1357     76  75.48538
## 1358     84  89.39629
## 1359     87  90.79489
## 1360     55  72.17537
## 1361     96  98.82305
## 1362     88  91.07166
## 1363     68  74.70489
## 1364     68  74.08408
## 1365     83  75.45591
## 1366     75  75.76548
## 1367     85  67.74708
## 1368     76  63.73702
## 1369     81  63.09795
## 1370     92  79.36250
## 1371     80  85.89266
## 1372     85  88.93737
## 1373     92  86.42517
## 1374    100  79.24316
## 1375     85  80.67956
## 1376     98  74.32748
## 1377     53  77.51726
## 1378     63  74.68399
## 1379     56  90.58364
## 1380     47  84.36898
## 1381     66  89.61974
## 1382     53  87.27046
## 1383     74  91.72583
## 1384     67  90.46183
## 1385     40  92.84914
## 1386     96  83.81388
## 1387     80  78.44059
## 1388     94  78.52910
## 1389     84  84.60661
## 1390     88  80.35381
```

```
## 1391     86  81.01220
## 1392     88  81.58834
## 1393     67  83.27561
## 1394     70  83.42052
## 1395     67  88.09058
## 1396     66  80.71739
## 1397    103  82.57078
## 1398     76  79.27823
## 1399     69  79.45690
## 1400     48  87.45363
## 1401     83  82.67362
## 1402     77  83.26842
## 1403     78  79.96190
## 1404     86  83.17930
## 1405     59  81.28235
## 1406     45  88.91157
## 1407     77  79.89535
## 1408     73  82.59148
## 1409     86  83.88969
## 1410     98  79.11848
## 1411    108  80.02279
## 1412    104  79.38053
## 1413     97  79.62617
## 1414     92  81.86965
## 1415     88  81.52487
## 1416    104  75.59362
## 1417    100  76.78916
## 1418     94  74.30279
## 1419     85  76.30224
## 1420     84  69.65634
## 1421     73  74.26352
## 1422     67  83.82491
## 1423     88  84.75257
## 1424     67  73.73032
## 1425     78  75.06767
## 1426    104  84.63083
## 1427     88  78.97639
## 1428     49  76.45999
## 1429     94  71.29329
## 1430     79  81.13600
## 1431     85  80.70558
## 1432     78  78.14821
## 1433     79  71.04719
## 1434     75  80.36170
## 1435     68  81.90376
## 1436     83  93.33645
## 1437     84  88.19413
## 1438     73  90.25974
## 1439     76  84.66770
## 1440     97  82.96837
## 1441     85  89.12204
## 1442     78  87.95702
## 1443     92  82.57955
## 1444     80  82.70475
```

```
## 1445     98  79.25230
## 1446    104  80.20210
## 1447     98  71.72168
## 1448     93  74.42225
## 1449    104  79.63366
## 1450     95  81.94346
## 1451     95  82.75808
## 1452     87  80.78277
## 1453     67  76.62972
## 1454     76  77.85689
## 1455     88  68.36665
## 1456     92  69.11303
## 1457     75  66.24626
## 1458     79  81.84655
## 1459     76  78.35787
## 1460     77  69.94233
## 1461     93  75.76441
## 1462     66  81.02432
## 1463     92  74.52049
## 1464     71  81.01784
## 1465     98  71.46302
## 1466     94  75.61764
## 1467     94  78.34152
## 1468    105  77.83881
## 1469     79  77.22750
## 1470     87  86.79195
## 1471     71  77.71251
## 1472     89  82.62507
## 1473     77  81.35461
## 1474     96  89.93796
## 1475     86  96.51608
## 1476     98  86.16285
## 1477     56  91.33961
## 1478     81  86.93554
## 1479     87  84.63384
## 1480     99  72.70103
## 1481     72  71.08248
## 1482     77  91.35231
## 1483     94  86.00672
## 1484     77  91.94352
## 1485     80  85.43193
## 1486     83  83.53370
## 1487     90  85.50160
## 1488     69  80.42214
## 1489     70  87.39467
## 1490     68  93.21013
## 1491     82  83.54572
## 1492    111  77.31713
## 1493     79  89.01612
## 1494     56  89.92811
## 1495     94  76.09057
## 1496     73  77.41702
## 1497     80  85.67225
## 1498     77  87.78315
```

```
## 1499    102  71.42951
## 1500     93  78.50359
## 1501     61  70.81870
## 1502     84  80.91072
## 1503     85  75.32227
## 1504     78  68.16678
## 1505     80  79.64992
## 1506     85  71.17387
## 1507     92  79.90700
## 1508     93  84.00078
## 1509     71  82.16401
## 1510     82  77.16445
## 1511     80  76.00898
## 1512     79  78.13104
## 1513     81  72.61470
## 1514     60  77.78654
## 1515     75  77.61568
## 1516     77  72.10171
## 1517     73  81.84125
## 1518     77  77.74045
## 1519     80  78.32464
## 1520     21  67.84336
## 1521     87  80.92785
```

```
mean (apply(predicted_values, 1, min)/apply(predicted_values, 1, max)) # calculate accuracy
```

```
## [1] 0.8580047
```

The prediction accuracy here is at 85.85%

```
predicted <- predict(Model5, newx = test_baseball)# predict on test data
predicted_values <- cbind (actual=test_baseball$TARGET_WINS, predicted)  # combine
```

```
## Warning in cbind(actual = test_baseball$TARGET_WINS, predicted): number of
## rows of result is not a multiple of vector length (arg 1)
```

```
predicted_values
```

```
##      actual predicted
## 1        75  67.44342
## 2        85  67.45113
## 3        76  70.07451
## 4        81  75.43397
## 5        92  67.72684
## 6        80  67.72587
## 7        85  63.76240
## 8        92  77.57821
## 9       100  87.50168
## 10       85  75.33778
## 11       98  86.57140
## 12       53  76.10532
## 13       63  79.30210
```

```
## 14       56  85.05514
## 15       47  89.33395
## 16       66  86.35917
## 17       53  75.87327
## 18       74  75.27661
## 19       67  76.92580
## 20       40  72.38257
## 21       96  91.24888
## 22       80  81.21417
## 23       94  76.85482
## 24       84  75.73465
## 25       88  92.95906
## 26       86  79.74796
## 27       88  82.59460
## 28       67  82.93132
## 29       70  84.01025
## 30       67  83.67314
## 31       66  74.98318
## 32      103  94.40116
## 33       76  85.35736
## 34       69  86.31272
## 35       48  85.20792
## 36       83  79.86968
## 37       77  71.39335
## 38       78  79.65920
## 39       86  88.48478
## 40       59  92.89590
## 41       45  95.70531
## 42       77  80.15908
## 43       73  77.64883
## 44       86  67.18263
## 45       98  64.94270
## 46      108  62.58915
## 47      104  71.88851
## 48       97  67.42428
## 49       92  66.89260
## 50       88  74.11101
## 51      104  84.19595
## 52      100  77.19389
## 53       94  81.04087
## 54       85  78.77630
## 55       84  78.14143
## 56       73  76.04199
## 57       67  79.46042
## 58       88  73.69400
## 59       67  74.54247
## 60       78  75.74929
## 61      104  74.97222
## 62       88  75.44410
## 63       49  79.02010
## 64       94  69.75831
## 65       79  70.64981
## 66       85  69.12150
## 67       78  74.71038
```

```
## 68       79  69.62466
## 69       75  70.17964
## 70       68  70.25523
## 71       83  69.23863
## 72       84  71.09084
## 73       73  72.67250
## 74       76  67.57287
## 75       97  73.96325
## 76       85  82.29090
## 77       78  76.84995
## 78       92  80.96384
## 79       80  86.00466
## 80       98  80.99199
## 81      104  90.30120
## 82       98  86.68144
## 83       93  76.35323
## 84      104  83.38313
## 85       95  86.70403
## 86       95  88.15920
## 87       87  77.29176
## 88       67  78.74435
## 89       76  78.89828
## 90       88  84.86922
## 91       92  76.52274
## 92       75  68.53117
## 93       79  80.63921
## 94       76  79.04574
## 95       77  91.06636
## 96       93  78.71918
## 97       66  73.94550
## 98       92  72.47404
## 99       71  76.97092
## 100      98  78.30645
## 101      94  77.84933
## 102      94  73.92133
## 103     105  87.30513
## 104      79  76.38672
## 105      87  87.67499
## 106      71  77.46469
## 107      89  82.05258
## 108      77  85.24247
## 109      96  87.16002
## 110      86  83.45556
## 111      98  83.35626
## 112      56  86.73727
## 113      81  91.21914
## 114      87  88.04261
## 115      99  78.51105
## 116      72  81.95540
## 117      77  82.76242
## 118      94  81.70924
## 119      77  85.27418
## 120      80  69.20964
## 121      83  69.37010
```

```
## 122      90  78.11881
## 123      69  69.72151
## 124      70  66.82142
## 125      68  65.28600
## 126      82  71.84567
## 127     111  71.77897
## 128      79  76.57889
## 129      56  79.01450
## 130      94  85.26364
## 131      73  75.40463
## 132      80  82.54799
## 133      77  78.50081
## 134     102  82.48798
## 135      93  84.16834
## 136      61  81.26085
## 137      84  80.88645
## 138      85  75.53196
## 139      78  80.44320
## 140      80  84.92377
## 141      85  75.02694
## 142      92  75.21066
## 143      93  71.99137
## 144      71  71.21597
## 145      82  81.47611
## 146      80  80.90290
## 147      79  80.66912
## 148      81  82.13911
## 149      60  74.68290
## 150      75  73.79684
## 151      77  69.14518
## 152      73  69.07586
## 153      77  74.64032
## 154      80  75.73128
## 155      21  71.89865
## 156      87  78.09027
## 157      82  76.71291
## 158      73  77.55373
## 159      74  77.24322
## 160      67  75.43189
## 161      76  81.07324
## 162      66  75.80984
## 163      95  75.01335
## 164      92  87.32782
## 165      93  87.38260
## 166      64  85.36363
## 167      87  72.33313
## 168      94  88.77234
## 169      81  80.72687
## 170     101  81.51611
## 171      79  79.82869
## 172      88  79.48619
## 173      79  79.26894
## 174      74  88.37324
## 175      86  86.43100
```

```
## 176      83  85.22346
## 177      85  83.65356
## 178      59  80.09002
## 179      85  88.30086
## 180      68  79.69794
## 181      66  84.31400
## 182      43  74.81675
## 183      72  80.47708
## 184      64  79.70448
## 185      76  78.06843
## 186      80  85.14572
## 187      92  82.05263
## 188      65  94.71854
## 189      72  96.65872
## 190      81  87.64550
## 191      83  90.21192
## 192      76  87.23107
## 193      81  78.75851
## 194      89  79.00789
## 195      82  81.25164
## 196      65  81.88150
## 197      88  84.03268
## 198      79  82.40478
## 199      67  95.78563
## 200      62  80.30579
## 201      72  75.01089
## 202      60  95.34271
## 203      94  87.50287
## 204      79  87.19443
## 205      97  76.17791
## 206      75  76.06067
## 207      84  64.91472
## 208      85  65.72480
## 209      74  72.39191
## 210      93  77.65363
## 211     105  84.15721
## 212      86  82.04471
## 213      88  86.08161
## 214      94  87.63339
## 215     110  75.05263
## 216      97  74.98438
## 217     104  83.25936
## 218      98  82.71286
## 219      88  75.53272
## 220      86  76.88389
## 221      85  71.86619
## 222      88  76.06333
## 223      88  74.51141
## 224      92  70.23775
## 225      93  73.52272
## 226      90  79.16581
## 227      83  86.89115
## 228      80  87.13819
## 229      74  85.46662
```

```
## 230      66  89.75357
## 231      86  91.54361
## 232      87  90.59831
## 233      87  95.02468
## 234      68  75.40648
## 235      68  79.71284
## 236      69  75.96297
## 237      71  85.91518
## 238      80  78.85678
## 239      91  76.84806
## 240      97  86.14862
## 241      77  78.05120
## 242      69  77.47512
## 243      74  78.23876
## 244      89  73.50749
## 245      67  86.78095
## 246      92  79.08171
## 247      53  84.72774
## 248      56  78.34657
## 249      62  82.76116
## 250      64  86.34586
## 251      66  87.92397
## 252      91  84.11055
## 253      85  77.99811
## 254      61  85.45186
## 255      60  77.80467
## 256      91  75.25132
## 257      69  76.95882
## 258      85  68.38314
## 259      85  73.50657
## 260      51  94.02116
## 261      50  93.71058
## 262      66  89.96523
## 263      90  80.98936
## 264      98  84.84977
## 265      92  85.19842
## 266     101  79.47680
## 267      91  79.10396
## 268      75  95.64288
## 269      66  86.71087
## 270      83  85.77351
## 271      59  88.98923
## 272      99  80.79937
## 273      54  77.98964
## 274      84  75.28088
## 275      75  77.35333
## 276     104  76.26755
## 277      99  75.33738
## 278      92  88.17904
## 279      93  96.27179
## 280      80  89.92458
## 281      89  92.67221
## 282      84  84.05349
## 283      99  74.83770
```

```
## 284     103  85.95691
## 285      91  78.72426
## 286      87  78.71273
## 287      89  81.26346
## 288      86  85.39316
## 289     114  94.85380
## 290     104  98.58835
## 291      90  82.07467
## 292      87  75.18221
## 293      87  76.37619
## 294     108  82.90378
## 295      74  88.14256
## 296     112  81.52590
## 297     107  82.87604
## 298      63  78.59107
## 299      57  75.43887
## 300      52  75.87870
## 301      88  77.95901
## 302      74  77.42188
## 303      54  79.91021
## 304      77  75.70473
## 305      75  72.30603
## 306     102  74.72546
## 307      77  77.04007
## 308      99  84.57445
## 309      96  79.03250
## 310      85  75.61020
## 311      88  85.12809
## 312      96  83.01153
## 313      59  81.21832
## 314      83  77.58116
## 315      84  76.16731
## 316      94  71.52026
## 317      78  75.22951
## 318      55  75.48431
## 319      65  76.87937
## 320      53  75.93558
## 321      45  75.95209
## 322      67  82.46585
## 323      65  74.48463
## 324      67  77.49127
## 325      49  75.66515
## 326      92  77.13311
## 327      63  79.35974
## 328      67  78.58116
## 329      61  85.79704
## 330      71  87.45633
## 331      86  82.37835
## 332      81  86.72390
## 333      75  75.70592
## 334      78  80.05905
## 335      70  80.56484
## 336      67  81.56194
## 337      75  79.82098
```

```
## 338      77  87.10452
## 339      74  79.65431
## 340      92  76.07465
## 341     105  83.48986
## 342      92  80.22453
## 343     103  82.98359
## 344      85  82.75474
## 345      68  75.27901
## 346      84  79.49034
## 347     101  82.43166
## 348      84  82.29389
## 349      79  84.23164
## 350      82  76.59806
## 351      73  74.96025
## 352      65  76.25467
## 353      87  68.62133
## 354      75  73.26311
## 355      53  75.04697
## 356      81  81.57823
## 357      97  76.97103
## 358     100  84.95317
## 359      96  88.65480
## 360      72  81.29242
## 361      67  80.11348
## 362      52  81.05779
## 363      61  78.21721
## 364      71  78.12708
## 365      69  77.36967
## 366      81  77.29128
## 367      89  81.79287
## 368      75  87.54212
## 369      91  85.17605
## 370      76  81.07061
## 371      98  85.04887
## 372      66  78.68070
## 373      64  77.38031
## 374      57  70.92452
## 375      68  74.03955
## 376      82  71.75593
## 377      93  75.89851
## 378      69  76.24042
## 379      78  78.59571
## 380      92  82.86759
## 381      92  86.95993
## 382      97  84.03568
## 383      91  87.55968
## 384      58  85.98648
## 385      64  82.14632
## 386      85  85.69631
## 387      82  87.07823
## 388      77  86.90617
## 389      70  81.14569
## 390      84  74.98691
## 391      87  74.22010
```

```
## 392      88  66.23871
## 393      90  70.80897
## 394      72  79.02572
## 395      74  74.46659
## 396      87  76.17838
## 397      85  78.85746
## 398      77  75.16575
## 399      68  90.13430
## 400      90  76.43702
## 401      88  80.42516
## 402      86  71.76322
## 403      90  85.62887
## 404     101  88.54009
## 405      75  85.41391
## 406      56  80.87090
## 407      52  76.33751
## 408      84  85.13585
## 409      82  76.30060
## 410     100  72.82338
## 411     101  85.53497
## 412      92  88.16092
## 413      76  94.67713
## 414      89  92.65560
## 415     112  90.99936
## 416     102  85.43587
## 417      85  92.80923
## 418      80  89.13903
## 419      75  87.66494
## 420      93  86.26550
## 421      93  87.30838
## 422      87  95.64263
## 423      76  89.77173
## 424      90  96.58498
## 425      78  89.08766
## 426      72  84.87436
## 427      70  79.09352
## 428      73  83.13129
## 429      95  74.78158
## 430      85  75.01630
## 431      69  75.37997
## 432      69  91.42830
## 433      63  89.53917
## 434      70  80.57424
## 435      62  77.44170
## 436      86  80.56396
## 437      64  86.13667
## 438      79  86.17377
## 439      70  76.58950
## 440      77  76.80901
## 441      83  83.22035
## 442      77  78.71702
## 443      53  68.39029
## 444      86  74.51458
## 445      96  65.97702
```

```
## 446      96  69.94125
## 447      76  78.07337
## 448      84  79.56762
## 449      87  75.47085
## 450      55  78.95959
## 451      96  82.04844
## 452      88  78.93698
## 453      68  75.55269
## 454      68  73.11614
## 455      83  71.16156
## 456      75  68.87201
## 457      85  74.10248
## 458      76  80.17785
## 459      81  77.30115
## 460      92  72.49663
## 461      80  77.63991
## 462      85  72.55387
## 463      92  75.26255
## 464     100  89.56768
## 465      85  89.34414
## 466      98  81.88434
## 467      53  84.87743
## 468      63  81.65693
## 469      56  82.50675
## 470      47  81.01440
## 471      66  75.36897
## 472      53  74.83014
## 473      74  86.95051
## 474      67  75.64894
## 475      40  71.91194
## 476      96  88.80437
## 477      80  72.45152
## 478      94  85.64352
## 479      84  87.18982
## 480      88  88.23807
## 481      86  97.61437
## 482      88  94.15000
## 483      67  85.46822
## 484      70  81.29913
## 485      67  83.24641
## 486      66  73.97669
## 487     103  78.98370
## 488      76  87.86617
## 489      69  90.77691
## 490      48  82.77272
## 491      83  81.84398
## 492      77  78.91267
## 493      78  78.56951
## 494      86  90.81276
## 495      59  82.15841
## 496      45  77.31243
## 497      77  91.95794
## 498      73  87.79896
## 499      86  76.99782
```

```
## 500      98  79.89077
## 501     108  74.67528
## 502     104  86.66090
## 503      97  76.67202
## 504      92  84.36278
## 505      88  82.37365
## 506     104  77.49743
## 507     100  72.37151
## 508      94  75.22012
## 509      85  73.18528
## 510      84  71.40753
## 511      73  78.79972
## 512      67  77.39860
## 513      88  81.06421
## 514      67  78.92919
## 515      78  83.29334
## 516     104  92.13062
## 517      88  86.88210
## 518      49  86.09691
## 519      94  83.65993
## 520      79  80.00122
## 521      85  75.79357
## 522      78  71.29807
## 523      79  80.14395
## 524      75  85.04878
## 525      68  86.42506
## 526      83  77.88073
## 527      84  89.99265
## 528      73  93.12574
## 529      76  86.35583
## 530      97  82.66900
## 531      85  86.55700
## 532      78  76.00762
## 533      92  75.10227
## 534      80  76.24862
## 535      98  93.95561
## 536     104  86.80163
## 537      98  94.03234
## 538      93  86.58101
## 539     104  86.56917
## 540      95  88.22369
## 541      95  85.39937
## 542      87  81.25140
## 543      67  82.13859
## 544      76  80.30914
## 545      88  74.44826
## 546      92  76.10264
## 547      75  74.08154
## 548      79  69.92862
## 549      76  69.95132
## 550      77  74.53685
## 551      93  77.21791
## 552      66  68.76742
## 553      92  67.21485
```

```
## 554      71  78.58185
## 555      98  74.37172
## 556      94  77.15918
## 557      94  83.53078
## 558     105  84.49880
## 559      79  86.41669
## 560      87  75.22195
## 561      71  73.30565
## 562      89  76.43071
## 563      77  67.01121
## 564      96  78.15236
## 565      86  84.22717
## 566      98 103.62586
## 567      56  99.78573
## 568      81  92.94880
## 569      87  96.14162
## 570      99  93.83560
## 571      72  87.21972
## 572      77  80.56718
## 573      94  75.47100
## 574      77  85.48369
## 575      80  81.64591
## 576      83  80.88031
## 577      90  72.00168
## 578      69  98.77644
## 579      70  95.99900
## 580      68  95.74313
## 581      82  81.34501
## 582     111  90.01800
## 583      79  90.45721
## 584      56  93.97424
## 585      94  79.58646
## 586      73  83.30092
## 587      80  80.62590
## 588      77  74.64967
## 589     102  81.50964
## 590      93  77.75612
## 591      61  68.67546
## 592      84  67.09276
## 593      85  74.64771
## 594      78  74.22550
## 595      80  91.69689
## 596      85  88.52477
## 597      92  83.85911
## 598      93  85.75061
## 599      71  84.90506
## 600      82  78.19194
## 601      80  86.69231
## 602      79  84.65045
## 603      81  90.56613
## 604      60  88.21833
## 605      75  89.20599
## 606      77  96.33209
## 607      73  89.50356
```

```
## 608      77  96.00533
## 609      80  94.62374
## 610      21  93.39055
## 611      87  95.00998
## 612      82  80.83454
## 613      73  75.38982
## 614      74  76.75518
## 615      67  78.99632
## 616      76  84.62905
## 617      66  82.87983
## 618      95  89.69512
## 619      92  80.89995
## 620      93  77.82870
## 621      64  76.45063
## 622      87  78.45039
## 623      94  76.99874
## 624      81  87.95743
## 625     101  92.81547
## 626      79  80.81087
## 627      88  78.03260
## 628      79  81.21524
## 629      74  79.27102
## 630      86  80.71248
## 631      83  81.70592
## 632      85  78.63917
## 633      59  75.29746
## 634      85  72.33944
## 635      68  76.57523
## 636      66  81.96928
## 637      43  82.33040
## 638      72  85.74432
## 639      64  85.14898
## 640      76  84.98782
## 641      80  90.15115
## 642      92  87.60581
## 643      65  92.92149
## 644      72  95.31093
## 645      81  81.02289
## 646      83  76.88477
## 647      76  84.77585
## 648      81  88.93106
## 649      89  82.10155
## 650      82  89.52683
## 651      65  86.02564
## 652      88  79.37780
## 653      79  84.48734
## 654      67  82.93491
## 655      62  68.22435
## 656      72  75.99833
## 657      60  85.70357
## 658      94  79.57618
## 659      79  70.89438
## 660      97  89.49693
## 661      75  77.55810
```

```
## 662      84  81.40149
## 663      85  73.90587
## 664      74  73.55784
## 665      93  83.10389
## 666     105  78.39615
## 667      86  77.05779
## 668      88  80.72315
## 669      94  73.46132
## 670     110  61.90398
## 671      97  62.28685
## 672     104  73.70169
## 673      98  74.54397
## 674      88  73.84600
## 675      86  61.25132
## 676      85  79.47124
## 677      88  83.26075
## 678      88  71.60880
## 679      92  83.55334
## 680      93  75.68196
## 681      90  76.36565
## 682      83  80.20358
## 683      80  79.05666
## 684      74  77.81118
## 685      66  83.28782
## 686      86  74.22485
## 687      87  76.82253
## 688      87  77.12806
## 689      68  80.63736
## 690      68  77.08373
## 691      69  71.73085
## 692      71  74.57166
## 693      80  79.22417
## 694      91  86.49261
## 695      97  82.07615
## 696      77  86.23495
## 697      69  96.73694
## 698      74  83.63311
## 699      89  82.47753
## 700      67  79.80981
## 701      92  80.53284
## 702      53  81.83612
## 703      56  76.42994
## 704      62  73.20961
## 705      64  75.03780
## 706      66  74.59428
## 707      91  84.51842
## 708      85  84.17397
## 709      61  85.30279
## 710      60  78.78881
## 711      91  84.07513
## 712      69  79.38092
## 713      85  80.61640
## 714      85  78.04178
## 715      51  84.50836
```

```
## 716      50  83.54679
## 717      66  80.08120
## 718      90  82.91023
## 719      98  78.49132
## 720      92  83.35613
## 721     101  73.32258
## 722      91  71.67373
## 723      75  98.19085
## 724      66  98.01751
## 725      83  78.46787
## 726      59  89.91365
## 727      99  75.16995
## 728      54  75.50304
## 729      84  67.49380
## 730      75  79.75337
## 731     104  83.97999
## 732      99  85.34265
## 733      92  82.35342
## 734      93  80.79035
## 735      80  75.78706
## 736      89  74.82875
## 737      84  75.08221
## 738      99  82.44515
## 739     103  78.18402
## 740      91  76.62192
## 741      87  84.27091
## 742      89  78.70232
## 743      86  80.78674
## 744     114  76.73710
## 745     104  76.18862
## 746      90  68.60446
## 747      87  71.96965
## 748      87  84.90424
## 749     108  80.17055
## 750      74  83.99648
## 751     112  86.59310
## 752     107  75.47677
## 753      63  75.53225
## 754      57  86.19133
## 755      52  85.54998
## 756      88  89.02909
## 757      74  86.18981
## 758      54  97.91304
## 759      77  98.63173
## 760      75  92.47344
## 761     102 104.48665
## 762      77  97.33576
## 763      99 100.85301
## 764      96  92.93128
## 765      85  82.82575
## 766      88  87.45635
## 767      96  85.89462
## 768      59  82.10294
## 769      83  89.28083
```

```
## 770      84  92.29101
## 771      94  72.82231
## 772      78  72.27080
## 773      55  67.86196
## 774      65  62.46698
## 775      53  74.97552
## 776      45  82.33372
## 777      67  73.87224
## 778      65  76.05064
## 779      67  77.17043
## 780      49  82.98197
## 781      92  77.89406
## 782      63  88.99333
## 783      67  86.81349
## 784      61  86.70833
## 785      71  85.03963
## 786      86  73.59994
## 787      81  81.58437
## 788      75  75.50149
## 789      78  72.12338
## 790      70  68.86495
## 791      67  80.02704
## 792      75  77.39364
## 793      77  78.07165
## 794      74  77.63262
## 795      92  82.97429
## 796     105  79.14137
## 797      92  89.78282
## 798     103  89.27390
## 799      85  83.86970
## 800      68  77.30894
## 801      84  69.58990
## 802     101  72.98084
## 803      84  88.69279
## 804      79  87.92394
## 805      82  91.70302
## 806      73  82.67101
## 807      65  71.85679
## 808      87  68.10150
## 809      75  81.84717
## 810      53  78.12253
## 811      81  69.83695
## 812      97  76.46566
## 813     100  86.11409
## 814      96  90.88470
## 815      72  91.80391
## 816      67  92.11302
## 817      52  84.80266
## 818      61  74.65166
## 819      71  77.57825
## 820      69  76.78635
## 821      81  82.36152
## 822      89  82.50355
## 823      75  88.04743
```

```
## 824      91  87.72790
## 825      76  76.63681
## 826      98  75.42680
## 827      66  82.96084
## 828      64  78.21579
## 829      57  76.27439
## 830      68  73.79846
## 831      82  81.10167
## 832      93  75.42576
## 833      69  76.49613
## 834      78  74.59911
## 835      92  89.20900
## 836      92  87.22479
## 837      97  75.04804
## 838      91  71.78931
## 839      58  78.02374
## 840      64  76.61472
## 841      85  63.46132
## 842      82  73.67504
## 843      77  77.48161
## 844      70  80.45738
## 845      84  78.22131
## 846      87  80.05917
## 847      88  80.85200
## 848      90  80.57371
## 849      72  86.57068
## 850      74  85.80458
## 851      87  83.86479
## 852      85  83.11116
## 853      77  86.62546
## 854      68  86.96736
## 855      90  81.63054
## 856      88  84.51782
## 857      86  83.66754
## 858      90  90.72007
## 859     101  84.56464
## 860      75  79.70922
## 861      56  80.94532
## 862      52  79.55306
## 863      84  83.57741
## 864      82  76.30040
## 865     100  79.15809
## 866     101  72.63969
## 867      92  77.97984
## 868      76  80.06715
## 869      89  75.38970
## 870     112  80.66269
## 871     102  75.27283
## 872      85  83.24337
## 873      80  74.94358
## 874      75  81.51085
## 875      93  83.24421
## 876      93  84.04611
## 877      87  77.97604
```

```
## 878      76  75.08056
## 879      90  72.00019
## 880      78  78.40749
## 881      72  74.36827
## 882      70  69.61521
## 883      73  78.99502
## 884      95  76.21166
## 885      85  79.95868
## 886      69  85.97882
## 887      69  74.10177
## 888      63  74.43285
## 889      70  76.14842
## 890      62  76.55578
## 891      86  71.64883
## 892      64  88.44513
## 893      79  82.81897
## 894      70  79.19356
## 895      77  74.99922
## 896      83  84.37986
## 897      77  74.92339
## 898      53  81.98816
## 899      86  83.05794
## 900      96  79.14053
## 901      96  76.41732
## 902      76  77.00170
## 903      84  83.40517
## 904      87  77.80671
## 905      55  80.72405
## 906      96  84.92448
## 907      88  76.37162
## 908      68  80.14464
## 909      68  81.99346
## 910      83  79.12052
## 911      75  63.72078
## 912      85  60.35400
## 913      76  66.78161
## 914      81  60.68783
## 915      92  59.18272
## 916      80  70.53416
## 917      85  81.46170
## 918      92  71.45605
## 919     100  74.57813
## 920      85  71.72674
## 921      98  75.34780
## 922      53  81.52163
## 923      63  83.94570
## 924      56  91.68011
## 925      47  85.36356
## 926      66  81.99874
## 927      53  84.14420
## 928      74  73.31153
## 929      67  78.85476
## 930      40  76.00103
## 931      96  77.32058
```

```
## 932      80  73.77252
## 933      94  90.41431
## 934      84  85.21387
## 935      88  73.53234
## 936      86  76.49770
## 937      88  70.54324
## 938      67  84.21458
## 939      70  97.21267
## 940      67  80.64946
## 941      66  79.82280
## 942     103  76.58877
## 943      76  76.01548
## 944      69  75.33370
## 945      48  71.64498
## 946      83  84.87058
## 947      77  79.01173
## 948      78  81.63345
## 949      86  77.29707
## 950      59  79.07006
## 951      45  78.70672
## 952      77  98.16423
## 953      73  85.26197
## 954      86  90.39484
## 955      98  94.47383
## 956     108 104.75405
## 957     104  92.78122
## 958      97  93.66794
## 959      92  95.31631
## 960      88 102.26987
## 961     104  95.77832
## 962     100  93.74198
## 963      94  87.68429
## 964      85  85.41279
## 965      84  86.50211
## 966      73  86.91786
## 967      67  86.91658
## 968      88  92.97927
## 969      67 100.04520
## 970      78  92.21314
## 971     104  86.56469
## 972      88  90.68067
## 973      49  89.20259
## 974      94  95.19330
## 975      79  84.80754
## 976      85  85.47340
## 977      78  79.99373
## 978      79  66.40110
## 979      75  74.33150
## 980      68  80.55107
## 981      83  77.92324
## 982      84  75.59030
## 983      73  74.68523
## 984      76  73.58853
## 985      97  79.11850
```

```
## 986      85  86.74757
## 987      78  91.30102
## 988      92  80.72652
## 989      80  91.67552
## 990      98  85.58264
## 991     104  85.24275
## 992      98  79.61319
## 993      93  94.02997
## 994     104  91.34655
## 995      95  86.76379
## 996      95  83.15531
## 997      87  80.73357
## 998      67  78.38100
## 999      76  82.05406
## 1000     88  84.55653
## 1001     92  85.12766
## 1002     75  85.11712
## 1003     79  94.73952
## 1004     76  92.33759
## 1005     77  89.29616
## 1006     93  89.76105
## 1007     66  87.44643
## 1008     92  91.54011
## 1009     71  93.41918
## 1010     98  92.01233
## 1011     94  91.22298
## 1012     94  93.76596
## 1013    105  80.15303
## 1014     79  74.74933
## 1015     87  75.69253
## 1016     71  76.00785
## 1017     89  71.17943
## 1018     77  66.08970
## 1019     96  80.56114
## 1020     86  85.16664
## 1021     98  87.91888
## 1022     56  74.29519
## 1023     81  73.74423
## 1024     87  72.48966
## 1025     99  76.45746
## 1026     72  79.55431
## 1027     77  82.36934
## 1028     94  77.09259
## 1029     77  85.04632
## 1030     80  87.48039
## 1031     83  94.97976
## 1032     90  91.99768
## 1033     69  90.99592
## 1034     70  89.30324
## 1035     68  81.71456
## 1036     82  79.33073
## 1037    111  89.00786
## 1038     79  87.12255
## 1039     56  81.65797
```

```
## 1040     94  81.35012
## 1041     73  68.60934
## 1042     80  67.88631
## 1043     77  68.54235
## 1044    102  67.56090
## 1045     93  69.46154
## 1046     61  82.63933
## 1047     84  85.29402
## 1048     85  74.25824
## 1049     78  76.50022
## 1050     80  79.50063
## 1051     85  78.73351
## 1052     92  77.56332
## 1053     93  71.55565
## 1054     71  70.39303
## 1055     82  74.11267
## 1056     80  85.19926
## 1057     79  78.36045
## 1058     81  77.45106
## 1059     60  83.83186
## 1060     75  76.32044
## 1061     77  86.73581
## 1062     73  83.39038
## 1063     77  83.93669
## 1064     80  83.00696
## 1065     21  86.60338
## 1066     87  86.10335
## 1067     82  88.47441
## 1068     73  83.14315
## 1069     74  91.80580
## 1070     67  88.58660
## 1071     76  82.50243
## 1072     66  79.89829
## 1073     95  91.77266
## 1074     92  85.07039
## 1075     93  83.01118
## 1076     64  85.68794
## 1077     87  95.76558
## 1078     94  90.58290
## 1079     81  91.94511
## 1080    101  96.31662
## 1081     79  84.96631
## 1082     88  69.83334
## 1083     79  73.56281
## 1084     74  72.00398
## 1085     86  74.46932
## 1086     83  73.08326
## 1087     85  71.27250
## 1088     59  84.53501
## 1089     85  83.59480
## 1090     68  81.98597
## 1091     66  86.48434
## 1092     43  81.75761
## 1093     72  76.77313
```

```
## 1094     64  76.37420
## 1095     76  75.89710
## 1096     80  77.61684
## 1097     92  81.74684
## 1098     65  81.41920
## 1099     72  76.82860
## 1100     81  74.59177
## 1101     83  79.96919
## 1102     76  94.95448
## 1103     81  83.66543
## 1104     89  76.84529
## 1105     82  87.13900
## 1106     65  71.93989
## 1107     88  78.74975
## 1108     79  78.76011
## 1109     67  66.61349
## 1110     62  70.79383
## 1111     72  70.84328
## 1112     60  71.91145
## 1113     94  67.12428
## 1114     79  71.62438
## 1115     97  71.33810
## 1116     75  71.90173
## 1117     84  73.78630
## 1118     85  81.48361
## 1119     74  84.80836
## 1120     93  83.19514
## 1121    105  81.29440
## 1122     86  73.80876
## 1123     88  67.63890
## 1124     94  73.59700
## 1125    110  79.16576
## 1126     97  72.63248
## 1127    104  72.41918
## 1128     98  73.04657
## 1129     88  66.49953
## 1130     86  73.95977
## 1131     85  71.42417
## 1132     88  69.30896
## 1133     88  71.54921
## 1134     92  76.78362
## 1135     93  82.63552
## 1136     90  95.12850
## 1137     83  83.90767
## 1138     80  84.47296
## 1139     74  87.77827
## 1140     66  80.02842
## 1141     86  84.42900
## 1142     87  85.19428
## 1143     87  73.29018
## 1144     68  78.51938
## 1145     68  75.82602
## 1146     69  71.69911
## 1147     71  75.95149
```

```
## 1148     80  85.64490
## 1149     91  83.34346
## 1150     97  77.77329
## 1151     77  74.54778
## 1152     69  86.37762
## 1153     74  80.14170
## 1154     89  83.43499
## 1155     67  85.01173
## 1156     92  79.65990
## 1157     53  97.22181
## 1158     56  78.07314
## 1159     62  91.60057
## 1160     64  92.29360
## 1161     66  92.95479
## 1162     91  89.26299
## 1163     85  79.41539
## 1164     61  80.53341
## 1165     60  73.59357
## 1166     91  81.24245
## 1167     69  83.56699
## 1168     85  82.38249
## 1169     85  93.33796
## 1170     51  85.35170
## 1171     50  76.41227
## 1172     66  79.07561
## 1173     90  72.52526
## 1174     98  95.66395
## 1175     92  87.66176
## 1176    101  92.90300
## 1177     91  99.28524
## 1178     75  86.39980
## 1179     66  81.58923
## 1180     83  85.33114
## 1181     59  93.70586
## 1182     99  93.36064
## 1183     54  77.33035
## 1184     84  79.55576
## 1185     75  75.40082
## 1186    104  79.76631
## 1187     99  81.47024
## 1188     92  80.17441
## 1189     93  77.22722
## 1190     80  81.74268
## 1191     89  78.60004
## 1192     84  84.63944
## 1193     99  79.37534
## 1194    103  76.55477
## 1195     91  78.77538
## 1196     87  85.52381
## 1197     89  84.44809
## 1198     86  77.59152
## 1199    114  87.63939
## 1200    104  86.29028
## 1201     90  84.57447
```

```
## 1202     87  87.19745
## 1203     87  77.20342
## 1204    108  73.46173
## 1205     74  73.29697
## 1206    112  78.67011
## 1207    107  74.68087
## 1208     63  71.33847
## 1209     57  70.73626
## 1210     52  70.52728
## 1211     88  78.88400
## 1212     74  71.53207
## 1213     54  71.39736
## 1214     77  74.93688
## 1215     75  81.09400
## 1216    102  72.57754
## 1217     77  75.39038
## 1218     99  75.68757
## 1219     96  80.16107
## 1220     85  89.40135
## 1221     88  72.16460
## 1222     96  83.38028
## 1223     59  81.32313
## 1224     83  84.89702
## 1225     84  84.84828
## 1226     94  80.52925
## 1227     78  76.62719
## 1228     55  79.65161
## 1229     65  79.83045
## 1230     53  82.55894
## 1231     45  73.84234
## 1232     67  81.12999
## 1233     65  73.31931
## 1234     67  63.40639
## 1235     49  75.90363
## 1236     92  67.21770
## 1237     63  70.21542
## 1238     67  68.87801
## 1239     61  72.77913
## 1240     71  70.17411
## 1241     86  71.05437
## 1242     81  81.34791
## 1243     75  79.48590
## 1244     78  80.68455
## 1245     70  72.75448
## 1246     67  74.81253
## 1247     75  78.33119
## 1248     77  80.06916
## 1249     74  73.74964
## 1250     92  74.50480
## 1251    105  75.05737
## 1252     92  80.77484
## 1253    103  80.54898
## 1254     85  80.71122
## 1255     68  80.60354
```

```
## 1256     84  84.06425
## 1257    101  81.92056
## 1258     84  74.13072
## 1259     79  74.34099
## 1260     82  82.79877
## 1261     73  78.63570
## 1262     65  79.79308
## 1263     87  86.05803
## 1264     75  77.78018
## 1265     53  79.84836
## 1266     81  84.47589
## 1267     97  79.79337
## 1268    100  76.72617
## 1269     96  79.21975
## 1270     72  80.25741
## 1271     67  79.14570
## 1272     52  83.44643
## 1273     61  93.98953
## 1274     71  94.00703
## 1275     69  93.68490
## 1276     81  91.38872
## 1277     89  93.75317
## 1278     75  92.88496
## 1279     91  91.70101
## 1280     76  85.48907
## 1281     98  80.65960
## 1282     66  74.69654
## 1283     64  75.32830
## 1284     57  81.12020
## 1285     68  70.93756
## 1286     82  70.80932
## 1287     93  74.57660
## 1288     69  85.39189
## 1289     78  86.64623
## 1290     92  84.02929
## 1291     92  79.94081
## 1292     97  78.06523
## 1293     91  84.11563
## 1294     58  88.72233
## 1295     64  86.22934
## 1296     85  79.80524
## 1297     82  86.74477
## 1298     77  87.62938
## 1299     70  92.83543
## 1300     84  94.36469
## 1301     87  86.01547
## 1302     88  80.54337
## 1303     90  73.29776
## 1304     72  78.68654
## 1305     74  82.39342
## 1306     87  78.33648
## 1307     85  79.95985
## 1308     77  81.96134
## 1309     68  81.59286
```

```
## 1310     90  79.05989
## 1311     88  77.86516
## 1312     86  82.34401
## 1313     90  75.16981
## 1314    101  80.07663
## 1315     75  82.51050
## 1316     56  82.61926
## 1317     52  96.35798
## 1318     84  91.78498
## 1319     82  91.83003
## 1320    100  96.02732
## 1321    101  86.45282
## 1322     92  82.53612
## 1323     76  86.96077
## 1324     89  87.22632
## 1325    112  84.18313
## 1326    102  83.26233
## 1327     85  89.16195
## 1328     80  81.82784
## 1329     75  80.15164
## 1330     93  76.91678
## 1331     93  72.78148
## 1332     87  70.53999
## 1333     76  78.85112
## 1334     90  81.40689
## 1335     78  83.17569
## 1336     72  87.87173
## 1337     70  75.63225
## 1338     73  78.58231
## 1339     95  73.46579
## 1340     85  78.71896
## 1341     69  82.70960
## 1342     69  73.33920
## 1343     63  77.90761
## 1344     70  79.98687
## 1345     62  70.52942
## 1346     86  84.46694
## 1347     64  98.40246
## 1348     79  86.63365
## 1349     70  78.47395
## 1350     77  85.92126
## 1351     83  83.71443
## 1352     77  87.85776
## 1353     53  94.71885
## 1354     86  90.06946
## 1355     96  81.89370
## 1356     96  71.95118
## 1357     76  75.62014
## 1358     84  88.90679
## 1359     87  95.62316
## 1360     55  72.47442
## 1361     96  97.62852
## 1362     88  88.88781
## 1363     68  73.04341
```

```
## 1364     68  74.20625
## 1365     83  74.85814
## 1366     75  75.56058
## 1367     85  67.96074
## 1368     76  64.83668
## 1369     81  66.12139
## 1370     92  77.36962
## 1371     80  86.11549
## 1372     85  88.08887
## 1373     92  84.97467
## 1374    100  79.91790
## 1375     85  82.47619
## 1376     98  76.35949
## 1377     53  78.68785
## 1378     63  76.76146
## 1379     56  90.75052
## 1380     47  83.07884
## 1381     66  90.01429
## 1382     53  88.37454
## 1383     74  92.99445
## 1384     67  90.64963
## 1385     40  90.30104
## 1386     96  83.17853
## 1387     80  79.07535
## 1388     94  77.46767
## 1389     84  84.16275
## 1390     88  79.55429
## 1391     86  80.16002
## 1392     88  80.89242
## 1393     67  83.19119
## 1394     70  81.77316
## 1395     67  88.11476
## 1396     66  78.96010
## 1397    103  82.05215
## 1398     76  78.88763
## 1399     69  79.95473
## 1400     48  87.94550
## 1401     83  84.21109
## 1402     77  82.94389
## 1403     78  81.63757
## 1404     86  82.86589
## 1405     59  82.07041
## 1406     45  88.16516
## 1407     77  83.92208
## 1408     73  82.40573
## 1409     86  82.94323
## 1410     98  79.72355
## 1411    108  80.43830
## 1412    104  81.25349
## 1413     97  77.38370
## 1414     92  78.72661
## 1415     88  78.92403
## 1416    104  73.09722
## 1417    100  76.51427
```

```
## 1418     94  75.89040
## 1419     85  75.09768
## 1420     84  69.03655
## 1421     73  73.11790
## 1422     67  80.32345
## 1423     88  81.62448
## 1424     67  72.46322
## 1425     78  73.05070
## 1426    104  82.60617
## 1427     88  76.80164
## 1428     49  74.71730
## 1429     94  72.53451
## 1430     79  79.24098
## 1431     85  80.59521
## 1432     78  77.15610
## 1433     79  74.10392
## 1434     75  79.46361
## 1435     68  81.59949
## 1436     83  92.84525
## 1437     84  86.74162
## 1438     73  89.00812
## 1439     76  83.81009
## 1440     97  82.45990
## 1441     85  86.81278
## 1442     78  87.08329
## 1443     92  81.02582
## 1444     80  81.79036
## 1445     98  76.09764
## 1446    104  79.58062
## 1447     98  70.85242
## 1448     93  73.77270
## 1449    104  78.06725
## 1450     95  82.56246
## 1451     95  82.22541
## 1452     87  82.97301
## 1453     67  77.64168
## 1454     76  79.54539
## 1455     88  68.24060
## 1456     92  70.24206
## 1457     75  67.33869
## 1458     79  80.95609
## 1459     76  77.96052
## 1460     77  70.39466
## 1461     93  75.42098
## 1462     66  79.98365
## 1463     92  72.26114
## 1464     71  80.33432
## 1465     98  72.40334
## 1466     94  76.28209
## 1467     94  77.74120
## 1468    105  76.98433
## 1469     79  74.78655
## 1470     87  84.57421
## 1471     71  77.67434
```

```
## 1472     89  84.70901
## 1473     77  78.88817
## 1474     96  86.51928
## 1475     86  94.59262
## 1476     98  83.68060
## 1477     56  90.09426
## 1478     81  86.45699
## 1479     87  85.18612
## 1480     99  73.40551
## 1481     72  73.27707
## 1482     77  91.99847
## 1483     94  86.52118
## 1484     77  92.42330
## 1485     80  84.04087
## 1486     83  83.86156
## 1487     90  85.52067
## 1488     69  80.89763
## 1489     70  86.58041
## 1490     68  91.77189
## 1491     82  82.89224
## 1492    111  78.98014
## 1493     79  87.14506
## 1494     56  89.37809
## 1495     94  74.04740
## 1496     73  75.81540
## 1497     80  84.01966
## 1498     77  88.51943
## 1499    102  72.64024
## 1500     93  79.09215
## 1501     61  70.49611
## 1502     84  79.19742
## 1503     85  74.80890
## 1504     78  70.05598
## 1505     80  79.93610
## 1506     85  71.64301
## 1507     92  81.23458
## 1508     93  84.56800
## 1509     71  81.05502
## 1510     82  77.11747
## 1511     80  78.69532
## 1512     79  76.56764
## 1513     81  74.31725
## 1514     60  79.24925
## 1515     75  77.95751
## 1516     77  73.35033
## 1517     73  82.50775
## 1518     77  77.38719
## 1519     80  79.51796
## 1520     21  67.76209
## 1521     87  80.89661
```

```
mean (apply(predicted_values, 1, min)/apply(predicted_values, 1, max)) # calculate accuracy
```

```
## [1] 0.8597468
```

The prediction accuracy for the OLS Model5 is at 85.94% which is not bad for this purpose. But lets compare it to the Champion Model- The improved Ridge Regression.

```
predicted <- predict(Model6_Improved, newx = test_Ind)# predict on test data
predicted_values <- cbind (actual=test_baseball$TARGET_WINS, predicted)  # combine
predicted_values
```

```
##      actual        s0
## 3        75  74.07865
## 5        85  80.67628
## 6        76  74.05432
## 13       81  79.91531
## 15       92  90.09281
## 19       80  79.25698
## 23       85  84.19374
## 29       92  90.87854
## 32      100  97.27260
## 33       85  85.16326
## 35       98  95.04600
## 46       53  57.22048
## 47       63  66.61415
## 50       56  61.47513
## 51       47  60.43835
## 57       66  70.56369
## 61       53  62.29553
## 63       74  75.75168
## 64       67  69.60744
## 68       40  52.73368
## 79       96  93.19580
## 82       80  81.36292
## 84       94  89.55803
## 87       84  81.37620
## 88       88  85.33687
## 89       86  82.62563
## 96       88  84.72422
## 97       67  69.82035
## 98       70  71.51265
## 101      67  70.51944
## 104      66  69.05660
## 113     103  97.61109
## 120      76  73.98883
## 121      69  70.24485
## 127      48  56.73932
## 130      83  82.53503
## 132      77  79.12461
## 133      78  78.61621
## 135      86  85.15338
## 137      59  66.86171
## 139      45  58.70318
## 141      77  77.40360
## 153      73  74.45864
## 158      86  83.54686
## 161      98  92.01525
## 165     108 100.17113
```

```
## 166     104  97.23854
## 168      97  93.02032
## 170      92  88.27217
## 171      88  84.39851
## 174     104  96.61028
## 175     100  95.44612
## 176      94  91.59146
## 178      85  83.25367
## 179      84  84.32441
## 180      73  75.44189
## 181      67  72.45175
## 189      88  89.52664
## 194      67  71.42255
## 195      78  80.78964
## 199     104  99.62572
## 200      88  85.11669
## 201      49  57.69170
## 202      94  91.83884
## 209      79  76.53932
## 214      85  85.09520
## 218      78  79.85781
## 219      79  79.02413
## 236      75  75.77080
## 241      68  72.50719
## 255      83  83.64777
## 257      84  83.23368
## 258      73  73.28172
## 260      76  80.84667
## 261      97  94.22110
## 262      85  86.67966
## 263      78  80.18647
## 264      92  90.43048
## 273      80  78.79848
## 274      98  89.17362
## 276     104  94.21167
## 280      98  93.77417
## 281      93  89.95138
## 289     104  99.87144
## 290      95  94.91558
## 292      95  89.17566
## 304      87  84.85421
## 311      67  72.27381
## 312      76  76.67231
## 319      88  84.62182
## 320      92  87.07585
## 324      75  76.56518
## 326      79  79.50301
## 332      76  78.56993
## 333      77  77.99872
## 334      93  88.72771
## 347      66  72.51551
## 349      92  87.93529
## 350      71  72.68989
## 351      98  90.19879
```

```
## 354      94  86.37968
## 358      94  88.76311
## 359     105  96.98935
## 367      79  81.68741
## 368      87  88.72303
## 370      71  75.30778
## 371      89  87.57944
## 375      77  77.19730
## 382      96  92.05930
## 388      86  85.88513
## 390      98  90.58535
## 393      56  62.01530
## 396      81  79.73716
## 403      87  85.41161
## 404      99  93.56622
## 407      72  74.04643
## 408      77  78.60858
## 412      94  91.11958
## 413      77  83.04647
## 415      80  82.05875
## 418      83  83.45027
## 421      90  89.46627
## 429      69  70.82448
## 430      70  71.42142
## 433      68  73.91503
## 436      82  80.47332
## 438     111 101.12657
## 439      79  78.99482
## 446      56  63.01100
## 455      94  86.90987
## 463      73  75.13658
## 466      80  81.77184
## 474      77  77.31686
## 481     102  99.78298
## 483      93  89.72576
## 486      61  66.48160
## 489      84  84.66249
## 498      85  85.84073
## 500      78  78.53055
## 502      80  81.49056
## 511      85  83.38623
## 513      92  87.29065
## 515      93  87.52093
## 520      71  76.08320
## 533      82  80.93853
## 542      80  79.79037
## 546      79  78.50452
## 547      81  78.41600
## 552      60  64.85554
## 553      75  73.46205
## 555      77  75.89477
## 561      73  74.46642
## 562      77  78.12477
## 574      80  82.82445
```

```
## 577      21  39.00263
## 578      87  88.47148
## 583      82  85.90729
## 585      73  76.41570
## 586      74  77.66433
## 588      67  71.72776
## 589      76  79.24566
## 591      66  67.82089
## 594      95  87.63211
## 596      92  89.78696
## 599      93  90.54697
## 600      64  70.09078
## 602      87  87.87327
## 608      94  94.13143
## 623      81  81.22788
## 625     101  96.37246
## 626      79  80.68586
## 628      88  85.57174
## 630      79  79.39658
## 636      74  76.31980
## 637      86  85.30160
## 639      83  82.75292
## 642      85  83.92568
## 646      59  65.87407
## 650      85  86.75791
## 651      68  73.64785
## 654      66  71.26541
## 656      43  53.76351
## 657      72  76.10788
## 659      64  68.58377
## 660      76  79.44538
## 661      80  79.67508
## 662      92  88.56235
## 669      65  68.88406
## 675      72  70.51963
## 676      81  80.43831
## 688      83  80.98940
## 689      76  78.45011
## 692      81  80.22524
## 700      89  86.56282
## 701      82  83.08435
## 704      65  68.46411
## 707      88  85.22556
## 708      79  79.35248
## 716      67  73.95831
## 722      62  68.48650
## 724      72  78.36999
## 725      60  65.84121
## 726      94  91.68146
## 733      79  79.32536
## 739      97  91.49096
## 740      75  76.40066
## 742      84  82.38514
## 743      85  83.13765
```

```
## 745      74  76.14721
## 750      93  89.69729
## 751     105  98.61727
## 752      86  84.40549
## 757      88  86.62848
## 759      94  92.31263
## 761     110 104.81811
## 762      97  94.72569
## 763     104  99.17090
## 764      98  94.11495
## 765      88  85.83052
## 768      86  84.29128
## 775      85  83.22573
## 776      88  86.93034
## 780      88  86.34568
## 781      92  86.82936
## 792      93  88.20955
## 794      90  86.04568
## 796      83  80.67988
## 805      80  81.62259
## 809      74  75.85894
## 811      66  68.44287
## 815      86  84.97602
## 817      87  86.40217
## 820      87  83.71990
## 829      68  71.80859
## 831      68  73.14907
## 838      69  69.01628
## 842      71  71.89235
## 844      80  79.38002
## 846      91  87.40029
## 852      97  93.37473
## 857      77  80.76891
## 859      69  74.72701
## 861      74  77.01859
## 863      89  86.42158
## 864      67  70.48374
## 865      92  87.96347
## 868      53  62.93355
## 871      56  63.46363
## 872      62  69.21944
## 873      64  68.16330
## 874      66  69.72709
## 878      91  85.84046
## 886      85  84.99858
## 888      61  67.27762
## 889      60  67.30272
## 893      91  89.39334
## 902      69  74.57867
## 903      85  85.05559
## 909      85  82.58521
## 911      51  55.86381
## 912      50  55.03678
## 913      66  67.07862
```

```
## 921      90  85.36974
## 922      98  92.32758
## 924      92  90.24696
## 925     101  94.65113
## 927      91  88.14610
## 936      75  75.92853
## 937      66  69.16423
## 938      83  83.02408
## 939      59  70.06344
## 941      99  91.33927
## 945      54  59.78845
## 948      84  82.16805
## 949      75  75.09822
## 952     104  99.42330
## 959      99  96.26093
## 967      92  88.07983
## 980      93  88.46863
## 983      80  78.28057
## 984      89  84.84419
## 985      84  82.38454
## 988      99  93.36169
## 989     103  97.07410
## 991      91  88.47516
## 992      87  85.03850
## 995      89  87.81637
## 996      86  84.85991
## 1003    114 106.63388
## 1007    104  98.18502
## 1013     90  86.63951
## 1014     87  82.64925
## 1016     87  82.74036
## 1021    108 100.80724
## 1028     74  75.56315
## 1031    112 105.12639
## 1032    107 101.80781
## 1035     63  70.15573
## 1039     57  65.73751
## 1045     52  60.57781
## 1046     88  87.16895
## 1047     74  77.58185
## 1048     54  61.21902
## 1051     77  77.04724
## 1053     75  74.03010
## 1057    102  93.50762
## 1061     77  78.66181
## 1064     99  92.49106
## 1067     96  92.05674
## 1078     85  84.79150
## 1080     88  89.17997
## 1081     96  91.84400
## 1083     59  66.55260
## 1088     83  82.03449
## 1089     84  82.56346
## 1091     94  90.03228
```

```
## 1092     78  78.51008
## 1103     55  66.48644
## 1108     65  70.52334
## 1111     53  59.73185
## 1112     45  55.22987
## 1114     67  69.71269
## 1117     65  69.53627
## 1122     67  69.75062
## 1124     49  58.26780
## 1127     92  85.65420
## 1130     63  67.39770
## 1131     67  69.62018
## 1132     61  65.79444
## 1133     71  71.94865
## 1135     86  84.93885
## 1139     81  82.67345
## 1140     75  76.28956
## 1146     78  77.90860
## 1147     70  73.49290
## 1150     67  72.11543
## 1151     75  77.42724
## 1152     77  81.83518
## 1157     74  79.35411
## 1159     92  89.26932
## 1162    105  98.68474
## 1163     92  87.08020
## 1166    103  94.40368
## 1170     85  84.54342
## 1172     68  70.91109
## 1174     84  84.48915
## 1177    101  97.69945
## 1182     84  86.05961
## 1183     79  78.88440
## 1192     82  82.68712
## 1194     73  74.91342
## 1199     65  72.36246
## 1200     87  86.17062
## 1201     75  77.30421
## 1203     53  62.14274
## 1212     81  79.69621
## 1215     97  91.68607
## 1216    100  91.95080
## 1226     96  91.50050
## 1231     72  73.06033
## 1233     67  70.61613
## 1234     52  57.49362
## 1237     61  64.49382
## 1240     71  71.05371
## 1242     69  72.89520
## 1243     81  80.39189
## 1247     89  84.64708
## 1248     75  76.80269
## 1253     91  88.49694
## 1254     76  79.06338
```

```
## 1255     98  92.27042
## 1258     66  70.15116
## 1261     64  68.30705
## 1262     57  64.54914
## 1267     68  71.60078
## 1272     82  82.98199
## 1281     93  92.13678
## 1283     69  73.56670
## 1284     78  80.61854
## 1290     92  87.25919
## 1295     92  88.73501
## 1297     97  92.58498
## 1312     91  87.72387
## 1313     58  64.64741
## 1316     64  70.17995
## 1317     85  85.55916
## 1318     82  83.18222
## 1319     77  80.41126
## 1322     70  72.63856
## 1324     84  83.64125
## 1325     87  85.07907
## 1332     88  82.63402
## 1334     90  86.03900
## 1337     72  73.56889
## 1339     74  73.85974
## 1341     87  84.72266
## 1344     85  83.82986
## 1347     77  81.81360
## 1349     68  72.15103
## 1350     90  87.89547
## 1351     88  88.01430
## 1352     86  87.00602
## 1354     90  88.64602
## 1355    101  95.60534
## 1356     75  76.63309
## 1367     56  61.45353
## 1369     52  56.97437
## 1380     84  83.91178
## 1381     82  83.68151
## 1383    100  96.29959
## 1389    101  95.02712
## 1391     92  89.00587
## 1393     76  78.71032
## 1395     89  87.90121
## 1397    112 101.63790
## 1399    102  94.61578
## 1404     85  84.37227
## 1408     80  80.87611
## 1411     75  77.06872
## 1415     93  88.84448
## 1416     93  87.04021
## 1421     87  83.34897
## 1422     76  78.22113
## 1423     90  87.67625
```

```
## 1424     78  77.51203
## 1428     72  73.72132
## 1433     70  71.95170
## 1435     73  75.91076
## 1438     95  92.22728
## 1441     85  87.84080
## 1445     69  72.63016
## 1446     69  73.09523
## 1449     63  69.88057
## 1450     70  74.42339
## 1455     62  65.83215
## 1458     86  83.86898
## 1460     64  67.07571
## 1462     79  78.48199
## 1467     70  73.43925
## 1471     77  77.06716
## 1472     83  83.65332
## 1473     77  78.84142
## 1481     53  60.10157
## 1483     86  86.16772
## 1484     96  93.13162
## 1489     96  93.46028
## 1492     76  77.42201
## 1493     84  85.24817
## 1497     87  86.98428
## 1504     55  60.53689
## 1507     96  90.41522
## 1512     88  85.16859
## 1514     68  72.00931
## 1516     68  71.44634
## 1517     83  82.08759
```

Lets calculate the accuracy of using Model6 for our predictions

```
mean (apply(predicted_values, 1, min)/apply(predicted_values, 1, max)) # calculate accuracy
```

```
## [1] 0.9564133
```

The prediction accuracy of the improved Ridge Regression Model is 95.75%.

```
ModelName <- c("Model3", "Model5","Model6")
Model_Accuracy <- c("85.85%", "85.85%", "95.75%")
AccuracyCompared <- data.frame(ModelName,Model_Accuracy)
AccuracyCompared
```

ModelName	Model_Accuracy
Model3	85.85%
Model5	85.85%
Model6	95.75%

The prediction accuracy of the improved Ridge Regression Model6 is at 95.75% which is very good for this purpose.

Conclusion

The improved Model6 shows significant improvement from all the OLS Models when the R-Squared and the RMSE of the Models are compared. THis Model also predict TARGET WINS better than the OLS models because it is more stable and less prone to overfitting.

The chosen OLS Model3 and Model5 are due to the improved F-Statistic, positive variable coefficients and low Standard Errors. We will chose to make our predictions with the champion model the improved Ridge Regression Model6 because it beats all the OLs models when the model performance metrics are compared as well as the predictive ability of this model.

For Models 3 and 4, the variables were chosen just to test how the offfensive categories only would affect the model and how only defensive variables would affect the model. Based on the Coefficients for each model, the third model took the highest coefficient from each category model.

For offense, the two highest were HR and Triples. Which intuively does make sense because the HR and triple are two of the highest objectives a hitter can achieve when batting and thus the higher the totals in those categories the higher the runs scored which help a team win. And on the defensive side, the two highest cooeficients were Hits and WALKS. Which again just looking at it from a common sense point does make sense because as a pitcher, what they want to do is limit the numbers of times a batter gets on base whether by a hit or walk. Unless its an error, if a batter does not get a hit or walk then the outcome would be an out which would in essence limit the amount of runs scored by the opposing team.