Assignment-1

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

2/12/2020

Table of Contents

## 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

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

##   
## Attaching package: 'psych'

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

#library(statsr)  
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

library(PerformanceAnalytics)

## Loading required package: xts

## 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)  
library(reshape2)

##   
## Attaching package: 'reshape2'

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

library(rcompanion)

##   
## 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)

## 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)  
library(huxtable)

##   
## 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)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

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

## Loaded glmnet 4.1-1

##   
## Attaching package: 'glmnet'

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

library(sjPlot)

##   
## Attaching package: 'sjPlot'

## The following object is masked from 'package:huxtable':  
##   
## font\_size

library(modelr)

# Load data sets  
  
baseball\_eva <- read.csv("https://raw.githubusercontent.com/anilak1978/data621/master/moneyball-evaluation-data.csv")  
baseball\_train <- read.csv("https://raw.githubusercontent.com/anilak1978/data621/master/moneyball-training-data.csv")

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 ...

TARGET\_Wins<-as.numeric(baseball\_train$TARGET\_WINS)

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** | **range** | **skew** | **kurtosis** | **se** | **pct\_missing** |
| INDEX | 1 | 2.28e+03 | 1.27e+03 | 736 | 1.27e+03 | 1.27e+03 | 953 | 1 | 2.54e+03 | 2.53e+03 | 0.00421 | -1.22 | 15.4 | 1 |
| TARGET\_WINS | 2 | 2.28e+03 | 80.8 | 15.8 | 82 | 81.3 | 14.8 | 0 | 146 | 146 | -0.399 | 1.03 | 0.33 | 1 |
| TEAM\_BATTING\_H | 3 | 2.28e+03 | 1.47e+03 | 145 | 1.45e+03 | 1.46e+03 | 114 | 891 | 2.55e+03 | 1.66e+03 | 1.57 | 7.28 | 3.03 | 1 |
| TEAM\_BATTING\_2B | 4 | 2.28e+03 | 241 | 46.8 | 238 | 240 | 47.4 | 69 | 458 | 389 | 0.215 | 0.00616 | 0.981 | 1 |
| TEAM\_BATTING\_3B | 5 | 2.28e+03 | 55.2 | 27.9 | 47 | 52.2 | 23.7 | 0 | 223 | 223 | 1.11 | 1.5 | 0.586 | 1 |
| TEAM\_BATTING\_HR | 6 | 2.28e+03 | 99.6 | 60.5 | 102 | 97.4 | 78.6 | 0 | 264 | 264 | 0.186 | -0.963 | 1.27 | 1 |

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

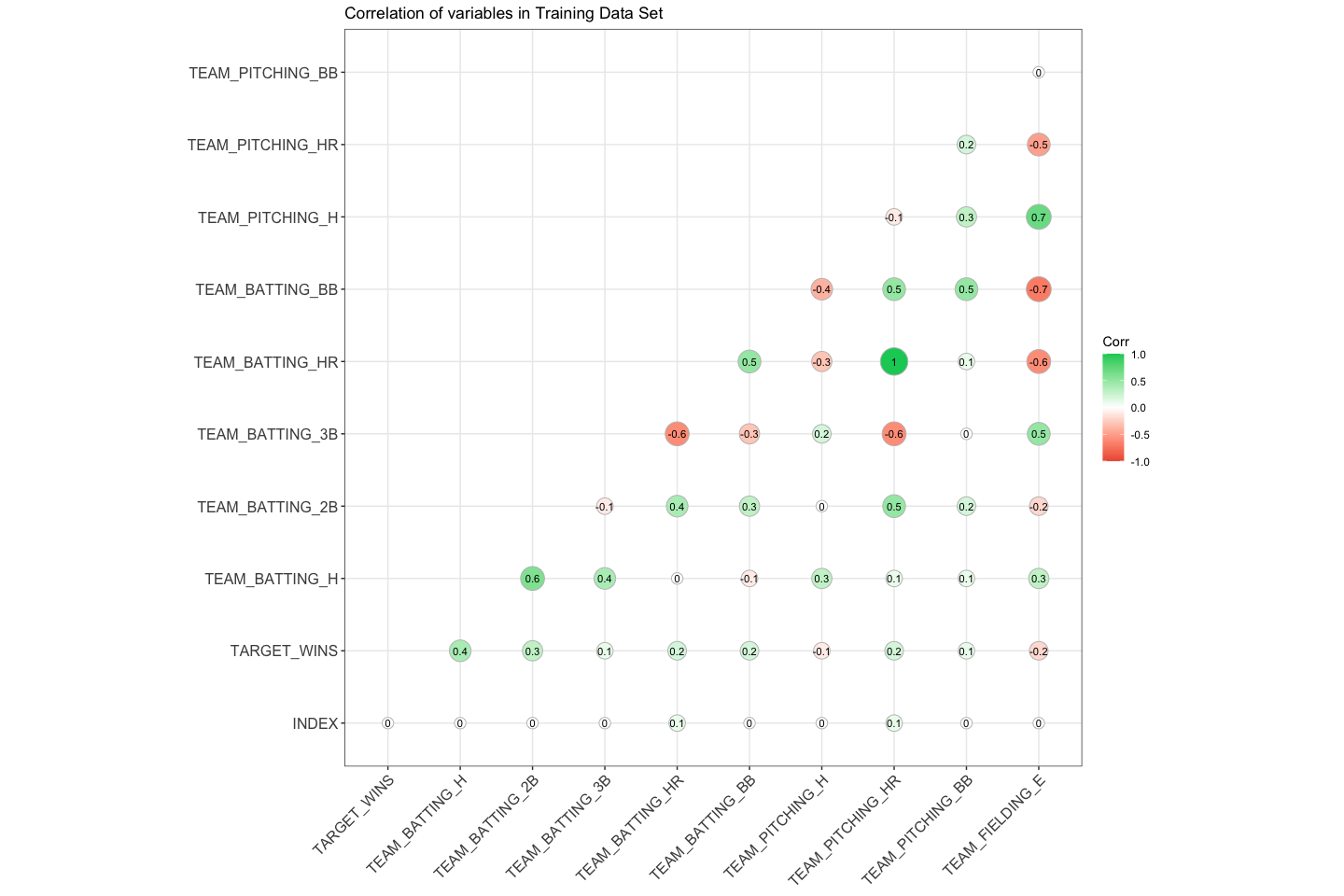
# Percentage of missing values  
missing\_values <- metastats %>%  
 filter(pct\_missing < 1) %>%  
 dplyr::select(STATS, pct\_missing) %>%  
 arrange(pct\_missing)  
missing\_values

|  |  |
| --- | --- |
| **STATS** | **pct\_missing** |
| TEAM\_BATTING\_HBP | 0.084 |
| TEAM\_BASERUN\_CS | 0.661 |
| TEAM\_FIELDING\_DP | 0.874 |
| TEAM\_BASERUN\_SB | 0.942 |
| TEAM\_BATTING\_SO | 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  
baseball\_train$TARGET\_WINS<-as.numeric(baseball\_train$TARGET\_WINS)  
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)



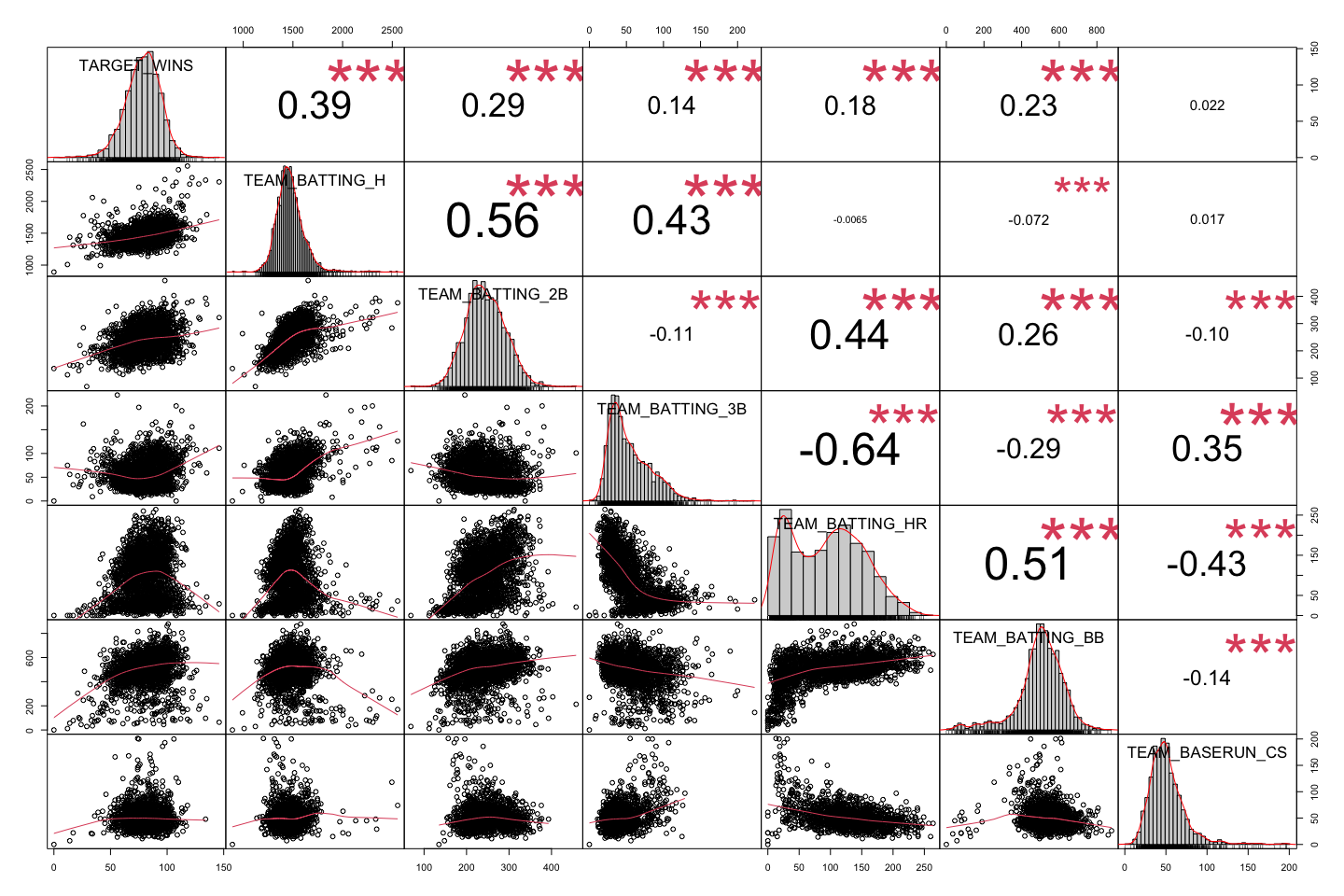
Team\_Batting\_H and Team\_Batting\_2B have the strongest positive correlation with Target\_Wins. We also see that, there is a strong correlation between Team\_Batting\_H and Team\_Batting\_2B, Team\_Pitching\_B and TEAM\_FIELDING\_E. We will consider these findings on model creation as collinearity 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

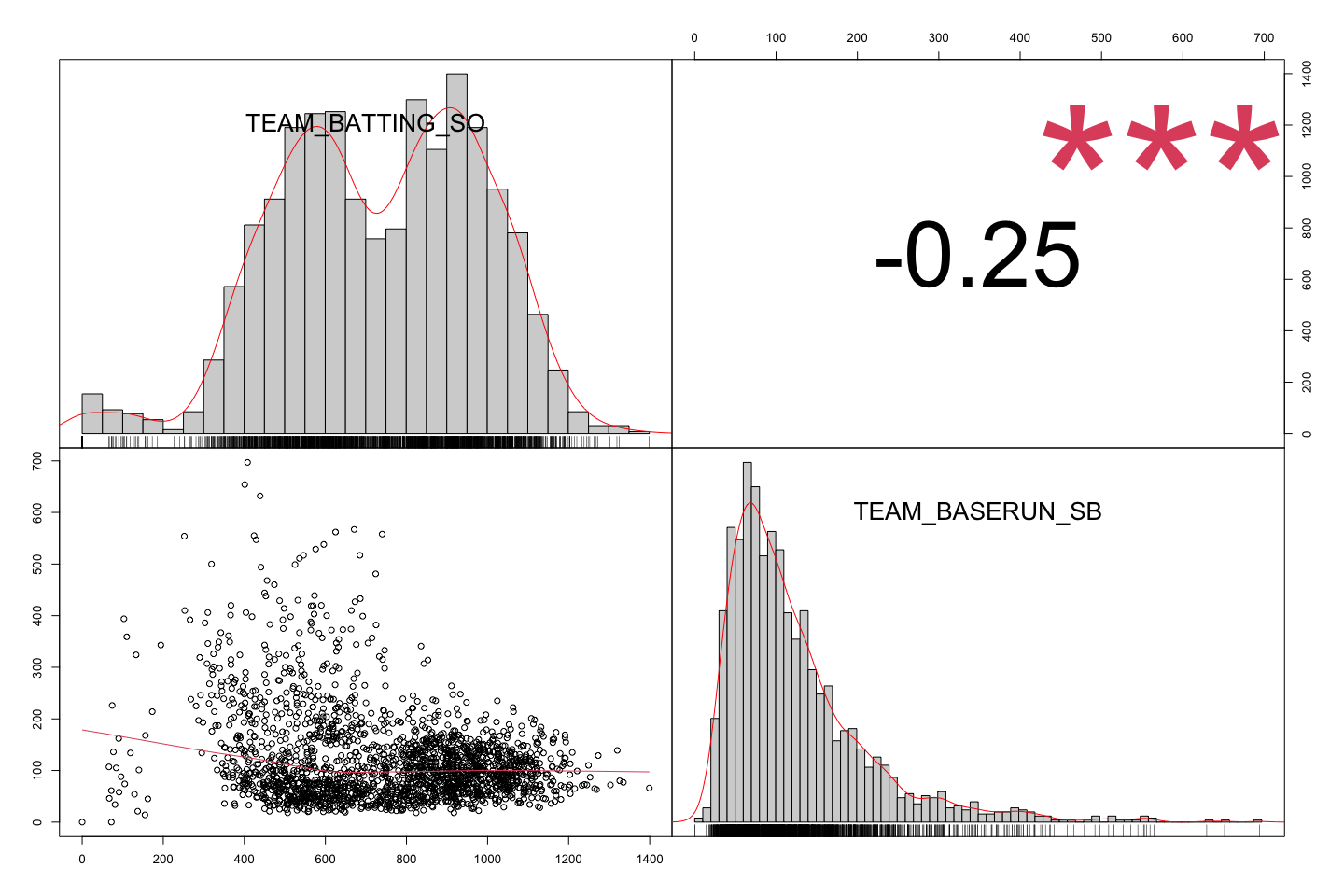
# Batting Correlations  
chart.Correlation(Batting\_df, histogram=TRUE, pch=19)



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

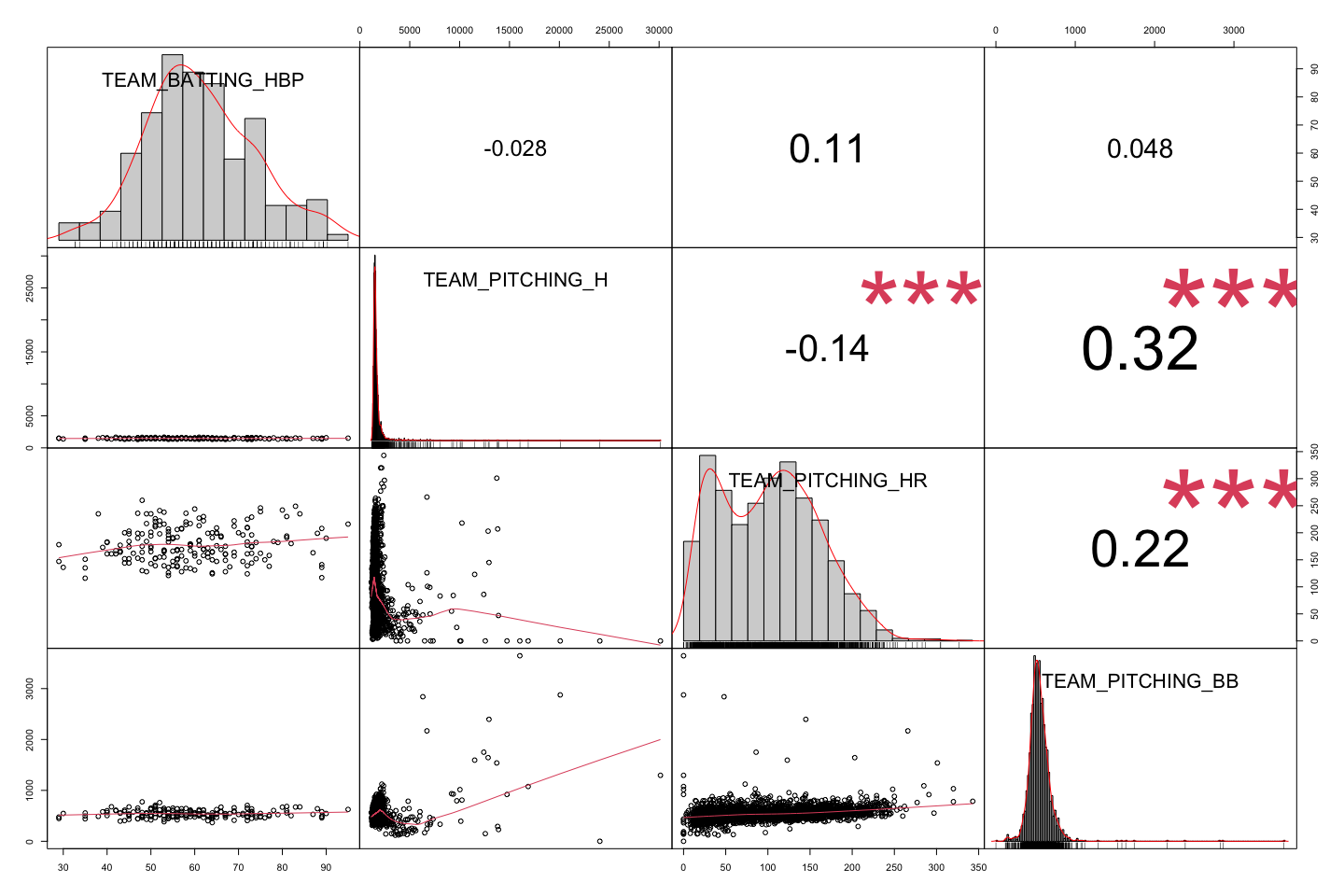
# baserunning Correlation  
  
chart.Correlation(BaseRunning\_df, histogram=TRUE, pch=19)



TEAM\_BASERUN\_SB is right skewed and TEAM\_BATTING\_SO is bimodal.

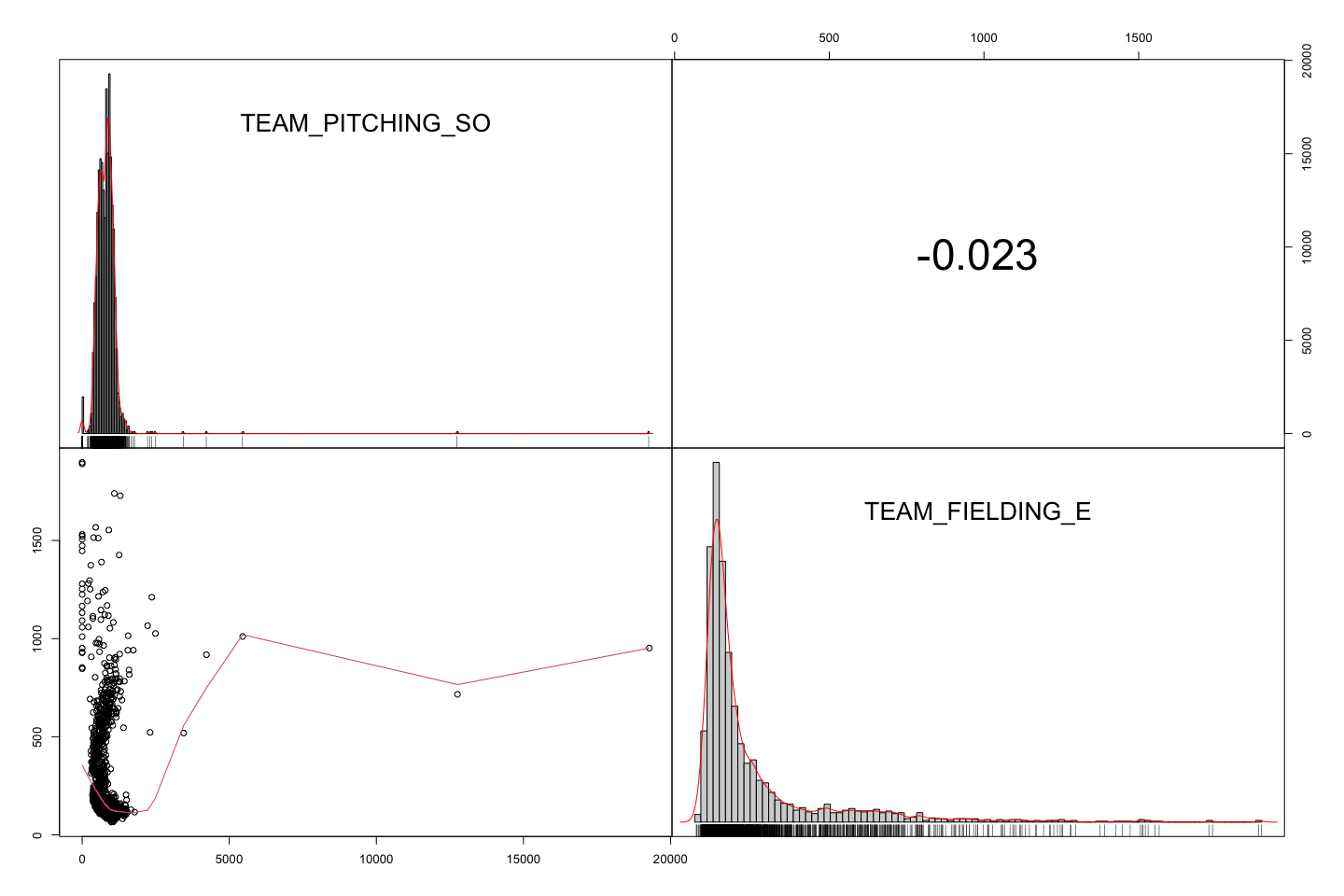
#### Pitching

#pitching correlations  
chart.Correlation(Pitching\_df, histogram=TRUE, pch=19)



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.

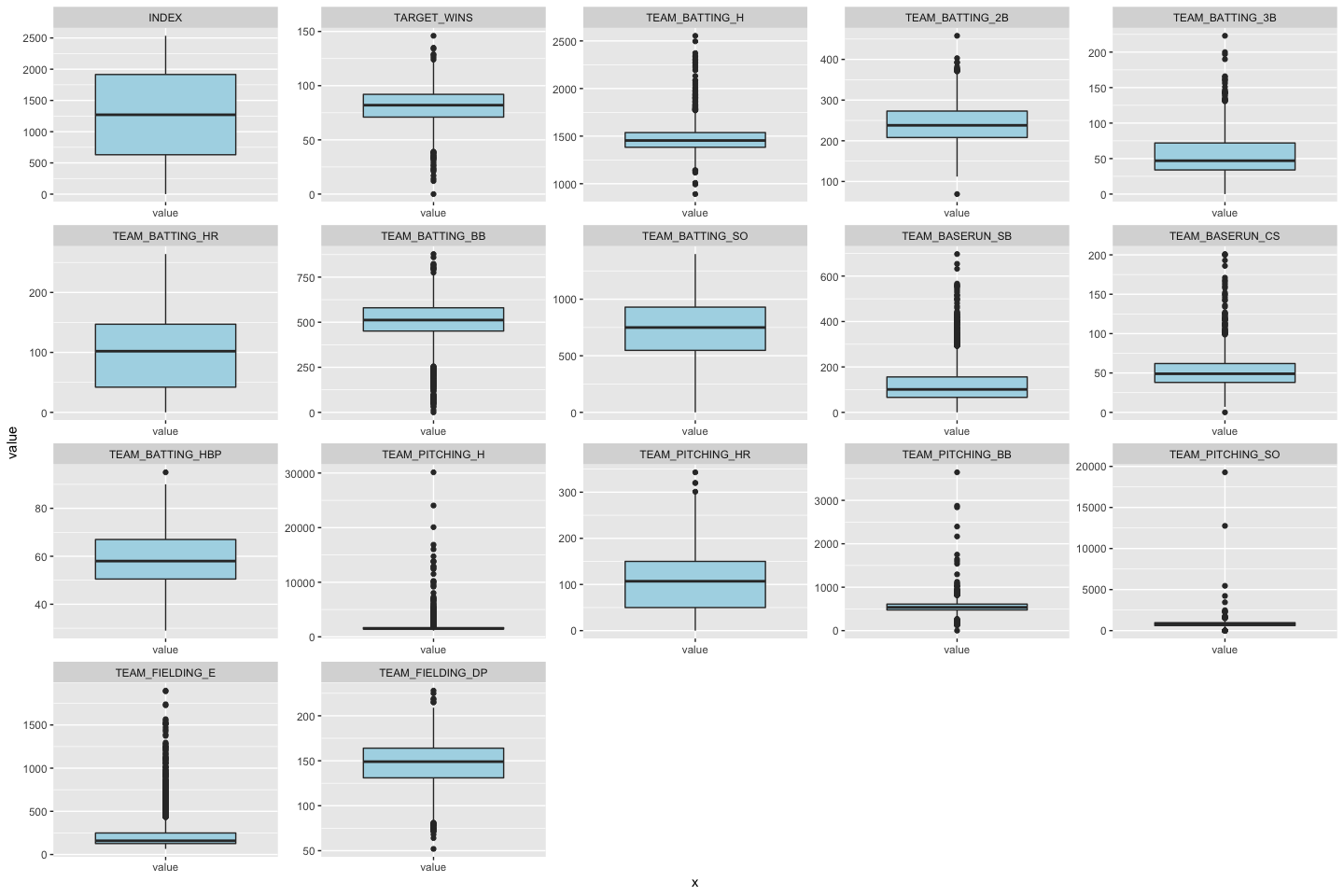
### Outliers and Skewness

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).



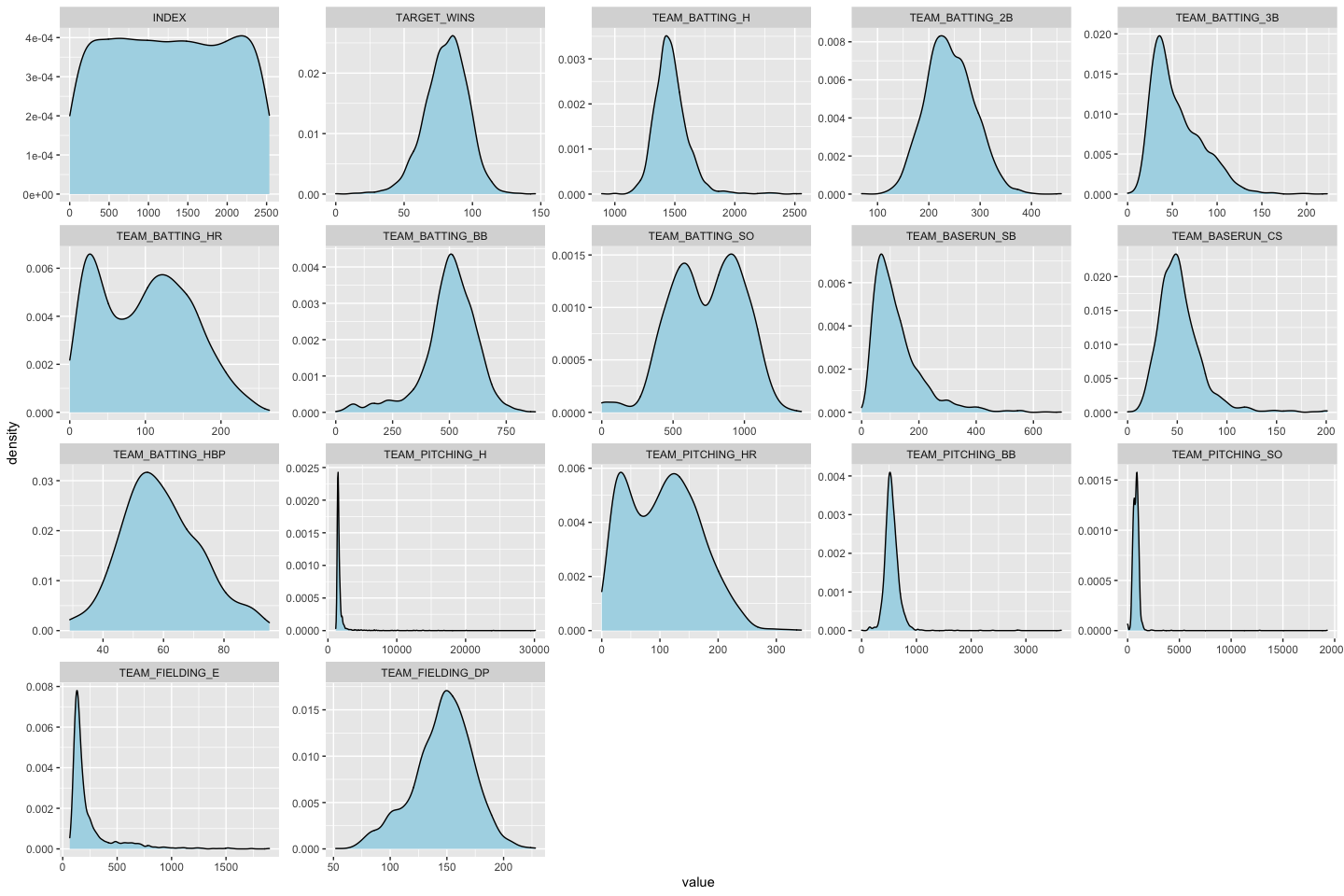
Based on the boxplot we created, TEAM\_FIELDING\_DP, TEAM\_PITCHING\_HR, TEAM\_BATTING\_HR and TEAM\_BATTING\_SO seem to have the least amount of outliers.

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\_SO | 22.2 |
| TEAM\_PITCHING\_H | 10.3 |
| TEAM\_PITCHING\_BB | 6.74 |
| TEAM\_FIELDING\_E | 2.99 |
| TEAM\_BASERUN\_CS | 1.98 |
| TEAM\_BASERUN\_SB | 1.97 |
| TEAM\_BATTING\_H | 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.

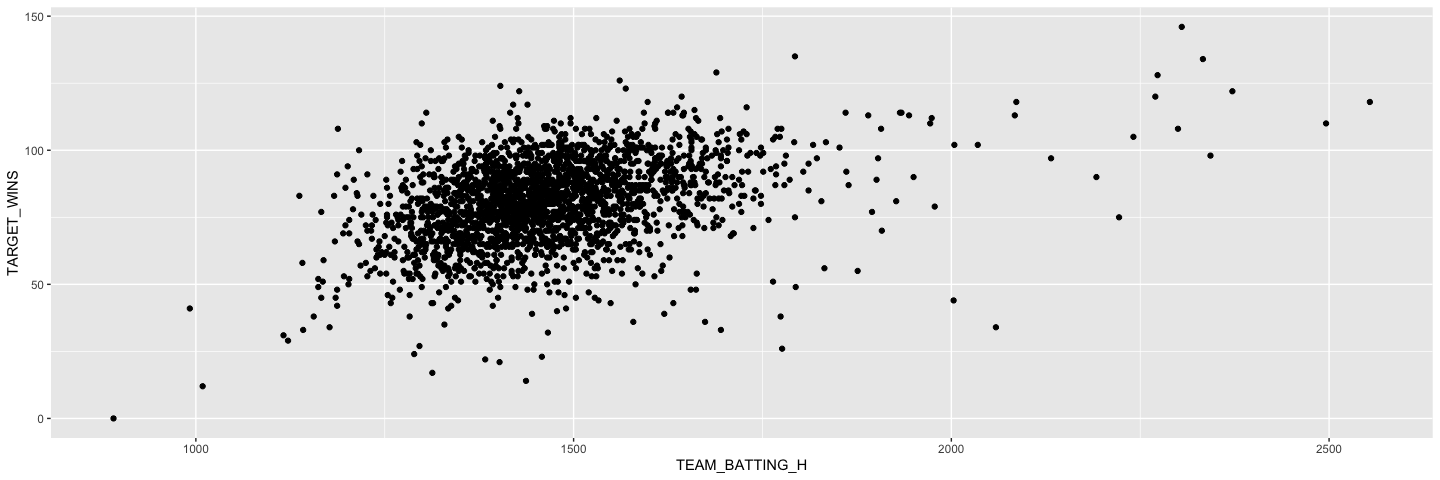
When we are creating a linear regression model, we are looking for the fitting line with the least sum of squares, that has the small residuals with minimized squared residuals. From our correlation analysis, we can see that the explatory variable that has the strongest correlation with TARGET\_WINS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

### Simple Model Example

#library(statsr)  
# line that follows the best assocation between two variables  
  
#plot\_ss(x = TEAM\_BATTING\_H, y = TARGET\_WINS, data=baseball\_train, showSquares = TRUE, leastSquares = TRUE)

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.

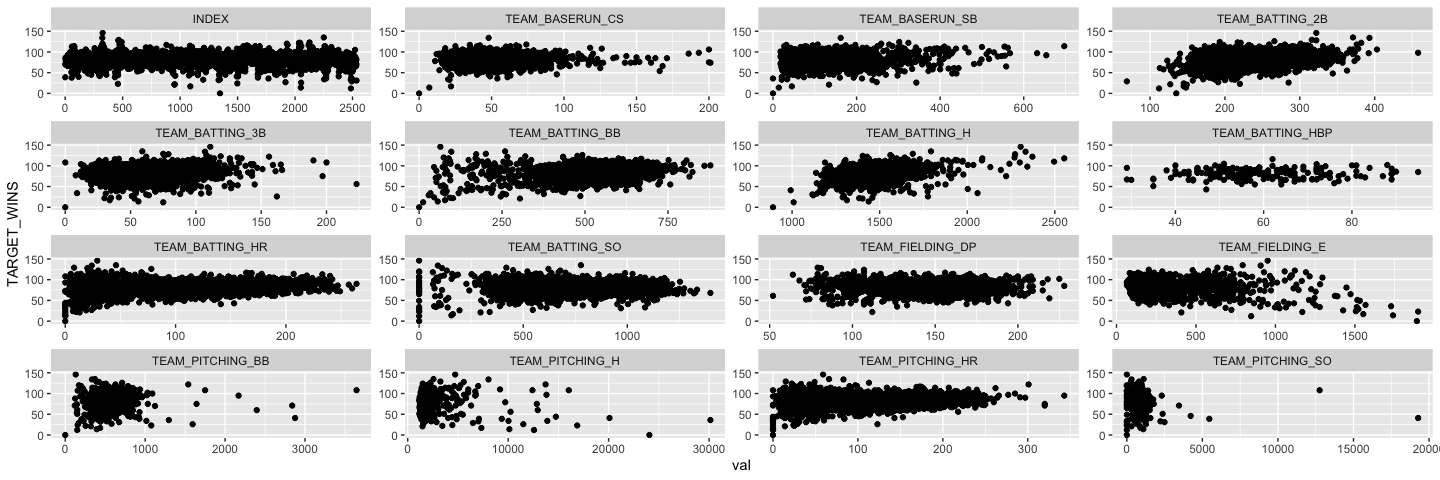
# scatter plot between TEAM\_BATTING\_H and TARGET\_WINS  
  
ggplot(baseball\_train, aes(x=TEAM\_BATTING\_H, y=TARGET\_WINS))+  
 geom\_point()



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).



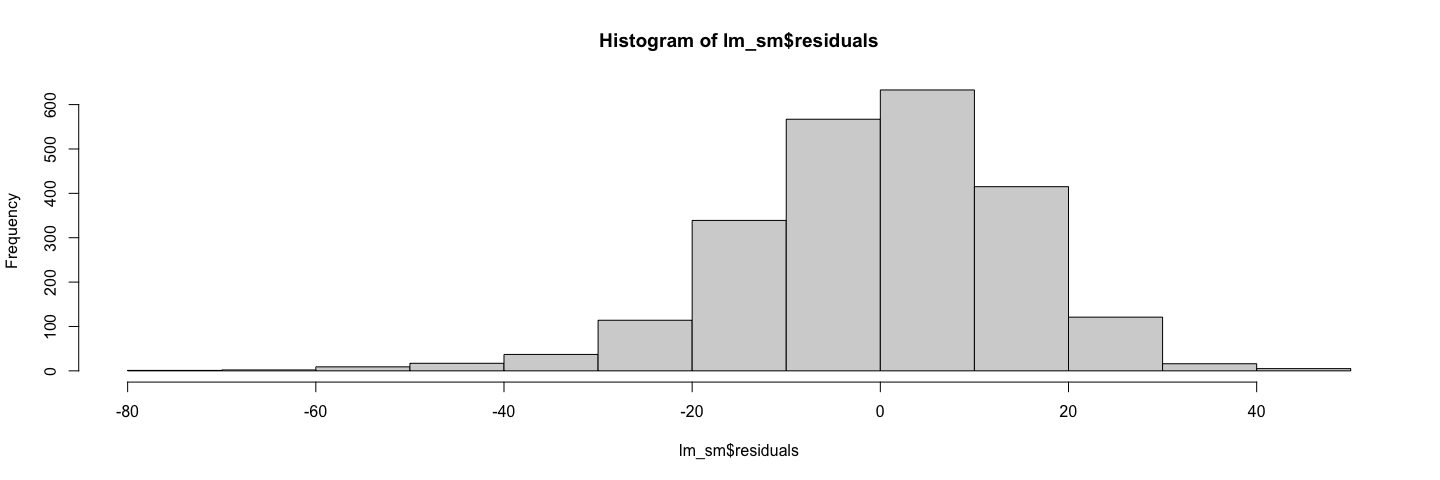
As we displayed earlier, hits walks and home runs have the strongest correlations with TARGET\_WINS and also meets the linearity condition.

# create a simple example model  
lm\_sm <- lm(baseball\_train$TARGET\_WINS ~ baseball\_train$TEAM\_BATTING\_H)  
summary(lm\_sm)

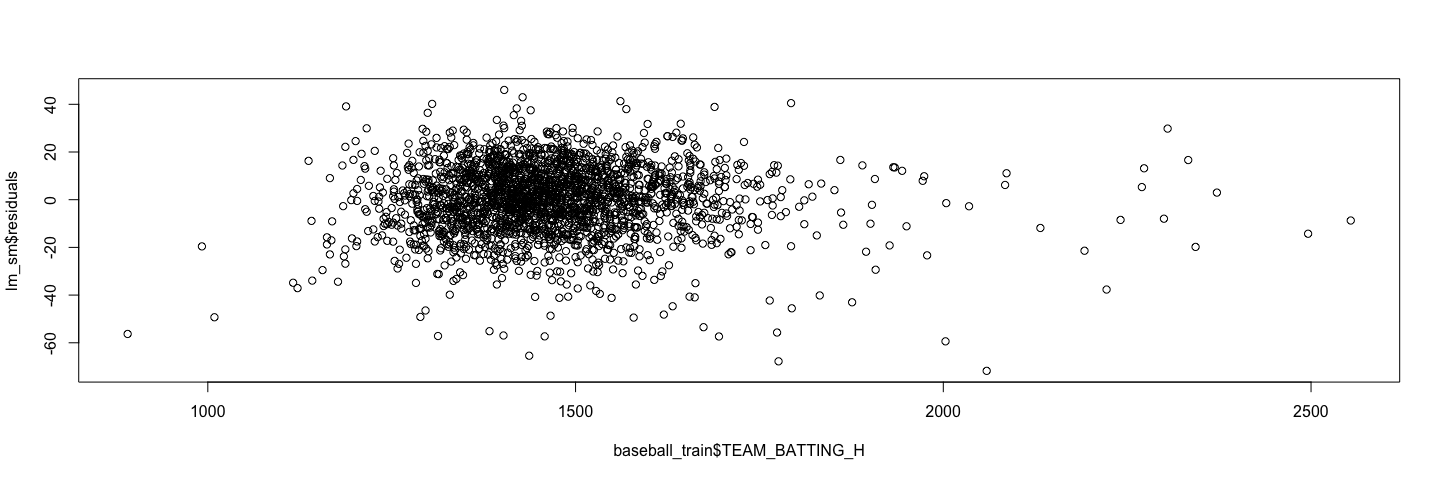
##   
## Call:  
## lm(formula = baseball\_train$TARGET\_WINS ~ baseball\_train$TEAM\_BATTING\_H)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -71.768 -8.757 0.856 9.762 46.016   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 18.562326 3.107523 5.973 2.69e-09 \*\*\*  
## 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\_BATTING\_HBP | 0.084 |
| TEAM\_BASERUN\_CS | 0.661 |
| TEAM\_FIELDING\_DP | 0.874 |
| TEAM\_BASERUN\_SB | 0.942 |
| TEAM\_BATTING\_SO | 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

|  |  |
| --- | --- |
| **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

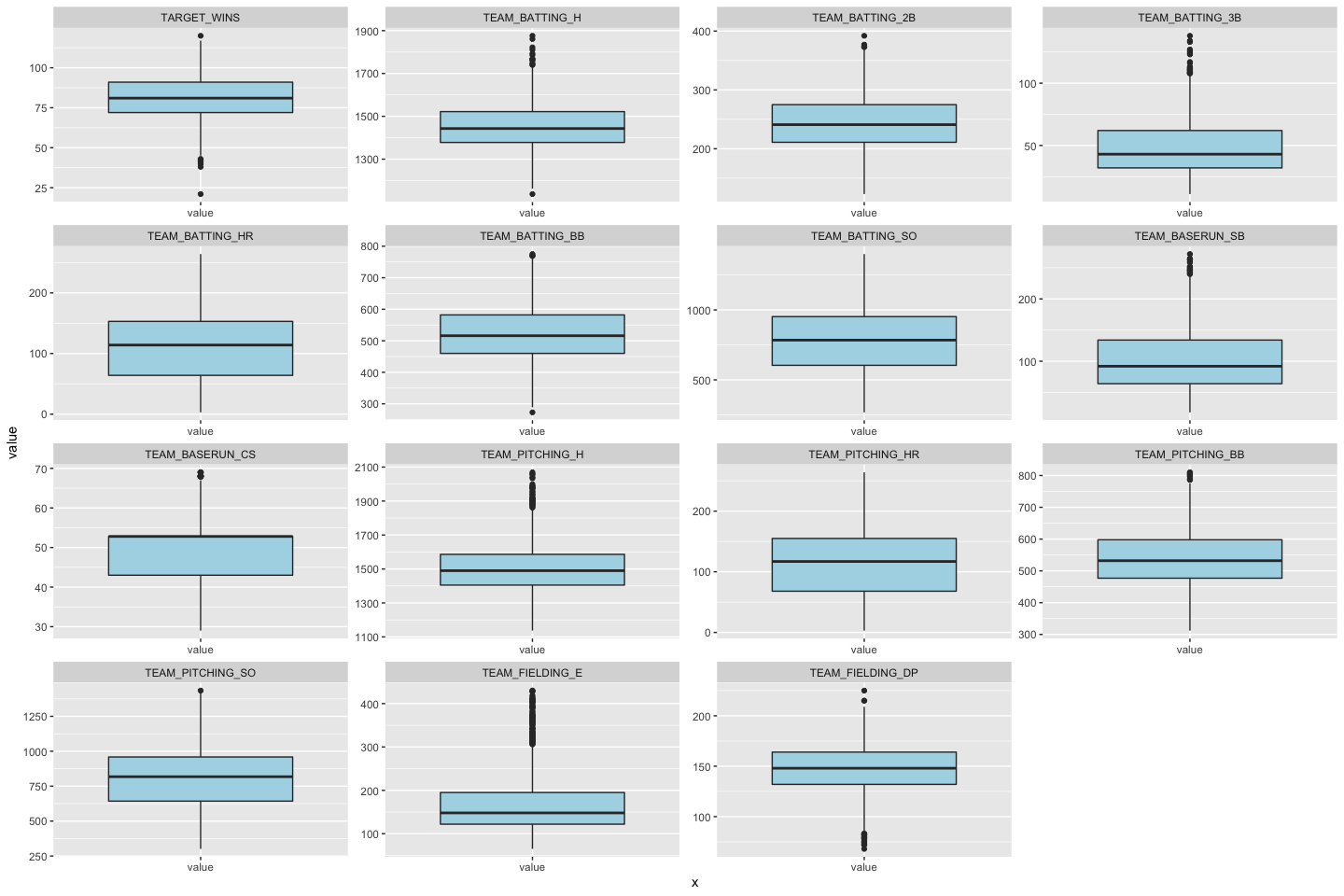
baseball\_train\_02 <- baseball\_train\_01 %>%  
 filter(  
 TEAM\_BASERUN\_SB\_Outlier != "Outlier" &  
 TEAM\_BASERUN\_CS\_Outlier != "Outlier" &  
 TEAM\_PITCHING\_H\_Outlier != "Outlier" &  
 TEAM\_PITCHING\_BB\_Outlier != "Outlier" &  
 TEAM\_PITCHING\_SO\_Outlier != "Outlier" &  
 TEAM\_FIELDING\_E\_Outlier != "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\_Outlier, TEAM\_PITCHING\_SO\_Outlier, TEAM\_FIELDING\_E\_Outlier as id variables

suppressWarnings(ggplot(datasub\_1, aes(x= "value", y=value)) +   
 geom\_boxplot(fill='lightblue') + facet\_wrap(~variable, scales = 'free') )



### Adjust Skewed values

#### Box Cox Transformation

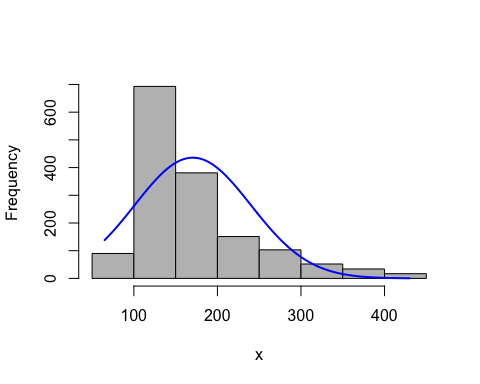
Removing the outliers tranformed each variable to a closer to a normal distribution and checking the skewness of the variables confirm this with the exception of TEAM\_FIELDING\_E. This variable is still skewed and not normal. In this section, we will use the Box Cox tranformation from the MASS library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

metastats\_02 <- data.frame(describe(baseball\_train\_02))  
metastats\_02 <- tibble::rownames\_to\_column(metastats\_02, "STATS")   
   
metastats\_02 %>%  
 filter(skew > 1 | skew < -1) %>%  
 dplyr::select(STATS, skew) %>%  
 arrange(desc(skew))

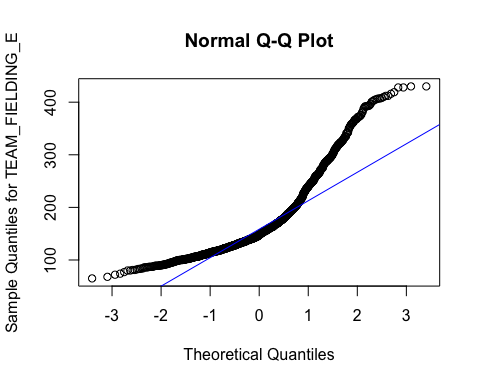
|  |  |
| --- | --- |
| **STATS** | **skew** |
| TEAM\_FIELDING\_E | 1.45 |

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")

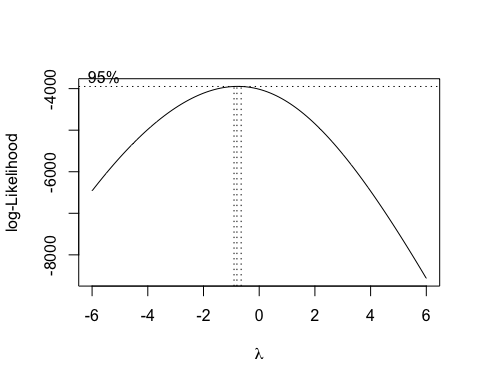


The following Box Cox transformation section is based on the tutorial at the link below:

[://rcompanion.org/handbook/I\_12.html][Summary and Analysis of Extension Program Evaluation in R]

The Box Cox procedure uses a log-likelihood to find the lambda to use to transform a variable to a normal distribution.

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))



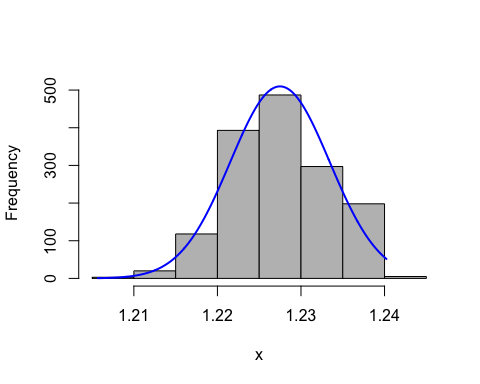
#Creates a dataframe with results  
Cox = data.frame(Box$x, Box$y)  
  
# Order the new data frame by decreasing y to find the best lambda.Displays the lambda with the greatest log likelihood.  
Cox2 = Cox[with(Cox, order(-Cox$Box.y)),]  
Cox2[1,]

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

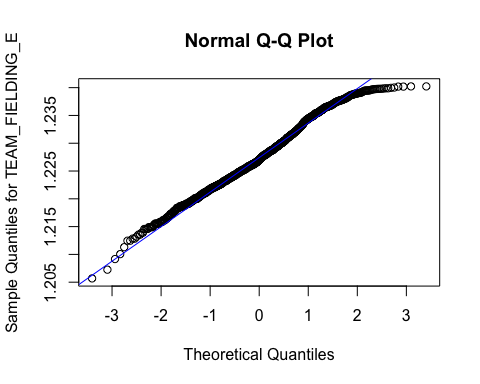
#Extract that lambda and Transform the data  
lambda = Cox2[1, "Box.x"]  
T\_box = (TEAM\_FIELDING\_E ^ lambda - 1)/lambda

We can now see that TEAM\_FIELDING\_E has a normal distribution.

plotNormalHistogram(T\_box)



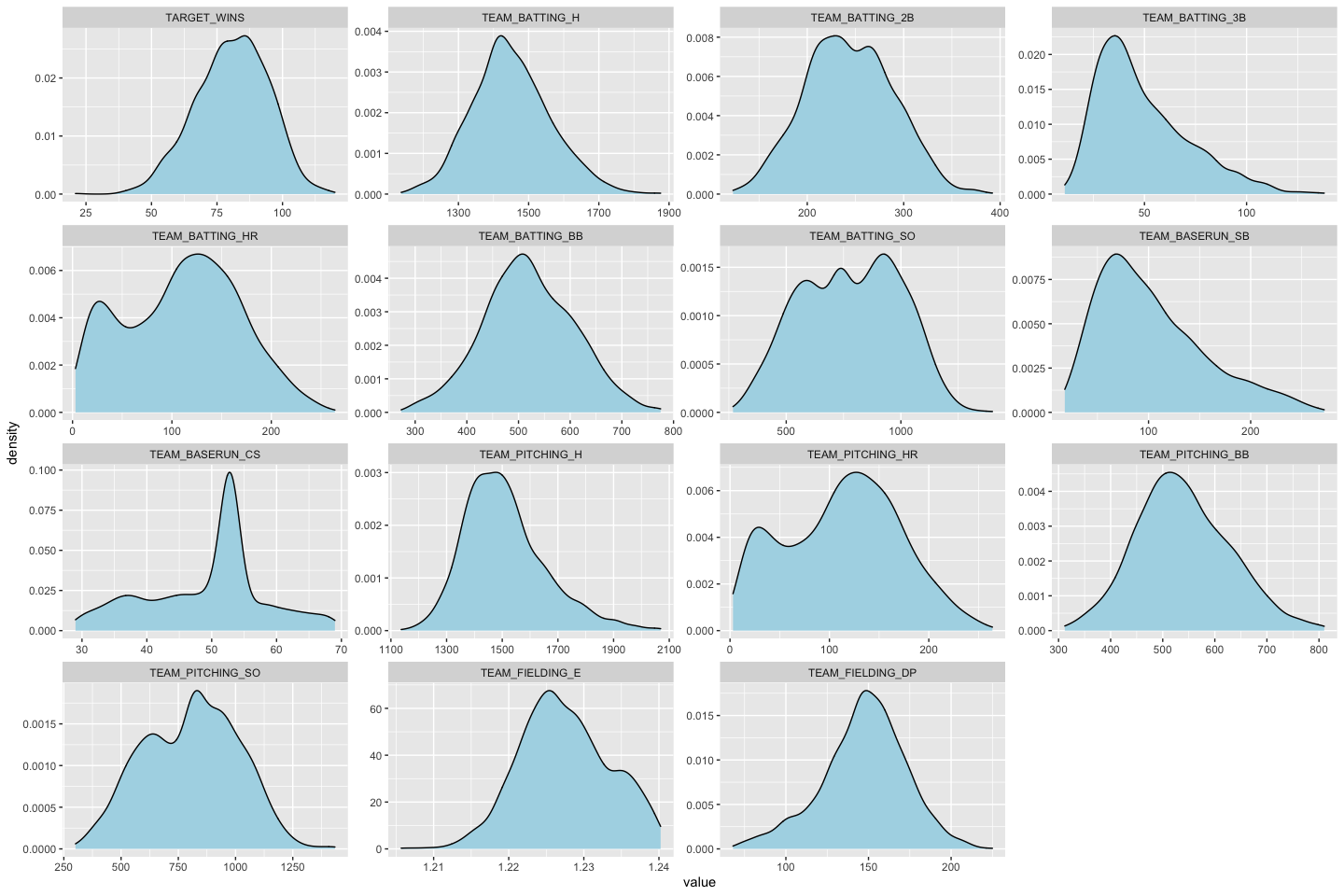
qqnorm(T\_box, ylab="Sample Quantiles for TEAM\_FIELDING\_E")  
qqline(T\_box,  
 col="blue")



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 : num 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 : num 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 = 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\_PITCHING\_SO + TEAM\_BATTING\_SO + TEAM\_BATTING\_2B,data=baseball\_train\_03)  
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 = 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\_BATTING\_SO + TEAM\_BASERUN\_CS + TEAM\_BATTING\_3B + TEAM\_BASERUN\_SB,data=baseball\_train\_03)  
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 = 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 + TEAM\_FIELDING\_E,data=baseball\_train\_03)  
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 = 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 + TEAM\_BATTING\_3B + TEAM\_BASERUN\_SB,data=baseball\_train\_03)  
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 = 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 = lambdas)   
##   
## Df %Dev Lambda  
## 1 15 30.27 100.000  
## 2 15 35.20 79.430  
## 3 15 40.50 63.100  
## 4 15 46.09 50.120  
## 5 15 51.86 39.810  
## 6 15 57.69 31.620  
## 7 15 63.46 25.120  
## 8 15 69.03 19.950  
## 9 15 74.26 15.850  
## 10 15 79.05 12.590  
## 11 15 83.31 10.000  
## 12 15 86.99 7.943  
## 13 15 90.07 6.310  
## 14 15 92.57 5.012  
## 15 15 94.55 3.981  
## 16 15 96.07 3.162  
## 17 15 97.21 2.512  
## 18 15 98.05 1.995  
## 19 15 98.66 1.585  
## 20 15 99.09 1.259  
## 21 15 99.39 1.000  
## 22 15 99.59 0.794  
## 23 15 99.73 0.631  
## 24 15 99.82 0.501  
## 25 15 99.88 0.398  
## 26 15 99.93 0.316  
## 27 15 99.95 0.251  
## 28 15 99.97 0.200  
## 29 15 99.98 0.158  
## 30 15 99.99 0.126  
## 31 15 99.99 0.100  
## 32 15 99.99 0.079  
## 33 15 100.00 0.063  
## 34 15 100.00 0.050  
## 35 15 100.00 0.040  
## 36 15 100.00 0.032  
## 37 15 100.00 0.025  
## 38 15 100.00 0.020  
## 39 15 100.00 0.016  
## 40 15 100.00 0.013  
## 41 15 100.00 0.010  
## 42 15 100.00 0.008  
## 43 15 100.00 0.006  
## 44 15 100.00 0.005  
## 45 15 100.00 0.004  
## 46 15 100.00 0.003  
## 47 15 100.00 0.003  
## 48 15 100.00 0.002  
## 49 15 100.00 0.002  
## 50 15 100.00 0.001  
## 51 15 100.00 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.

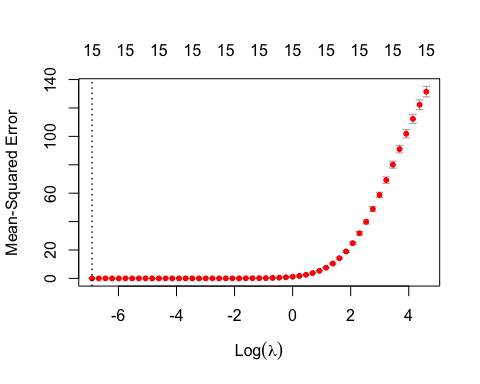
CrossVal\_ridge <- cv.glmnet(train\_Ind,train\_Dep, alpha = 0, lambda = lambdas)  
optimal\_lambda <- CrossVal\_ridge$lambda.min  
optimal\_lambda #The optimal lambda is 0.001 which we will use to penelize the Ridge Regression model.

## [1] 0.001

coef(CrossVal\_ridge) # Shows the coefficients

## 16 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 1.902084e-01  
## TARGET\_WINS 9.998800e-01  
## TEAM\_BATTING\_H -1.030075e-06  
## TEAM\_BATTING\_2B -9.035245e-06  
## TEAM\_BATTING\_3B 1.008225e-05  
## TEAM\_BATTING\_HR -5.142774e-04  
## TEAM\_BATTING\_BB 1.522053e-05  
## TEAM\_BATTING\_SO 6.134586e-05  
## TEAM\_BASERUN\_SB 1.642783e-05  
## TEAM\_BASERUN\_CS -2.630848e-05  
## TEAM\_PITCHING\_H 5.093995e-06  
## TEAM\_PITCHING\_HR 5.020784e-04  
## TEAM\_PITCHING\_BB -1.092104e-05  
## TEAM\_PITCHING\_SO -6.162997e-05  
## TEAM\_FIELDING\_E -1.483976e-01  
## TEAM\_FIELDING\_DP -2.340585e-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 predictability, 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 = 6.310)  
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) 1.771139e+02  
## TARGET\_WINS 6.227634e-01  
## TEAM\_BATTING\_H 6.621183e-03  
## TEAM\_BATTING\_2B 6.708282e-04  
## TEAM\_BATTING\_3B 2.313006e-02  
## TEAM\_BATTING\_HR 8.597721e-03  
## TEAM\_BATTING\_BB 7.476724e-03  
## TEAM\_BATTING\_SO -1.108036e-03  
## TEAM\_BASERUN\_SB 1.300729e-02  
## TEAM\_BASERUN\_CS -7.890965e-03  
## TEAM\_PITCHING\_H 1.548065e-03  
## TEAM\_PITCHING\_HR 7.802634e-03  
## TEAM\_PITCHING\_BB 5.189471e-03  
## TEAM\_PITCHING\_SO -1.780654e-03  
## TEAM\_FIELDING\_E -1.338565e+02  
## TEAM\_FIELDING\_DP -1.897631e-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** |
| 4.32 | 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** |
| 4.52 | 0.898 |

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 82 69.49875  
## 2 80 67.56125  
## 3 85 68.38378  
## 4 78 73.25628  
## 5 87 67.37030  
## 6 81 66.86382  
## 7 91 63.36361  
## 8 87 76.58169  
## 9 85 89.59170  
## 10 70 76.54657  
## 11 82 86.70081  
## 12 75 77.31870  
## 13 92 78.60647  
## 14 92 84.73255  
## 15 98 89.15530  
## 16 51 86.01142  
## 17 111 76.15138  
## 18 86 76.18203  
## 19 55 79.02243  
## 20 74 72.03653  
## 21 66 89.30377  
## 22 70 84.16328  
## 23 63 79.30964  
## 24 81 76.00667  
## 25 87 93.36918  
## 26 90 79.72780  
## 27 80 83.96472  
## 28 87 81.83322  
## 29 84 86.75976  
## 30 88 83.85432  
## 31 86 75.24767  
## 32 81 96.12431  
## 33 76 85.18416  
## 34 70 87.23777  
## 35 67 86.66397  
## 36 82 80.90673  
## 37 65 71.73633  
## 38 95 81.22686  
## 39 90 90.64517  
## 40 83 94.18601  
## 41 88 95.53100  
## 42 66 80.08285  
## 43 56 75.86419  
## 44 66 66.74352  
## 45 45 64.57578  
## 46 69 60.91243  
## 47 61 72.28834  
## 48 97 68.48019  
## 49 84 65.67250  
## 50 97 72.48471  
## 51 54 86.44011  
## 52 76 80.65902  
## 53 85 82.06254  
## 54 74 78.70595  
## 55 104 77.58369  
## 56 94 72.97126  
## 57 107 78.18561  
## 58 100 72.24895  
## 59 64 76.25716  
## 60 94 77.02116  
## 61 45 75.30225  
## 62 68 76.46272  
## 63 78 78.36297  
## 64 87 68.74638  
## 65 80 70.51741  
## 66 92 66.00240  
## 67 85 74.32965  
## 68 84 68.02212  
## 69 98 71.72855  
## 70 84 69.41157  
## 71 88 69.61566  
## 72 97 70.79319  
## 73 78 69.48860  
## 74 85 65.80599  
## 75 95 70.70940  
## 76 75 82.73716  
## 77 62 77.23526  
## 78 114 81.04205  
## 79 110 86.68577  
## 80 98 81.16218  
## 81 86 88.42305  
## 82 88 85.99275  
## 83 87 73.58515  
## 84 64 82.22049  
## 85 68 85.06236  
## 86 65 86.09993  
## 87 89 76.29960  
## 88 75 79.91231  
## 89 81 77.53938  
## 90 79 84.34369  
## 91 80 74.79848  
## 92 71 70.02759  
## 93 76 82.39423  
## 94 77 79.28918  
## 95 82 91.97048  
## 96 65 78.96951  
## 97 79 74.19733  
## 98 93 72.42719  
## 99 83 77.14336  
## 100 83 76.15731  
## 101 63 77.37089  
## 102 53 72.05640  
## 103 56 76.89373  
## 104 79 75.92414  
## 105 90 85.81217  
## 106 71 75.59281  
## 107 77 80.60366  
## 108 78 85.62160  
## 109 54 82.00590  
## 110 94 84.74713  
## 111 99 84.88357  
## 112 86 86.74637  
## 113 99 89.85103  
## 114 92 88.78040  
## 115 95 78.46174  
## 116 67 84.86044  
## 117 79 83.42672  
## 118 75 82.79613  
## 119 64 82.72617  
## 120 90 66.91725  
## 121 74 69.51332  
## 122 71 78.66910  
## 123 87 72.33942  
## 124 71 67.50771  
## 125 77 64.77165  
## 126 80 69.79720  
## 127 75 71.15595  
## 128 95 77.34661  
## 129 93 77.95811  
## 130 69 86.54917  
## 131 77 73.89007  
## 132 63 79.45996  
## 133 64 76.41158  
## 134 111 81.88671  
## 135 62 84.23277  
## 136 62 81.62631  
## 137 78 78.66292  
## 138 102 78.01240  
## 139 93 82.19463  
## 140 94 84.27986  
## 141 64 73.82318  
## 142 77 75.50558  
## 143 73 72.20240  
## 144 79 71.73169  
## 145 80 78.70550  
## 146 70 82.60550  
## 147 86 79.89101  
## 148 87 80.98410  
## 149 108 73.62353  
## 150 91 74.78823  
## 151 86 70.09769  
## 152 81 69.79510  
## 153 76 73.21664  
## 154 85 76.42604  
## 155 66 70.00951  
## 156 69 79.24643  
## 157 92 76.55529  
## 158 94 77.45710  
## 159 99 77.29383  
## 160 93 74.83120  
## 161 87 80.57564  
## 162 82 75.27052  
## 163 72 72.96984  
## 164 97 88.01300  
## 165 93 89.33125  
## 166 79 86.25568  
## 167 79 69.87604  
## 168 81 89.64940  
## 169 62 80.48480  
## 170 76 81.38282  
## 171 80 78.12840  
## 172 61 80.46360  
## 173 57 77.90473  
## 174 113 86.93266  
## 175 87 88.60549  
## 176 21 86.22276  
## 177 77 85.13130  
## 178 72 80.44713  
## 179 76 88.03475  
## 180 95 80.82213  
## 181 70 83.47961  
## 182 82 72.80089  
## 183 88 80.37391  
## 184 85 79.72336  
## 185 95 78.69135  
## 186 79 85.57211  
## 187 72 83.20429  
## 188 82 89.48194  
## 189 103 96.68228  
## 190 79 87.75298  
## 191 58 91.87313  
## 192 74 86.26651  
## 193 87 77.82142  
## 194 98 76.04487  
## 195 88 84.88891  
## 196 79 81.55068  
## 197 75 83.70493  
## 198 79 80.38773  
## 199 95 101.92399  
## 200 64 81.10336  
## 201 76 73.46680  
## 202 54 93.36972  
## 203 64 86.05010  
## 204 79 85.59259  
## 205 65 73.88114  
## 206 72 74.83380  
## 207 81 64.08681  
## 208 80 65.84094  
## 209 80 71.98560  
## 210 83 79.13560  
## 211 65 85.15069  
## 212 82 84.82851  
## 213 102 88.64454  
## 214 89 87.36139  
## 215 65 73.46671  
## 216 80 73.32881  
## 217 91 76.75376  
## 218 92 80.18206  
## 219 72 75.03975  
## 220 83 75.66433  
## 221 94 70.51948  
## 222 75 74.54246  
## 223 75 72.27715  
## 224 82 70.35686  
## 225 84 72.18585  
## 226 70 79.91554  
## 227 66 88.74324  
## 228 99 87.01201  
## 229 88 85.96054  
## 230 94 89.84882  
## 231 86 93.28619  
## 232 94 91.12115  
## 233 100 94.67752  
## 234 85 73.79792  
## 235 88 80.75316  
## 236 89 76.99427  
## 237 96 87.39336  
## 238 88 77.79257  
## 239 95 75.87232  
## 240 73 87.87867  
## 241 95 79.02455  
## 242 78 79.53593  
## 243 90 77.39581  
## 244 88 71.37449  
## 245 77 85.39949  
## 246 86 78.33072  
## 247 64 86.04030  
## 248 69 79.55776  
## 249 96 82.75494  
## 250 71 86.19625  
## 251 78 87.67022  
## 252 56 85.00793  
## 253 67 80.45297  
## 254 81 87.60385  
## 255 75 82.44495  
## 256 45 77.50487  
## 257 73 79.20244  
## 258 89 69.31674  
## 259 70 74.80593  
## 260 69 83.14847  
## 261 67 91.63208  
## 262 67 93.48635  
## 263 67 84.53632  
## 264 66 86.35405  
## 265 77 86.94776  
## 266 75 82.27458  
## 267 81 80.34078  
## 268 85 97.84021  
## 269 77 90.77786  
## 270 81 86.29480  
## 271 80 89.59173  
## 272 69 81.45834  
## 273 95 79.98754  
## 274 50 76.10745  
## 275 66 78.12297  
## 276 86 76.91669  
## 277 65 74.90087  
## 278 90 87.24509  
## 279 98 95.14666  
## 280 91 90.93669  
## 281 71 90.70110  
## 282 61 81.29394  
## 283 84 72.64682  
## 284 93 82.98819  
## 285 99 75.16897  
## 286 95 78.10723  
## 287 116 82.14880  
## 288 106 84.91555  
## 289 93 97.32176  
## 290 99 101.10846  
## 291 93 84.72247  
## 292 108 75.83547  
## 293 103 76.87742  
## 294 106 84.10901  
## 295 72 89.08411  
## 296 99 81.50902  
## 297 100 81.45233  
## 298 99 78.28904  
## 299 103 74.25636  
## 300 79 75.53547  
## 301 91 78.88184  
## 302 89 77.90593  
## 303 86 81.68807  
## 304 74 76.22767  
## 305 92 71.70319  
## 306 96 74.06627  
## 307 96 74.02593  
## 308 101 79.72855  
## 309 87 78.32084  
## 310 87 75.75259  
## 311 101 83.80749  
## 312 110 79.14016  
## 313 108 77.77473  
## 314 66 77.15735  
## 315 56 76.99388  
## 316 74 68.85761  
## 317 104 74.32405  
## 318 63 76.06391  
## 319 56 76.38482  
## 320 56 77.40995  
## 321 52 77.07957  
## 322 74 81.59777  
## 323 66 73.66385  
## 324 59 79.18950  
## 325 62 77.63615  
## 326 82 79.20131  
## 327 54 79.58513  
## 328 77 78.45513  
## 329 76 85.18691  
## 330 78 87.36482  
## 331 74 81.06456  
## 332 102 86.55401  
## 333 69 76.85508  
## 334 91 80.08373  
## 335 100 80.76698  
## 336 87 78.96028  
## 337 83 81.18968  
## 338 84 84.35796  
## 339 78 78.91002  
## 340 93 75.96436  
## 341 46 85.42552  
## 342 82 80.55205  
## 343 68 84.56468  
## 344 45 82.79470  
## 345 92 76.50017  
## 346 49 78.80238  
## 347 82 81.74452  
## 348 76 81.81584  
## 349 63 83.28464  
## 350 61 75.40413  
## 351 80 74.86536  
## 352 70 75.32920  
## 353 77 70.05289  
## 354 65 72.91093  
## 355 105 73.69728  
## 356 92 84.47503  
## 357 103 72.99975  
## 358 91 83.43029  
## 359 85 87.95245  
## 360 73 78.31703  
## 361 89 78.64492  
## 362 99 81.71404  
## 363 90 80.02600  
## 364 91 77.59988  
## 365 88 79.43621  
## 366 72 79.28112  
## 367 82 83.90819  
## 368 85 93.54258  
## 369 73 86.40102  
## 370 95 82.58538  
## 371 75 85.89380  
## 372 53 79.81702  
## 373 69 76.95876  
## 374 94 70.17323  
## 375 74 74.13669  
## 376 80 72.30131  
## 377 88 75.25157  
## 378 92 76.54545  
## 379 98 78.41872  
## 380 57 81.53011  
## 381 95 85.93305  
## 382 98 83.95041  
## 383 75 88.37161  
## 384 73 86.31066  
## 385 72 82.44512  
## 386 67 83.91708  
## 387 63 86.67909  
## 388 60 87.44296  
## 389 81 82.21262  
## 390 92 76.52642  
## 391 89 75.16841  
## 392 91 64.24604  
## 393 74 72.58621  
## 394 66 77.46501  
## 395 88 72.80068  
## 396 57 78.05726  
## 397 67 80.25288  
## 398 73 76.11026  
## 399 83 90.32384  
## 400 88 74.10965  
## 401 76 79.40059  
## 402 116 70.55224  
## 403 78 86.59573  
## 404 103 87.70315  
## 405 88 83.05623  
## 406 79 78.62248  
## 407 92 75.43148  
## 408 97 83.51626  
## 409 58 74.53313  
## 410 70 75.06487  
## 411 83 86.07172  
## 412 73 87.84534  
## 413 89 94.14527  
## 414 90 96.04200  
## 415 80 90.90022  
## 416 75 86.34220  
## 417 86 94.79885  
## 418 90 86.76487  
## 419 92 87.56303  
## 420 42 85.09908  
## 421 68 88.22749  
## 422 61 93.78125  
## 423 58 90.14901  
## 424 82 97.39963  
## 425 82 88.67552  
## 426 97 87.20702  
## 427 106 80.27788  
## 428 76 85.93653  
## 429 87 74.39622  
## 430 101 74.75851  
## 431 97 72.61288  
## 432 103 90.56274  
## 433 112 89.47333  
## 434 89 78.69775  
## 435 83 77.35048  
## 436 93 79.53762  
## 437 80 83.39287  
## 438 83 83.62948  
## 439 97 76.57712  
## 440 69 77.10248  
## 441 72 83.34904  
## 442 86 78.56847  
## 443 57 69.57909  
## 444 83 73.96331  
## 445 88 63.06532  
## 446 72 69.99296  
## 447 86 76.64262  
## 448 96 79.98629  
## 449 87 73.50146  
## 450 89 77.92409  
## 451 86 82.13105  
## 452 80 77.90857  
## 453 71 72.95778  
## 454 76 69.73774  
## 455 86 69.46233  
## 456 82 67.52634  
## 457 65 72.19872  
## 458 67 78.59388  
## 459 68 73.35113  
## 460 82 71.86038  
## 461 80 75.50214  
## 462 85 70.66007  
## 463 78 73.52882  
## 464 87 88.14247  
## 465 81 88.25647  
## 466 91 81.56281  
## 467 87 85.11315  
## 468 85 80.24222  
## 469 70 81.24244  
## 470 82 82.69428  
## 471 75 73.51443  
## 472 92 75.48643  
## 473 92 88.59721  
## 474 98 74.75932  
## 475 51 69.56123  
## 476 111 88.86014  
## 477 86 71.28452  
## 478 55 86.22500  
## 479 74 88.95228  
## 480 66 92.69140  
## 481 70 101.10893  
## 482 63 93.85807  
## 483 81 84.70288  
## 484 87 82.88195  
## 485 90 84.57159  
## 486 80 73.01808  
## 487 87 78.50309  
## 488 84 87.77309  
## 489 88 90.41206  
## 490 86 81.01700  
## 491 81 81.18505  
## 492 76 78.76486  
## 493 70 79.22418  
## 494 67 89.33282  
## 495 82 81.16533  
## 496 65 79.74078  
## 497 95 92.39093  
## 498 90 89.28796  
## 499 83 77.12440  
## 500 88 79.77105  
## 501 66 75.68524  
## 502 56 87.43997  
## 503 66 76.52938  
## 504 45 84.78700  
## 505 69 80.50877  
## 506 61 77.35972  
## 507 97 72.72118  
## 508 84 80.34685  
## 509 97 72.55936  
## 510 54 70.06595  
## 511 76 83.25037  
## 512 85 80.59516  
## 513 74 81.71215  
## 514 104 78.13880  
## 515 94 82.04003  
## 516 107 88.64843  
## 517 100 84.59548  
## 518 64 90.67361  
## 519 94 84.94869  
## 520 45 82.43145  
## 521 68 77.98885  
## 522 78 72.89823  
## 523 87 83.43799  
## 524 80 86.92602  
## 525 92 87.18417  
## 526 85 77.09404  
## 527 84 89.40353  
## 528 98 93.34790  
## 529 84 86.96778  
## 530 88 82.14507  
## 531 97 83.43017  
## 532 78 75.10548  
## 533 85 74.03722  
## 534 95 74.16809  
## 535 75 94.69567  
## 536 62 85.56016  
## 537 114 93.39981  
## 538 110 85.25712  
## 539 98 87.29422  
## 540 86 87.46496  
## 541 88 83.61357  
## 542 87 79.45141  
## 543 64 79.45387  
## 544 68 78.43478  
## 545 65 73.38522  
## 546 89 76.03883  
## 547 75 72.47713  
## 548 81 66.23869  
## 549 79 67.67506  
## 550 80 74.29160  
## 551 71 76.15938  
## 552 76 68.53339  
## 553 77 64.18492  
## 554 82 76.55550  
## 555 65 72.31920  
## 556 79 75.70448  
## 557 93 86.25488  
## 558 83 87.13625  
## 559 83 85.66963  
## 560 63 75.01403  
## 561 53 72.48383  
## 562 56 76.99319  
## 563 79 67.65273  
## 564 90 78.88975  
## 565 71 85.81330  
## 566 77 99.38599  
## 567 78 102.30433  
## 568 54 93.87881  
## 569 94 100.37487  
## 570 99 98.32826  
## 571 86 89.11290  
## 572 99 78.88624  
## 573 92 73.97117  
## 574 95 86.87866  
## 575 67 82.38174  
## 576 79 82.41774  
## 577 75 70.81168  
## 578 64 94.27943  
## 579 90 96.56763  
## 580 74 96.17381  
## 581 71 83.56929  
## 582 87 91.26643  
## 583 71 93.49057  
## 584 77 95.36925  
## 585 80 80.85944  
## 586 75 84.39534  
## 587 95 82.97421  
## 588 93 77.45613  
## 589 69 82.72740  
## 590 77 77.65384  
## 591 63 67.80243  
## 592 64 66.62810  
## 593 111 74.11153  
## 594 62 75.35735  
## 595 62 91.41911  
## 596 78 89.37015  
## 597 102 82.88452  
## 598 93 82.28763  
## 599 94 84.44431  
## 600 64 77.11652  
## 601 77 89.87563  
## 602 73 87.62389  
## 603 79 93.76432  
## 604 80 87.78726  
## 605 70 93.25463  
## 606 86 97.35486  
## 607 87 92.79773  
## 608 108 98.47135  
## 609 91 94.72234  
## 610 86 93.80173  
## 611 81 96.01043  
## 612 76 82.86228  
## 613 85 76.13348  
## 614 66 77.17624  
## 615 69 77.23491  
## 616 92 84.72118  
## 617 94 84.54481  
## 618 99 91.45792  
## 619 93 81.48643  
## 620 87 76.05063  
## 621 82 75.66432  
## 622 72 77.26623  
## 623 97 77.11791  
## 624 93 85.23634  
## 625 79 94.58567  
## 626 79 82.27350  
## 627 81 76.21672  
## 628 62 80.88749  
## 629 76 77.09902  
## 630 80 79.78842  
## 631 61 80.55472  
## 632 57 78.32830  
## 633 113 74.66971  
## 634 87 70.44247  
## 635 21 75.76856  
## 636 77 81.25690  
## 637 72 85.21570  
## 638 76 87.45149  
## 639 95 83.24715  
## 640 70 85.12337  
## 641 82 90.08131  
## 642 88 85.69963  
## 643 85 91.72563  
## 644 95 95.57921  
## 645 79 80.82969  
## 646 72 76.76201  
## 647 82 84.40290  
## 648 103 88.69413  
## 649 79 83.20373  
## 650 58 92.68599  
## 651 74 82.30209  
## 652 87 79.07586  
## 653 98 86.73771  
## 654 88 81.29509  
## 655 79 66.71945  
## 656 75 73.26650  
## 657 79 86.38450  
## 658 95 80.74925  
## 659 64 72.12318  
## 660 76 86.47275  
## 661 54 79.04706  
## 662 64 82.85552  
## 663 79 74.31326  
## 664 65 75.84208  
## 665 72 84.08627  
## 666 81 79.05523  
## 667 80 79.31311  
## 668 80 80.23662  
## 669 83 75.09644  
## 670 65 61.44278  
## 671 82 60.04693  
## 672 102 73.69484  
## 673 89 75.61348  
## 674 65 75.09419  
## 675 80 60.58337  
## 676 91 79.31909  
## 677 92 85.07393  
## 678 72 71.05731  
## 679 83 82.34659  
## 680 94 75.70126  
## 681 75 76.36214  
## 682 75 79.97192  
## 683 82 79.96235  
## 684 84 77.60617  
## 685 70 75.63627  
## 686 66 72.69408  
## 687 99 78.08013  
## 688 88 77.26828  
## 689 94 81.21731  
## 690 86 75.48051  
## 691 94 71.40040  
## 692 100 74.64094  
## 693 85 78.67222  
## 694 88 88.47661  
## 695 89 82.65871  
## 696 96 89.85068  
## 697 88 97.41705  
## 698 95 84.06305  
## 699 73 83.32637  
## 700 95 79.06699  
## 701 78 81.32051  
## 702 90 81.98676  
## 703 88 72.57010  
## 704 77 73.44789  
## 705 86 74.81811  
## 706 64 74.98640  
## 707 69 84.21973  
## 708 96 84.56636  
## 709 71 82.75465  
## 710 78 77.97886  
## 711 56 84.76507  
## 712 67 79.18286  
## 713 81 82.33910  
## 714 75 78.44898  
## 715 45 81.50181  
## 716 73 84.18849  
## 717 89 80.75256  
## 718 70 87.77617  
## 719 69 79.01648  
## 720 67 84.53152  
## 721 67 73.76380  
## 722 67 74.64648  
## 723 66 97.05403  
## 724 77 95.97254  
## 725 75 77.70426  
## 726 81 91.18758  
## 727 85 75.73231  
## 728 77 73.62204  
## 729 81 65.45411  
## 730 80 78.44786  
## 731 69 84.83279  
## 732 95 85.78446  
## 733 50 82.58456  
## 734 66 80.45664  
## 735 86 74.78727  
## 736 65 70.95567  
## 737 90 73.82233  
## 738 98 84.23530  
## 739 91 79.76443  
## 740 71 75.55942  
## 741 61 83.90318  
## 742 84 78.55669  
## 743 93 80.46496  
## 744 99 76.10448  
## 745 95 77.89772  
## 746 116 71.32829  
## 747 106 73.19595  
## 748 93 84.23038  
## 749 99 80.71645  
## 750 93 82.88104  
## 751 108 86.58766  
## 752 103 77.59884  
## 753 106 76.13671  
## 754 72 87.43058  
## 755 99 86.48412  
## 756 100 90.45556  
## 757 99 86.15989  
## 758 103 97.07526  
## 759 79 95.61839  
## 760 91 91.30071  
## 761 89 102.70882  
## 762 86 94.41229  
## 763 74 97.02811  
## 764 92 89.96791  
## 765 96 82.18003  
## 766 96 83.80303  
## 767 101 85.00570  
## 768 87 80.53210  
## 769 87 87.92249  
## 770 101 94.71954  
## 771 110 73.29979  
## 772 108 71.60731  
## 773 66 67.21755  
## 774 56 61.79597  
## 775 74 75.69141  
## 776 104 84.94305  
## 777 63 76.45704  
## 778 56 76.87543  
## 779 56 79.08576  
## 780 52 82.82991  
## 781 74 77.53815  
## 782 66 88.68762  
## 783 59 86.93289  
## 784 62 86.40456  
## 785 82 85.95538  
## 786 54 71.33692  
## 787 77 81.90683  
## 788 76 74.62795  
## 789 78 72.43715  
## 790 74 68.46631  
## 791 102 80.61536  
## 792 69 77.72868  
## 793 91 78.76628  
## 794 100 79.50862  
## 795 87 83.21869  
## 796 83 78.19842  
## 797 84 89.64355  
## 798 78 89.13728  
## 799 93 82.46795  
## 800 46 77.71627  
## 801 82 68.72122  
## 802 68 73.81261  
## 803 45 90.70552  
## 804 92 88.53050  
## 805 49 93.36524  
## 806 82 81.87482  
## 807 76 69.53519  
## 808 63 65.83305  
## 809 61 80.91972  
## 810 80 75.66819  
## 811 70 69.78134  
## 812 77 74.65439  
## 813 65 86.20933  
## 814 105 90.52296  
## 815 92 89.88804  
## 816 103 91.22499  
## 817 91 85.30952  
## 818 85 76.50755  
## 819 73 77.81370  
## 820 89 76.53768  
## 821 99 80.85762  
## 822 90 83.55077  
## 823 91 86.28180  
## 824 88 90.36860  
## 825 72 76.09565  
## 826 82 75.87291  
## 827 85 87.02377  
## 828 73 78.48445  
## 829 95 76.31062  
## 830 75 73.67242  
## 831 53 82.00670  
## 832 69 76.92192  
## 833 94 77.12030  
## 834 74 75.08059  
## 835 80 87.08930  
## 836 88 84.10366  
## 837 92 69.87058  
## 838 98 67.95321  
## 839 57 76.27043  
## 840 95 77.82358  
## 841 98 61.48364  
## 842 75 73.93711  
## 843 73 80.09102  
## 844 72 80.39881  
## 845 67 77.94942  
## 846 63 79.27261  
## 847 60 80.39794  
## 848 81 79.57612  
## 849 92 88.63689  
## 850 89 85.94038  
## 851 91 83.89021  
## 852 74 84.32137  
## 853 66 88.01176  
## 854 88 89.30777  
## 855 57 84.21572  
## 856 67 89.72324  
## 857 73 88.68351  
## 858 83 93.65513  
## 859 88 86.63118  
## 860 76 80.90752  
## 861 116 83.25118  
## 862 78 82.34711  
## 863 103 83.24225  
## 864 88 77.00634  
## 865 79 79.45941  
## 866 92 70.73381  
## 867 97 76.64514  
## 868 58 80.04985  
## 869 70 77.07574  
## 870 83 81.67504  
## 871 73 74.98745  
## 872 89 84.02452  
## 873 90 72.21179  
## 874 80 77.78425  
## 875 75 81.11352  
## 876 86 82.50825  
## 877 90 76.61177  
## 878 92 74.25842  
## 879 42 69.83767  
## 880 68 78.11892  
## 881 61 75.86576  
## 882 58 70.01321  
## 883 82 82.94723  
## 884 82 80.03326  
## 885 97 82.57406  
## 886 106 87.68125  
## 887 76 74.67282  
## 888 87 67.15518  
## 889 101 76.66567  
## 890 97 77.28677  
## 891 103 73.35303  
## 892 112 85.23277  
## 893 89 82.52631  
## 894 83 80.73811  
## 895 93 74.93185  
## 896 80 84.73855  
## 897 83 75.75555  
## 898 97 81.92132  
## 899 69 87.57509  
## 900 72 81.64124  
## 901 86 78.23981  
## 902 57 79.85345  
## 903 83 83.07927  
## 904 88 77.58223  
## 905 72 83.27506  
## 906 86 84.80112  
## 907 96 77.35284  
## 908 87 83.92642  
## 909 89 80.88043  
## 910 86 79.19680  
## 911 80 61.31942  
## 912 71 58.38833  
## 913 76 66.35593  
## 914 86 60.64071  
## 915 82 59.60618  
## 916 65 71.52196  
## 917 67 82.97232  
## 918 68 69.80963  
## 919 82 73.80638  
## 920 80 71.90841  
## 921 85 76.54061  
## 922 78 81.68224  
## 923 87 85.81771  
## 924 81 92.23216  
## 925 91 84.05760  
## 926 87 79.86707  
## 927 85 83.63096  
## 928 70 72.09129  
## 929 82 76.10990  
## 930 75 76.51728  
## 931 92 79.08606  
## 932 92 75.26965  
## 933 98 92.94321  
## 934 51 85.87283  
## 935 111 73.31166  
## 936 86 77.15060  
## 937 55 70.75125  
## 938 74 83.97144  
## 939 66 95.72627  
## 940 70 77.69374  
## 941 63 79.19794  
## 942 81 76.35351  
## 943 87 78.32832  
## 944 90 75.91017  
## 945 80 70.76003  
## 946 87 86.96236  
## 947 84 78.23717  
## 948 88 81.18502  
## 949 86 75.78016  
## 950 81 75.92134  
## 951 76 77.63943  
## 952 70 97.27400  
## 953 67 85.27491  
## 954 82 88.53154  
## 955 65 93.95050  
## 956 95 104.40963  
## 957 90 93.08013  
## 958 83 94.25311  
## 959 88 96.39318  
## 960 66 102.13149  
## 961 56 93.15151  
## 962 66 93.26443  
## 963 45 88.38960  
## 964 69 83.12927  
## 965 61 87.16569  
## 966 97 86.20118  
## 967 84 84.48924  
## 968 97 93.72888  
## 969 54 100.22576  
## 970 76 90.75092  
## 971 85 85.92259  
## 972 74 90.08918  
## 973 104 88.66595  
## 974 94 94.18213  
## 975 107 85.07843  
## 976 100 87.16259  
## 977 64 76.98871  
## 978 94 66.83786  
## 979 45 71.05327  
## 980 68 80.22158  
## 981 78 77.96724  
## 982 87 74.09699  
## 983 80 75.16087  
## 984 92 75.85082  
## 985 85 78.57326  
## 986 84 86.20376  
## 987 98 90.85388  
## 988 84 82.07620  
## 989 88 92.37084  
## 990 97 84.76236  
## 991 78 86.21000  
## 992 85 82.30628  
## 993 95 93.02867  
## 994 75 92.67147  
## 995 62 87.26267  
## 996 114 85.61029  
## 997 110 81.43227  
## 998 98 78.74050  
## 999 86 85.55504  
## 1000 88 85.67697  
## 1001 87 89.49447  
## 1002 64 89.33847  
## 1003 68 97.19323  
## 1004 65 94.41711  
## 1005 89 90.90383  
## 1006 75 89.17507  
## 1007 81 90.21799  
## 1008 79 93.50255  
## 1009 80 93.87559  
## 1010 71 95.81350  
## 1011 76 90.93701  
## 1012 77 92.44312  
## 1013 82 78.34708  
## 1014 65 72.71619  
## 1015 79 76.46792  
## 1016 93 74.59291  
## 1017 83 72.01169  
## 1018 83 63.68687  
## 1019 63 80.15956  
## 1020 53 87.11331  
## 1021 56 92.71264  
## 1022 79 72.71623  
## 1023 90 73.56595  
## 1024 71 69.61434  
## 1025 77 73.36825  
## 1026 78 79.86398  
## 1027 54 79.38112  
## 1028 94 76.32710  
## 1029 99 87.36658  
## 1030 86 87.46441  
## 1031 99 94.51520  
## 1032 92 93.38030  
## 1033 95 91.96627  
## 1034 67 89.23678  
## 1035 79 82.81230  
## 1036 75 80.88571  
## 1037 64 87.73838  
## 1038 90 86.40998  
## 1039 74 81.97909  
## 1040 71 82.52483  
## 1041 87 68.20531  
## 1042 71 66.22217  
## 1043 77 70.54708  
## 1044 80 67.31787  
## 1045 75 70.66388  
## 1046 95 84.08191  
## 1047 93 85.42203  
## 1048 69 72.46110  
## 1049 77 76.83792  
## 1050 63 75.06820  
## 1051 64 75.85712  
## 1052 111 76.59533  
## 1053 62 70.17289  
## 1054 62 68.75619  
## 1055 78 73.16580  
## 1056 102 82.69221  
## 1057 93 78.45319  
## 1058 94 74.46707  
## 1059 64 84.53133  
## 1060 77 72.54941  
## 1061 73 85.11678  
## 1062 79 83.35557  
## 1063 80 82.59932  
## 1064 70 83.53339  
## 1065 86 86.85838  
## 1066 87 84.46278  
## 1067 108 88.67685  
## 1068 91 83.37239  
## 1069 86 88.58651  
## 1070 81 89.59870  
## 1071 76 84.64656  
## 1072 85 81.64766  
## 1073 66 92.61935  
## 1074 69 86.27243  
## 1075 92 82.85143  
## 1076 94 85.66203  
## 1077 99 95.84800  
## 1078 93 90.95834  
## 1079 87 95.81290  
## 1080 82 98.69009  
## 1081 72 85.71811  
## 1082 97 69.76796  
## 1083 93 75.60889  
## 1084 79 71.34770  
## 1085 79 72.41132  
## 1086 81 72.15901  
## 1087 62 71.27513  
## 1088 76 84.49067  
## 1089 80 82.25327  
## 1090 61 81.79879  
## 1091 57 85.53387  
## 1092 113 80.98372  
## 1093 87 73.90174  
## 1094 21 73.92732  
## 1095 77 71.53401  
## 1096 72 74.23172  
## 1097 76 81.61689  
## 1098 95 82.79164  
## 1099 70 77.45100  
## 1100 82 76.58674  
## 1101 88 78.73276  
## 1102 85 96.77330  
## 1103 95 87.60659  
## 1104 79 76.93790  
## 1105 72 87.38424  
## 1106 82 72.88084  
## 1107 103 80.21365  
## 1108 79 79.76677  
## 1109 58 66.28857  
## 1110 74 69.89638  
## 1111 87 67.14770  
## 1112 98 70.31318  
## 1113 88 63.48091  
## 1114 79 71.00934  
## 1115 75 70.69567  
## 1116 79 70.69689  
## 1117 95 73.50918  
## 1118 64 78.85426  
## 1119 76 83.60694  
## 1120 54 83.10800  
## 1121 64 80.02899  
## 1122 79 71.49187  
## 1123 65 66.74572  
## 1124 72 73.57565  
## 1125 81 80.18532  
## 1126 80 72.35786  
## 1127 80 72.86273  
## 1128 83 71.97395  
## 1129 65 65.32023  
## 1130 82 73.37091  
## 1131 102 70.95516  
## 1132 89 67.70542  
## 1133 65 71.73850  
## 1134 80 76.98511  
## 1135 91 84.88781  
## 1136 92 94.03477  
## 1137 72 83.22815  
## 1138 83 84.15055  
## 1139 94 89.26872  
## 1140 75 79.14093  
## 1141 75 84.17269  
## 1142 82 84.81615  
## 1143 84 72.06287  
## 1144 70 77.75736  
## 1145 66 78.77255  
## 1146 99 72.63020  
## 1147 88 78.03885  
## 1148 94 90.17251  
## 1149 86 78.48975  
## 1150 94 79.38583  
## 1151 100 77.11963  
## 1152 85 90.17322  
## 1153 88 81.64387  
## 1154 89 83.99443  
## 1155 96 85.64188  
## 1156 88 82.44665  
## 1157 95 95.51429  
## 1158 73 77.02829  
## 1159 95 89.46990  
## 1160 78 94.55289  
## 1161 90 95.36864  
## 1162 88 91.57210  
## 1163 77 80.87037  
## 1164 86 82.79777  
## 1165 64 75.87495  
## 1166 69 80.97836  
## 1167 96 84.59811  
## 1168 71 81.97436  
## 1169 78 94.32579  
## 1170 56 85.66276  
## 1171 67 74.16725  
## 1172 81 77.35892  
## 1173 75 71.56327  
## 1174 45 89.51002  
## 1175 73 83.62805  
## 1176 89 96.01226  
## 1177 70 99.63880  
## 1178 69 86.66543  
## 1179 67 85.08412  
## 1180 67 88.86202  
## 1181 67 96.16164  
## 1182 66 94.35191  
## 1183 77 77.78130  
## 1184 75 79.97524  
## 1185 81 76.64023  
## 1186 85 81.35226  
## 1187 77 82.58319  
## 1188 81 83.45274  
## 1189 80 80.12935  
## 1190 69 82.74097  
## 1191 95 78.38666  
## 1192 50 85.76137  
## 1193 66 80.03589  
## 1194 86 75.26520  
## 1195 65 79.94641  
## 1196 90 86.44159  
## 1197 98 84.42915  
## 1198 91 76.77597  
## 1199 71 85.49227  
## 1200 61 85.43649  
## 1201 84 82.01907  
## 1202 93 85.49127  
## 1203 99 76.48413  
## 1204 95 72.47560  
## 1205 116 74.71764  
## 1206 106 79.17754  
## 1207 93 75.89540  
## 1208 99 72.52410  
## 1209 93 72.99217  
## 1210 108 73.95031  
## 1211 103 80.28342  
## 1212 106 76.84542  
## 1213 72 72.66814  
## 1214 99 79.43934  
## 1215 100 84.21485  
## 1216 99 76.13103  
## 1217 103 75.85239  
## 1218 79 80.34523  
## 1219 91 81.40989  
## 1220 89 89.32086  
## 1221 86 71.65314  
## 1222 74 83.47910  
## 1223 92 79.31333  
## 1224 96 83.99140  
## 1225 96 86.04019  
## 1226 101 80.58368  
## 1227 87 78.55305  
## 1228 87 81.19871  
## 1229 101 80.75316  
## 1230 110 83.34638  
## 1231 108 72.71053  
## 1232 66 83.05962  
## 1233 56 75.30919  
## 1234 74 61.84184  
## 1235 104 75.49493  
## 1236 63 66.04304  
## 1237 56 67.75688  
## 1238 56 69.12003  
## 1239 52 71.72482  
## 1240 74 69.96031  
## 1241 66 70.90493  
## 1242 59 82.19781  
## 1243 62 78.83265  
## 1244 82 79.97341  
## 1245 54 73.89507  
## 1246 77 74.58934  
## 1247 76 77.85825  
## 1248 78 80.11585  
## 1249 74 72.59577  
## 1250 102 73.24053  
## 1251 69 74.94112  
## 1252 91 80.93204  
## 1253 100 82.70670  
## 1254 87 83.81551  
## 1255 83 81.01255  
## 1256 84 83.63764  
## 1257 78 83.91016  
## 1258 93 75.41719  
## 1259 46 77.28019  
## 1260 82 85.05006  
## 1261 68 76.05888  
## 1262 45 78.48064  
## 1263 92 86.08551  
## 1264 49 75.94236  
## 1265 82 79.40133  
## 1266 76 84.21428  
## 1267 63 76.93622  
## 1268 61 76.13965  
## 1269 80 80.08179  
## 1270 70 80.51164  
## 1271 77 78.57486  
## 1272 65 82.35286  
## 1273 105 91.28096  
## 1274 92 96.60549  
## 1275 103 94.87582  
## 1276 91 92.31089  
## 1277 85 94.05118  
## 1278 73 94.20321  
## 1279 89 96.22611  
## 1280 99 87.91910  
## 1281 90 83.99419  
## 1282 91 78.77420  
## 1283 88 75.09218  
## 1284 72 82.55089  
## 1285 82 71.98056  
## 1286 85 69.20131  
## 1287 73 77.96775  
## 1288 95 87.78023  
## 1289 75 86.35932  
## 1290 53 82.21422  
## 1291 69 79.65022  
## 1292 94 75.35074  
## 1293 74 81.20123  
## 1294 80 90.70037  
## 1295 88 84.24705  
## 1296 92 78.67993  
## 1297 98 87.97745  
## 1298 57 87.69785  
## 1299 95 92.90616  
## 1300 98 96.57243  
## 1301 75 85.69885  
## 1302 73 78.76782  
## 1303 72 73.07410  
## 1304 67 77.54960  
## 1305 63 83.26963  
## 1306 60 78.98654  
## 1307 81 79.07793  
## 1308 92 80.37722  
## 1309 89 79.71387  
## 1310 91 78.50491  
## 1311 74 76.75080  
## 1312 66 80.27969  
## 1313 88 73.46866  
## 1314 57 79.25616  
## 1315 67 82.00955  
## 1316 73 79.89839  
## 1317 83 91.02800  
## 1318 88 88.08337  
## 1319 76 88.85369  
## 1320 116 94.45356  
## 1321 78 84.35462  
## 1322 103 78.05740  
## 1323 88 83.95127  
## 1324 79 86.33623  
## 1325 92 82.90238  
## 1326 97 81.67246  
## 1327 58 85.51611  
## 1328 70 80.18255  
## 1329 83 79.54560  
## 1330 73 75.54402  
## 1331 89 73.65885  
## 1332 90 71.01422  
## 1333 80 81.45969  
## 1334 75 82.86082  
## 1335 86 80.62995  
## 1336 90 88.25696  
## 1337 92 76.46766  
## 1338 42 80.34835  
## 1339 68 72.91620  
## 1340 61 76.90302  
## 1341 58 83.18826  
## 1342 82 70.94189  
## 1343 82 76.87526  
## 1344 97 80.45797  
## 1345 106 69.39293  
## 1346 76 84.77448  
## 1347 87 85.62956  
## 1348 101 83.98628  
## 1349 97 80.12358  
## 1350 103 86.59676  
## 1351 112 87.14948  
## 1352 89 89.34894  
## 1353 83 96.16870  
## 1354 93 89.87599  
## 1355 80 82.54057  
## 1356 83 72.35155  
## 1357 97 75.48538  
## 1358 69 89.39629  
## 1359 72 90.79489  
## 1360 86 72.17537  
## 1361 57 98.82305  
## 1362 83 91.07166  
## 1363 88 74.70489  
## 1364 72 74.08408  
## 1365 86 75.45591  
## 1366 96 75.76548  
## 1367 87 67.74708  
## 1368 89 63.73702  
## 1369 86 63.09795  
## 1370 80 79.36250  
## 1371 71 85.89266  
## 1372 76 88.93737  
## 1373 86 86.42517  
## 1374 82 79.24316  
## 1375 65 80.67956  
## 1376 67 74.32748  
## 1377 68 77.51726  
## 1378 82 74.68399  
## 1379 80 90.58364  
## 1380 85 84.36898  
## 1381 78 89.61974  
## 1382 87 87.27046  
## 1383 81 91.72583  
## 1384 91 90.46183  
## 1385 87 92.84914  
## 1386 85 83.81388  
## 1387 70 78.44059  
## 1388 82 78.52910  
## 1389 75 84.60661  
## 1390 92 80.35381  
## 1391 92 81.01220  
## 1392 98 81.58834  
## 1393 51 83.27561  
## 1394 111 83.42052  
## 1395 86 88.09058  
## 1396 55 80.71739  
## 1397 74 82.57078  
## 1398 66 79.27823  
## 1399 70 79.45690  
## 1400 63 87.45363  
## 1401 81 82.67362  
## 1402 87 83.26842  
## 1403 90 79.96190  
## 1404 80 83.17930  
## 1405 87 81.28235  
## 1406 84 88.91157  
## 1407 88 79.89535  
## 1408 86 82.59148  
## 1409 81 83.88969  
## 1410 76 79.11848  
## 1411 70 80.02279  
## 1412 67 79.38053  
## 1413 82 79.62617  
## 1414 65 81.86965  
## 1415 95 81.52487  
## 1416 90 75.59362  
## 1417 83 76.78916  
## 1418 88 74.30279  
## 1419 66 76.30224  
## 1420 56 69.65634  
## 1421 66 74.26352  
## 1422 45 83.82491  
## 1423 69 84.75257  
## 1424 61 73.73032  
## 1425 97 75.06767  
## 1426 84 84.63083  
## 1427 97 78.97639  
## 1428 54 76.45999  
## 1429 76 71.29329  
## 1430 85 81.13600  
## 1431 74 80.70558  
## 1432 104 78.14821  
## 1433 94 71.04719  
## 1434 107 80.36170  
## 1435 100 81.90376  
## 1436 64 93.33645  
## 1437 94 88.19413  
## 1438 45 90.25974  
## 1439 68 84.66770  
## 1440 78 82.96837  
## 1441 87 89.12204  
## 1442 80 87.95702  
## 1443 92 82.57955  
## 1444 85 82.70475  
## 1445 84 79.25230  
## 1446 98 80.20210  
## 1447 84 71.72168  
## 1448 88 74.42225  
## 1449 97 79.63366  
## 1450 78 81.94346  
## 1451 85 82.75808  
## 1452 95 80.78277  
## 1453 75 76.62972  
## 1454 62 77.85689  
## 1455 114 68.36665  
## 1456 110 69.11303  
## 1457 98 66.24626  
## 1458 86 81.84655  
## 1459 88 78.35787  
## 1460 87 69.94233  
## 1461 64 75.76441  
## 1462 68 81.02432  
## 1463 65 74.52049  
## 1464 89 81.01784  
## 1465 75 71.46302  
## 1466 81 75.61764  
## 1467 79 78.34152  
## 1468 80 77.83881  
## 1469 71 77.22750  
## 1470 76 86.79195  
## 1471 77 77.71251  
## 1472 82 82.62507  
## 1473 65 81.35461  
## 1474 79 89.93796  
## 1475 93 96.51608  
## 1476 83 86.16285  
## 1477 83 91.33961  
## 1478 63 86.93554  
## 1479 53 84.63384  
## 1480 56 72.70103  
## 1481 79 71.08248  
## 1482 90 91.35231  
## 1483 71 86.00672  
## 1484 77 91.94352  
## 1485 78 85.43193  
## 1486 54 83.53370  
## 1487 94 85.50160  
## 1488 99 80.42214  
## 1489 86 87.39467  
## 1490 99 93.21013  
## 1491 92 83.54572  
## 1492 95 77.31713  
## 1493 67 89.01612  
## 1494 79 89.92811  
## 1495 75 76.09057  
## 1496 64 77.41702  
## 1497 90 85.67225  
## 1498 74 87.78315  
## 1499 71 71.42951  
## 1500 87 78.50359  
## 1501 71 70.81870  
## 1502 77 80.91072  
## 1503 80 75.32227  
## 1504 75 68.16678  
## 1505 95 79.64992  
## 1506 93 71.17387  
## 1507 69 79.90700  
## 1508 77 84.00078  
## 1509 63 82.16401  
## 1510 64 77.16445  
## 1511 111 76.00898  
## 1512 62 78.13104  
## 1513 62 72.61470  
## 1514 78 77.78654  
## 1515 102 77.61568  
## 1516 93 72.10171  
## 1517 94 81.84125  
## 1518 64 77.74045  
## 1519 77 78.32464  
## 1520 73 67.84336  
## 1521 79 80.92785

mean (apply(predicted\_values, 1, min)/apply(predicted\_values, 1, max)) # calculate accuracy

## [1] 0.860924

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 82 67.44342  
## 2 80 67.45113  
## 3 85 70.07451  
## 4 78 75.43397  
## 5 87 67.72684  
## 6 81 67.72587  
## 7 91 63.76240  
## 8 87 77.57821  
## 9 85 87.50168  
## 10 70 75.33778  
## 11 82 86.57140  
## 12 75 76.10532  
## 13 92 79.30210  
## 14 92 85.05514  
## 15 98 89.33395  
## 16 51 86.35917  
## 17 111 75.87327  
## 18 86 75.27661  
## 19 55 76.92580  
## 20 74 72.38257  
## 21 66 91.24888  
## 22 70 81.21417  
## 23 63 76.85482  
## 24 81 75.73465  
## 25 87 92.95906  
## 26 90 79.74796  
## 27 80 82.59460  
## 28 87 82.93132  
## 29 84 84.01025  
## 30 88 83.67314  
## 31 86 74.98318  
## 32 81 94.40116  
## 33 76 85.35736  
## 34 70 86.31272  
## 35 67 85.20792  
## 36 82 79.86968  
## 37 65 71.39335  
## 38 95 79.65920  
## 39 90 88.48478  
## 40 83 92.89590  
## 41 88 95.70531  
## 42 66 80.15908  
## 43 56 77.64883  
## 44 66 67.18263  
## 45 45 64.94270  
## 46 69 62.58915  
## 47 61 71.88851  
## 48 97 67.42428  
## 49 84 66.89260  
## 50 97 74.11101  
## 51 54 84.19595  
## 52 76 77.19389  
## 53 85 81.04087  
## 54 74 78.77630  
## 55 104 78.14143  
## 56 94 76.04199  
## 57 107 79.46042  
## 58 100 73.69400  
## 59 64 74.54247  
## 60 94 75.74929  
## 61 45 74.97222  
## 62 68 75.44410  
## 63 78 79.02010  
## 64 87 69.75831  
## 65 80 70.64981  
## 66 92 69.12150  
## 67 85 74.71038  
## 68 84 69.62466  
## 69 98 70.17964  
## 70 84 70.25523  
## 71 88 69.23863  
## 72 97 71.09084  
## 73 78 72.67250  
## 74 85 67.57287  
## 75 95 73.96325  
## 76 75 82.29090  
## 77 62 76.84995  
## 78 114 80.96384  
## 79 110 86.00466  
## 80 98 80.99199  
## 81 86 90.30120  
## 82 88 86.68144  
## 83 87 76.35323  
## 84 64 83.38313  
## 85 68 86.70403  
## 86 65 88.15920  
## 87 89 77.29176  
## 88 75 78.74435  
## 89 81 78.89828  
## 90 79 84.86922  
## 91 80 76.52274  
## 92 71 68.53117  
## 93 76 80.63921  
## 94 77 79.04574  
## 95 82 91.06636  
## 96 65 78.71918  
## 97 79 73.94550  
## 98 93 72.47404  
## 99 83 76.97092  
## 100 83 78.30645  
## 101 63 77.84933  
## 102 53 73.92133  
## 103 56 87.30513  
## 104 79 76.38672  
## 105 90 87.67499  
## 106 71 77.46469  
## 107 77 82.05258  
## 108 78 85.24247  
## 109 54 87.16002  
## 110 94 83.45556  
## 111 99 83.35626  
## 112 86 86.73727  
## 113 99 91.21914  
## 114 92 88.04261  
## 115 95 78.51105  
## 116 67 81.95540  
## 117 79 82.76242  
## 118 75 81.70924  
## 119 64 85.27418  
## 120 90 69.20964  
## 121 74 69.37010  
## 122 71 78.11881  
## 123 87 69.72151  
## 124 71 66.82142  
## 125 77 65.28600  
## 126 80 71.84567  
## 127 75 71.77897  
## 128 95 76.57889  
## 129 93 79.01450  
## 130 69 85.26364  
## 131 77 75.40463  
## 132 63 82.54799  
## 133 64 78.50081  
## 134 111 82.48798  
## 135 62 84.16834  
## 136 62 81.26085  
## 137 78 80.88645  
## 138 102 75.53196  
## 139 93 80.44320  
## 140 94 84.92377  
## 141 64 75.02694  
## 142 77 75.21066  
## 143 73 71.99137  
## 144 79 71.21597  
## 145 80 81.47611  
## 146 70 80.90290  
## 147 86 80.66912  
## 148 87 82.13911  
## 149 108 74.68290  
## 150 91 73.79684  
## 151 86 69.14518  
## 152 81 69.07586  
## 153 76 74.64032  
## 154 85 75.73128  
## 155 66 71.89865  
## 156 69 78.09027  
## 157 92 76.71291  
## 158 94 77.55373  
## 159 99 77.24322  
## 160 93 75.43189  
## 161 87 81.07324  
## 162 82 75.80984  
## 163 72 75.01335  
## 164 97 87.32782  
## 165 93 87.38260  
## 166 79 85.36363  
## 167 79 72.33313  
## 168 81 88.77234  
## 169 62 80.72687  
## 170 76 81.51611  
## 171 80 79.82869  
## 172 61 79.48619  
## 173 57 79.26894  
## 174 113 88.37324  
## 175 87 86.43100  
## 176 21 85.22346  
## 177 77 83.65356  
## 178 72 80.09002  
## 179 76 88.30086  
## 180 95 79.69794  
## 181 70 84.31400  
## 182 82 74.81675  
## 183 88 80.47708  
## 184 85 79.70448  
## 185 95 78.06843  
## 186 79 85.14572  
## 187 72 82.05263  
## 188 82 94.71854  
## 189 103 96.65872  
## 190 79 87.64550  
## 191 58 90.21192  
## 192 74 87.23107  
## 193 87 78.75851  
## 194 98 79.00789  
## 195 88 81.25164  
## 196 79 81.88150  
## 197 75 84.03268  
## 198 79 82.40478  
## 199 95 95.78563  
## 200 64 80.30579  
## 201 76 75.01089  
## 202 54 95.34271  
## 203 64 87.50287  
## 204 79 87.19443  
## 205 65 76.17791  
## 206 72 76.06067  
## 207 81 64.91472  
## 208 80 65.72480  
## 209 80 72.39191  
## 210 83 77.65363  
## 211 65 84.15721  
## 212 82 82.04471  
## 213 102 86.08161  
## 214 89 87.63339  
## 215 65 75.05263  
## 216 80 74.98438  
## 217 91 83.25936  
## 218 92 82.71286  
## 219 72 75.53272  
## 220 83 76.88389  
## 221 94 71.86619  
## 222 75 76.06333  
## 223 75 74.51141  
## 224 82 70.23775  
## 225 84 73.52272  
## 226 70 79.16581  
## 227 66 86.89115  
## 228 99 87.13819  
## 229 88 85.46662  
## 230 94 89.75357  
## 231 86 91.54361  
## 232 94 90.59831  
## 233 100 95.02468  
## 234 85 75.40648  
## 235 88 79.71284  
## 236 89 75.96297  
## 237 96 85.91518  
## 238 88 78.85678  
## 239 95 76.84806  
## 240 73 86.14862  
## 241 95 78.05120  
## 242 78 77.47512  
## 243 90 78.23876  
## 244 88 73.50749  
## 245 77 86.78095  
## 246 86 79.08171  
## 247 64 84.72774  
## 248 69 78.34657  
## 249 96 82.76116  
## 250 71 86.34586  
## 251 78 87.92397  
## 252 56 84.11055  
## 253 67 77.99811  
## 254 81 85.45186  
## 255 75 77.80467  
## 256 45 75.25132  
## 257 73 76.95882  
## 258 89 68.38314  
## 259 70 73.50657  
## 260 69 94.02116  
## 261 67 93.71058  
## 262 67 89.96523  
## 263 67 80.98936  
## 264 66 84.84977  
## 265 77 85.19842  
## 266 75 79.47680  
## 267 81 79.10396  
## 268 85 95.64288  
## 269 77 86.71087  
## 270 81 85.77351  
## 271 80 88.98923  
## 272 69 80.79937  
## 273 95 77.98964  
## 274 50 75.28088  
## 275 66 77.35333  
## 276 86 76.26755  
## 277 65 75.33738  
## 278 90 88.17904  
## 279 98 96.27179  
## 280 91 89.92458  
## 281 71 92.67221  
## 282 61 84.05349  
## 283 84 74.83770  
## 284 93 85.95691  
## 285 99 78.72426  
## 286 95 78.71273  
## 287 116 81.26346  
## 288 106 85.39316  
## 289 93 94.85380  
## 290 99 98.58835  
## 291 93 82.07467  
## 292 108 75.18221  
## 293 103 76.37619  
## 294 106 82.90378  
## 295 72 88.14256  
## 296 99 81.52590  
## 297 100 82.87604  
## 298 99 78.59107  
## 299 103 75.43887  
## 300 79 75.87870  
## 301 91 77.95901  
## 302 89 77.42188  
## 303 86 79.91021  
## 304 74 75.70473  
## 305 92 72.30603  
## 306 96 74.72546  
## 307 96 77.04007  
## 308 101 84.57445  
## 309 87 79.03250  
## 310 87 75.61020  
## 311 101 85.12809  
## 312 110 83.01153  
## 313 108 81.21832  
## 314 66 77.58116  
## 315 56 76.16731  
## 316 74 71.52026  
## 317 104 75.22951  
## 318 63 75.48431  
## 319 56 76.87937  
## 320 56 75.93558  
## 321 52 75.95209  
## 322 74 82.46585  
## 323 66 74.48463  
## 324 59 77.49127  
## 325 62 75.66515  
## 326 82 77.13311  
## 327 54 79.35974  
## 328 77 78.58116  
## 329 76 85.79704  
## 330 78 87.45633  
## 331 74 82.37835  
## 332 102 86.72390  
## 333 69 75.70592  
## 334 91 80.05905  
## 335 100 80.56484  
## 336 87 81.56194  
## 337 83 79.82098  
## 338 84 87.10452  
## 339 78 79.65431  
## 340 93 76.07465  
## 341 46 83.48986  
## 342 82 80.22453  
## 343 68 82.98359  
## 344 45 82.75474  
## 345 92 75.27901  
## 346 49 79.49034  
## 347 82 82.43166  
## 348 76 82.29389  
## 349 63 84.23164  
## 350 61 76.59806  
## 351 80 74.96025  
## 352 70 76.25467  
## 353 77 68.62133  
## 354 65 73.26311  
## 355 105 75.04697  
## 356 92 81.57823  
## 357 103 76.97103  
## 358 91 84.95317  
## 359 85 88.65480  
## 360 73 81.29242  
## 361 89 80.11348  
## 362 99 81.05779  
## 363 90 78.21721  
## 364 91 78.12708  
## 365 88 77.36967  
## 366 72 77.29128  
## 367 82 81.79287  
## 368 85 87.54212  
## 369 73 85.17605  
## 370 95 81.07061  
## 371 75 85.04887  
## 372 53 78.68070  
## 373 69 77.38031  
## 374 94 70.92452  
## 375 74 74.03955  
## 376 80 71.75593  
## 377 88 75.89851  
## 378 92 76.24042  
## 379 98 78.59571  
## 380 57 82.86759  
## 381 95 86.95993  
## 382 98 84.03568  
## 383 75 87.55968  
## 384 73 85.98648  
## 385 72 82.14632  
## 386 67 85.69631  
## 387 63 87.07823  
## 388 60 86.90617  
## 389 81 81.14569  
## 390 92 74.98691  
## 391 89 74.22010  
## 392 91 66.23871  
## 393 74 70.80897  
## 394 66 79.02572  
## 395 88 74.46659  
## 396 57 76.17838  
## 397 67 78.85746  
## 398 73 75.16575  
## 399 83 90.13430  
## 400 88 76.43702  
## 401 76 80.42516  
## 402 116 71.76322  
## 403 78 85.62887  
## 404 103 88.54009  
## 405 88 85.41391  
## 406 79 80.87090  
## 407 92 76.33751  
## 408 97 85.13585  
## 409 58 76.30060  
## 410 70 72.82338  
## 411 83 85.53497  
## 412 73 88.16092  
## 413 89 94.67713  
## 414 90 92.65560  
## 415 80 90.99936  
## 416 75 85.43587  
## 417 86 92.80923  
## 418 90 89.13903  
## 419 92 87.66494  
## 420 42 86.26550  
## 421 68 87.30838  
## 422 61 95.64263  
## 423 58 89.77173  
## 424 82 96.58498  
## 425 82 89.08766  
## 426 97 84.87436  
## 427 106 79.09352  
## 428 76 83.13129  
## 429 87 74.78158  
## 430 101 75.01630  
## 431 97 75.37997  
## 432 103 91.42830  
## 433 112 89.53917  
## 434 89 80.57424  
## 435 83 77.44170  
## 436 93 80.56396  
## 437 80 86.13667  
## 438 83 86.17377  
## 439 97 76.58950  
## 440 69 76.80901  
## 441 72 83.22035  
## 442 86 78.71702  
## 443 57 68.39029  
## 444 83 74.51458  
## 445 88 65.97702  
## 446 72 69.94125  
## 447 86 78.07337  
## 448 96 79.56762  
## 449 87 75.47085  
## 450 89 78.95959  
## 451 86 82.04844  
## 452 80 78.93698  
## 453 71 75.55269  
## 454 76 73.11614  
## 455 86 71.16156  
## 456 82 68.87201  
## 457 65 74.10248  
## 458 67 80.17785  
## 459 68 77.30115  
## 460 82 72.49663  
## 461 80 77.63991  
## 462 85 72.55387  
## 463 78 75.26255  
## 464 87 89.56768  
## 465 81 89.34414  
## 466 91 81.88434  
## 467 87 84.87743  
## 468 85 81.65693  
## 469 70 82.50675  
## 470 82 81.01440  
## 471 75 75.36897  
## 472 92 74.83014  
## 473 92 86.95051  
## 474 98 75.64894  
## 475 51 71.91194  
## 476 111 88.80437  
## 477 86 72.45152  
## 478 55 85.64352  
## 479 74 87.18982  
## 480 66 88.23807  
## 481 70 97.61437  
## 482 63 94.15000  
## 483 81 85.46822  
## 484 87 81.29913  
## 485 90 83.24641  
## 486 80 73.97669  
## 487 87 78.98370  
## 488 84 87.86617  
## 489 88 90.77691  
## 490 86 82.77272  
## 491 81 81.84398  
## 492 76 78.91267  
## 493 70 78.56951  
## 494 67 90.81276  
## 495 82 82.15841  
## 496 65 77.31243  
## 497 95 91.95794  
## 498 90 87.79896  
## 499 83 76.99782  
## 500 88 79.89077  
## 501 66 74.67528  
## 502 56 86.66090  
## 503 66 76.67202  
## 504 45 84.36278  
## 505 69 82.37365  
## 506 61 77.49743  
## 507 97 72.37151  
## 508 84 75.22012  
## 509 97 73.18528  
## 510 54 71.40753  
## 511 76 78.79972  
## 512 85 77.39860  
## 513 74 81.06421  
## 514 104 78.92919  
## 515 94 83.29334  
## 516 107 92.13062  
## 517 100 86.88210  
## 518 64 86.09691  
## 519 94 83.65993  
## 520 45 80.00122  
## 521 68 75.79357  
## 522 78 71.29807  
## 523 87 80.14395  
## 524 80 85.04878  
## 525 92 86.42506  
## 526 85 77.88073  
## 527 84 89.99265  
## 528 98 93.12574  
## 529 84 86.35583  
## 530 88 82.66900  
## 531 97 86.55700  
## 532 78 76.00762  
## 533 85 75.10227  
## 534 95 76.24862  
## 535 75 93.95561  
## 536 62 86.80163  
## 537 114 94.03234  
## 538 110 86.58101  
## 539 98 86.56917  
## 540 86 88.22369  
## 541 88 85.39937  
## 542 87 81.25140  
## 543 64 82.13859  
## 544 68 80.30914  
## 545 65 74.44826  
## 546 89 76.10264  
## 547 75 74.08154  
## 548 81 69.92862  
## 549 79 69.95132  
## 550 80 74.53685  
## 551 71 77.21791  
## 552 76 68.76742  
## 553 77 67.21485  
## 554 82 78.58185  
## 555 65 74.37172  
## 556 79 77.15918  
## 557 93 83.53078  
## 558 83 84.49880  
## 559 83 86.41669  
## 560 63 75.22195  
## 561 53 73.30565  
## 562 56 76.43071  
## 563 79 67.01121  
## 564 90 78.15236  
## 565 71 84.22717  
## 566 77 103.62586  
## 567 78 99.78573  
## 568 54 92.94880  
## 569 94 96.14162  
## 570 99 93.83560  
## 571 86 87.21972  
## 572 99 80.56718  
## 573 92 75.47100  
## 574 95 85.48369  
## 575 67 81.64591  
## 576 79 80.88031  
## 577 75 72.00168  
## 578 64 98.77644  
## 579 90 95.99900  
## 580 74 95.74313  
## 581 71 81.34501  
## 582 87 90.01800  
## 583 71 90.45721  
## 584 77 93.97424  
## 585 80 79.58646  
## 586 75 83.30092  
## 587 95 80.62590  
## 588 93 74.64967  
## 589 69 81.50964  
## 590 77 77.75612  
## 591 63 68.67546  
## 592 64 67.09276  
## 593 111 74.64771  
## 594 62 74.22550  
## 595 62 91.69689  
## 596 78 88.52477  
## 597 102 83.85911  
## 598 93 85.75061  
## 599 94 84.90506  
## 600 64 78.19194  
## 601 77 86.69231  
## 602 73 84.65045  
## 603 79 90.56613  
## 604 80 88.21833  
## 605 70 89.20599  
## 606 86 96.33209  
## 607 87 89.50356  
## 608 108 96.00533  
## 609 91 94.62374  
## 610 86 93.39055  
## 611 81 95.00998  
## 612 76 80.83454  
## 613 85 75.38982  
## 614 66 76.75518  
## 615 69 78.99632  
## 616 92 84.62905  
## 617 94 82.87983  
## 618 99 89.69512  
## 619 93 80.89995  
## 620 87 77.82870  
## 621 82 76.45063  
## 622 72 78.45039  
## 623 97 76.99874  
## 624 93 87.95743  
## 625 79 92.81547  
## 626 79 80.81087  
## 627 81 78.03260  
## 628 62 81.21524  
## 629 76 79.27102  
## 630 80 80.71248  
## 631 61 81.70592  
## 632 57 78.63917  
## 633 113 75.29746  
## 634 87 72.33944  
## 635 21 76.57523  
## 636 77 81.96928  
## 637 72 82.33040  
## 638 76 85.74432  
## 639 95 85.14898  
## 640 70 84.98782  
## 641 82 90.15115  
## 642 88 87.60581  
## 643 85 92.92149  
## 644 95 95.31093  
## 645 79 81.02289  
## 646 72 76.88477  
## 647 82 84.77585  
## 648 103 88.93106  
## 649 79 82.10155  
## 650 58 89.52683  
## 651 74 86.02564  
## 652 87 79.37780  
## 653 98 84.48734  
## 654 88 82.93491  
## 655 79 68.22435  
## 656 75 75.99833  
## 657 79 85.70357  
## 658 95 79.57618  
## 659 64 70.89438  
## 660 76 89.49693  
## 661 54 77.55810  
## 662 64 81.40149  
## 663 79 73.90587  
## 664 65 73.55784  
## 665 72 83.10389  
## 666 81 78.39615  
## 667 80 77.05779  
## 668 80 80.72315  
## 669 83 73.46132  
## 670 65 61.90398  
## 671 82 62.28685  
## 672 102 73.70169  
## 673 89 74.54397  
## 674 65 73.84600  
## 675 80 61.25132  
## 676 91 79.47124  
## 677 92 83.26075  
## 678 72 71.60880  
## 679 83 83.55334  
## 680 94 75.68196  
## 681 75 76.36565  
## 682 75 80.20358  
## 683 82 79.05666  
## 684 84 77.81118  
## 685 70 83.28782  
## 686 66 74.22485  
## 687 99 76.82253  
## 688 88 77.12806  
## 689 94 80.63736  
## 690 86 77.08373  
## 691 94 71.73085  
## 692 100 74.57166  
## 693 85 79.22417  
## 694 88 86.49261  
## 695 89 82.07615  
## 696 96 86.23495  
## 697 88 96.73694  
## 698 95 83.63311  
## 699 73 82.47753  
## 700 95 79.80981  
## 701 78 80.53284  
## 702 90 81.83612  
## 703 88 76.42994  
## 704 77 73.20961  
## 705 86 75.03780  
## 706 64 74.59428  
## 707 69 84.51842  
## 708 96 84.17397  
## 709 71 85.30279  
## 710 78 78.78881  
## 711 56 84.07513  
## 712 67 79.38092  
## 713 81 80.61640  
## 714 75 78.04178  
## 715 45 84.50836  
## 716 73 83.54679  
## 717 89 80.08120  
## 718 70 82.91023  
## 719 69 78.49132  
## 720 67 83.35613  
## 721 67 73.32258  
## 722 67 71.67373  
## 723 66 98.19085  
## 724 77 98.01751  
## 725 75 78.46787  
## 726 81 89.91365  
## 727 85 75.16995  
## 728 77 75.50304  
## 729 81 67.49380  
## 730 80 79.75337  
## 731 69 83.97999  
## 732 95 85.34265  
## 733 50 82.35342  
## 734 66 80.79035  
## 735 86 75.78706  
## 736 65 74.82875  
## 737 90 75.08221  
## 738 98 82.44515  
## 739 91 78.18402  
## 740 71 76.62192  
## 741 61 84.27091  
## 742 84 78.70232  
## 743 93 80.78674  
## 744 99 76.73710  
## 745 95 76.18862  
## 746 116 68.60446  
## 747 106 71.96965  
## 748 93 84.90424  
## 749 99 80.17055  
## 750 93 83.99648  
## 751 108 86.59310  
## 752 103 75.47677  
## 753 106 75.53225  
## 754 72 86.19133  
## 755 99 85.54998  
## 756 100 89.02909  
## 757 99 86.18981  
## 758 103 97.91304  
## 759 79 98.63173  
## 760 91 92.47344  
## 761 89 104.48665  
## 762 86 97.33576  
## 763 74 100.85301  
## 764 92 92.93128  
## 765 96 82.82575  
## 766 96 87.45635  
## 767 101 85.89462  
## 768 87 82.10294  
## 769 87 89.28083  
## 770 101 92.29101  
## 771 110 72.82231  
## 772 108 72.27080  
## 773 66 67.86196  
## 774 56 62.46698  
## 775 74 74.97552  
## 776 104 82.33372  
## 777 63 73.87224  
## 778 56 76.05064  
## 779 56 77.17043  
## 780 52 82.98197  
## 781 74 77.89406  
## 782 66 88.99333  
## 783 59 86.81349  
## 784 62 86.70833  
## 785 82 85.03963  
## 786 54 73.59994  
## 787 77 81.58437  
## 788 76 75.50149  
## 789 78 72.12338  
## 790 74 68.86495  
## 791 102 80.02704  
## 792 69 77.39364  
## 793 91 78.07165  
## 794 100 77.63262  
## 795 87 82.97429  
## 796 83 79.14137  
## 797 84 89.78282  
## 798 78 89.27390  
## 799 93 83.86970  
## 800 46 77.30894  
## 801 82 69.58990  
## 802 68 72.98084  
## 803 45 88.69279  
## 804 92 87.92394  
## 805 49 91.70302  
## 806 82 82.67101  
## 807 76 71.85679  
## 808 63 68.10150  
## 809 61 81.84717  
## 810 80 78.12253  
## 811 70 69.83695  
## 812 77 76.46566  
## 813 65 86.11409  
## 814 105 90.88470  
## 815 92 91.80391  
## 816 103 92.11302  
## 817 91 84.80266  
## 818 85 74.65166  
## 819 73 77.57825  
## 820 89 76.78635  
## 821 99 82.36152  
## 822 90 82.50355  
## 823 91 88.04743  
## 824 88 87.72790  
## 825 72 76.63681  
## 826 82 75.42680  
## 827 85 82.96084  
## 828 73 78.21579  
## 829 95 76.27439  
## 830 75 73.79846  
## 831 53 81.10167  
## 832 69 75.42576  
## 833 94 76.49613  
## 834 74 74.59911  
## 835 80 89.20900  
## 836 88 87.22479  
## 837 92 75.04804  
## 838 98 71.78931  
## 839 57 78.02374  
## 840 95 76.61472  
## 841 98 63.46132  
## 842 75 73.67504  
## 843 73 77.48161  
## 844 72 80.45738  
## 845 67 78.22131  
## 846 63 80.05917  
## 847 60 80.85200  
## 848 81 80.57371  
## 849 92 86.57068  
## 850 89 85.80458  
## 851 91 83.86479  
## 852 74 83.11116  
## 853 66 86.62546  
## 854 88 86.96736  
## 855 57 81.63054  
## 856 67 84.51782  
## 857 73 83.66754  
## 858 83 90.72007  
## 859 88 84.56464  
## 860 76 79.70922  
## 861 116 80.94532  
## 862 78 79.55306  
## 863 103 83.57741  
## 864 88 76.30040  
## 865 79 79.15809  
## 866 92 72.63969  
## 867 97 77.97984  
## 868 58 80.06715  
## 869 70 75.38970  
## 870 83 80.66269  
## 871 73 75.27283  
## 872 89 83.24337  
## 873 90 74.94358  
## 874 80 81.51085  
## 875 75 83.24421  
## 876 86 84.04611  
## 877 90 77.97604  
## 878 92 75.08056  
## 879 42 72.00019  
## 880 68 78.40749  
## 881 61 74.36827  
## 882 58 69.61521  
## 883 82 78.99502  
## 884 82 76.21166  
## 885 97 79.95868  
## 886 106 85.97882  
## 887 76 74.10177  
## 888 87 74.43285  
## 889 101 76.14842  
## 890 97 76.55578  
## 891 103 71.64883  
## 892 112 88.44513  
## 893 89 82.81897  
## 894 83 79.19356  
## 895 93 74.99922  
## 896 80 84.37986  
## 897 83 74.92339  
## 898 97 81.98816  
## 899 69 83.05794  
## 900 72 79.14053  
## 901 86 76.41732  
## 902 57 77.00170  
## 903 83 83.40517  
## 904 88 77.80671  
## 905 72 80.72405  
## 906 86 84.92448  
## 907 96 76.37162  
## 908 87 80.14464  
## 909 89 81.99346  
## 910 86 79.12052  
## 911 80 63.72078  
## 912 71 60.35400  
## 913 76 66.78161  
## 914 86 60.68783  
## 915 82 59.18272  
## 916 65 70.53416  
## 917 67 81.46170  
## 918 68 71.45605  
## 919 82 74.57813  
## 920 80 71.72674  
## 921 85 75.34780  
## 922 78 81.52163  
## 923 87 83.94570  
## 924 81 91.68011  
## 925 91 85.36356  
## 926 87 81.99874  
## 927 85 84.14420  
## 928 70 73.31153  
## 929 82 78.85476  
## 930 75 76.00103  
## 931 92 77.32058  
## 932 92 73.77252  
## 933 98 90.41431  
## 934 51 85.21387  
## 935 111 73.53234  
## 936 86 76.49770  
## 937 55 70.54324  
## 938 74 84.21458  
## 939 66 97.21267  
## 940 70 80.64946  
## 941 63 79.82280  
## 942 81 76.58877  
## 943 87 76.01548  
## 944 90 75.33370  
## 945 80 71.64498  
## 946 87 84.87058  
## 947 84 79.01173  
## 948 88 81.63345  
## 949 86 77.29707  
## 950 81 79.07006  
## 951 76 78.70672  
## 952 70 98.16423  
## 953 67 85.26197  
## 954 82 90.39484  
## 955 65 94.47383  
## 956 95 104.75405  
## 957 90 92.78122  
## 958 83 93.66794  
## 959 88 95.31631  
## 960 66 102.26987  
## 961 56 95.77832  
## 962 66 93.74198  
## 963 45 87.68429  
## 964 69 85.41279  
## 965 61 86.50211  
## 966 97 86.91786  
## 967 84 86.91658  
## 968 97 92.97927  
## 969 54 100.04520  
## 970 76 92.21314  
## 971 85 86.56469  
## 972 74 90.68067  
## 973 104 89.20259  
## 974 94 95.19330  
## 975 107 84.80754  
## 976 100 85.47340  
## 977 64 79.99373  
## 978 94 66.40110  
## 979 45 74.33150  
## 980 68 80.55107  
## 981 78 77.92324  
## 982 87 75.59030  
## 983 80 74.68523  
## 984 92 73.58853  
## 985 85 79.11850  
## 986 84 86.74757  
## 987 98 91.30102  
## 988 84 80.72652  
## 989 88 91.67552  
## 990 97 85.58264  
## 991 78 85.24275  
## 992 85 79.61319  
## 993 95 94.02997  
## 994 75 91.34655  
## 995 62 86.76379  
## 996 114 83.15531  
## 997 110 80.73357  
## 998 98 78.38100  
## 999 86 82.05406  
## 1000 88 84.55653  
## 1001 87 85.12766  
## 1002 64 85.11712  
## 1003 68 94.73952  
## 1004 65 92.33759  
## 1005 89 89.29616  
## 1006 75 89.76105  
## 1007 81 87.44643  
## 1008 79 91.54011  
## 1009 80 93.41918  
## 1010 71 92.01233  
## 1011 76 91.22298  
## 1012 77 93.76596  
## 1013 82 80.15303  
## 1014 65 74.74933  
## 1015 79 75.69253  
## 1016 93 76.00785  
## 1017 83 71.17943  
## 1018 83 66.08970  
## 1019 63 80.56114  
## 1020 53 85.16664  
## 1021 56 87.91888  
## 1022 79 74.29519  
## 1023 90 73.74423  
## 1024 71 72.48966  
## 1025 77 76.45746  
## 1026 78 79.55431  
## 1027 54 82.36934  
## 1028 94 77.09259  
## 1029 99 85.04632  
## 1030 86 87.48039  
## 1031 99 94.97976  
## 1032 92 91.99768  
## 1033 95 90.99592  
## 1034 67 89.30324  
## 1035 79 81.71456  
## 1036 75 79.33073  
## 1037 64 89.00786  
## 1038 90 87.12255  
## 1039 74 81.65797  
## 1040 71 81.35012  
## 1041 87 68.60934  
## 1042 71 67.88631  
## 1043 77 68.54235  
## 1044 80 67.56090  
## 1045 75 69.46154  
## 1046 95 82.63933  
## 1047 93 85.29402  
## 1048 69 74.25824  
## 1049 77 76.50022  
## 1050 63 79.50063  
## 1051 64 78.73351  
## 1052 111 77.56332  
## 1053 62 71.55565  
## 1054 62 70.39303  
## 1055 78 74.11267  
## 1056 102 85.19926  
## 1057 93 78.36045  
## 1058 94 77.45106  
## 1059 64 83.83186  
## 1060 77 76.32044  
## 1061 73 86.73581  
## 1062 79 83.39038  
## 1063 80 83.93669  
## 1064 70 83.00696  
## 1065 86 86.60338  
## 1066 87 86.10335  
## 1067 108 88.47441  
## 1068 91 83.14315  
## 1069 86 91.80580  
## 1070 81 88.58660  
## 1071 76 82.50243  
## 1072 85 79.89829  
## 1073 66 91.77266  
## 1074 69 85.07039  
## 1075 92 83.01118  
## 1076 94 85.68794  
## 1077 99 95.76558  
## 1078 93 90.58290  
## 1079 87 91.94511  
## 1080 82 96.31662  
## 1081 72 84.96631  
## 1082 97 69.83334  
## 1083 93 73.56281  
## 1084 79 72.00398  
## 1085 79 74.46932  
## 1086 81 73.08326  
## 1087 62 71.27250  
## 1088 76 84.53501  
## 1089 80 83.59480  
## 1090 61 81.98597  
## 1091 57 86.48434  
## 1092 113 81.75761  
## 1093 87 76.77313  
## 1094 21 76.37420  
## 1095 77 75.89710  
## 1096 72 77.61684  
## 1097 76 81.74684  
## 1098 95 81.41920  
## 1099 70 76.82860  
## 1100 82 74.59177  
## 1101 88 79.96919  
## 1102 85 94.95448  
## 1103 95 83.66543  
## 1104 79 76.84529  
## 1105 72 87.13900  
## 1106 82 71.93989  
## 1107 103 78.74975  
## 1108 79 78.76011  
## 1109 58 66.61349  
## 1110 74 70.79383  
## 1111 87 70.84328  
## 1112 98 71.91145  
## 1113 88 67.12428  
## 1114 79 71.62438  
## 1115 75 71.33810  
## 1116 79 71.90173  
## 1117 95 73.78630  
## 1118 64 81.48361  
## 1119 76 84.80836  
## 1120 54 83.19514  
## 1121 64 81.29440  
## 1122 79 73.80876  
## 1123 65 67.63890  
## 1124 72 73.59700  
## 1125 81 79.16576  
## 1126 80 72.63248  
## 1127 80 72.41918  
## 1128 83 73.04657  
## 1129 65 66.49953  
## 1130 82 73.95977  
## 1131 102 71.42417  
## 1132 89 69.30896  
## 1133 65 71.54921  
## 1134 80 76.78362  
## 1135 91 82.63552  
## 1136 92 95.12850  
## 1137 72 83.90767  
## 1138 83 84.47296  
## 1139 94 87.77827  
## 1140 75 80.02842  
## 1141 75 84.42900  
## 1142 82 85.19428  
## 1143 84 73.29018  
## 1144 70 78.51938  
## 1145 66 75.82602  
## 1146 99 71.69911  
## 1147 88 75.95149  
## 1148 94 85.64490  
## 1149 86 83.34346  
## 1150 94 77.77329  
## 1151 100 74.54778  
## 1152 85 86.37762  
## 1153 88 80.14170  
## 1154 89 83.43499  
## 1155 96 85.01173  
## 1156 88 79.65990  
## 1157 95 97.22181  
## 1158 73 78.07314  
## 1159 95 91.60057  
## 1160 78 92.29360  
## 1161 90 92.95479  
## 1162 88 89.26299  
## 1163 77 79.41539  
## 1164 86 80.53341  
## 1165 64 73.59357  
## 1166 69 81.24245  
## 1167 96 83.56699  
## 1168 71 82.38249  
## 1169 78 93.33796  
## 1170 56 85.35170  
## 1171 67 76.41227  
## 1172 81 79.07561  
## 1173 75 72.52526  
## 1174 45 95.66395  
## 1175 73 87.66176  
## 1176 89 92.90300  
## 1177 70 99.28524  
## 1178 69 86.39980  
## 1179 67 81.58923  
## 1180 67 85.33114  
## 1181 67 93.70586  
## 1182 66 93.36064  
## 1183 77 77.33035  
## 1184 75 79.55576  
## 1185 81 75.40082  
## 1186 85 79.76631  
## 1187 77 81.47024  
## 1188 81 80.17441  
## 1189 80 77.22722  
## 1190 69 81.74268  
## 1191 95 78.60004  
## 1192 50 84.63944  
## 1193 66 79.37534  
## 1194 86 76.55477  
## 1195 65 78.77538  
## 1196 90 85.52381  
## 1197 98 84.44809  
## 1198 91 77.59152  
## 1199 71 87.63939  
## 1200 61 86.29028  
## 1201 84 84.57447  
## 1202 93 87.19745  
## 1203 99 77.20342  
## 1204 95 73.46173  
## 1205 116 73.29697  
## 1206 106 78.67011  
## 1207 93 74.68087  
## 1208 99 71.33847  
## 1209 93 70.73626  
## 1210 108 70.52728  
## 1211 103 78.88400  
## 1212 106 71.53207  
## 1213 72 71.39736  
## 1214 99 74.93688  
## 1215 100 81.09400  
## 1216 99 72.57754  
## 1217 103 75.39038  
## 1218 79 75.68757  
## 1219 91 80.16107  
## 1220 89 89.40135  
## 1221 86 72.16460  
## 1222 74 83.38028  
## 1223 92 81.32313  
## 1224 96 84.89702  
## 1225 96 84.84828  
## 1226 101 80.52925  
## 1227 87 76.62719  
## 1228 87 79.65161  
## 1229 101 79.83045  
## 1230 110 82.55894  
## 1231 108 73.84234  
## 1232 66 81.12999  
## 1233 56 73.31931  
## 1234 74 63.40639  
## 1235 104 75.90363  
## 1236 63 67.21770  
## 1237 56 70.21542  
## 1238 56 68.87801  
## 1239 52 72.77913  
## 1240 74 70.17411  
## 1241 66 71.05437  
## 1242 59 81.34791  
## 1243 62 79.48590  
## 1244 82 80.68455  
## 1245 54 72.75448  
## 1246 77 74.81253  
## 1247 76 78.33119  
## 1248 78 80.06916  
## 1249 74 73.74964  
## 1250 102 74.50480  
## 1251 69 75.05737  
## 1252 91 80.77484  
## 1253 100 80.54898  
## 1254 87 80.71122  
## 1255 83 80.60354  
## 1256 84 84.06425  
## 1257 78 81.92056  
## 1258 93 74.13072  
## 1259 46 74.34099  
## 1260 82 82.79877  
## 1261 68 78.63570  
## 1262 45 79.79308  
## 1263 92 86.05803  
## 1264 49 77.78018  
## 1265 82 79.84836  
## 1266 76 84.47589  
## 1267 63 79.79337  
## 1268 61 76.72617  
## 1269 80 79.21975  
## 1270 70 80.25741  
## 1271 77 79.14570  
## 1272 65 83.44643  
## 1273 105 93.98953  
## 1274 92 94.00703  
## 1275 103 93.68490  
## 1276 91 91.38872  
## 1277 85 93.75317  
## 1278 73 92.88496  
## 1279 89 91.70101  
## 1280 99 85.48907  
## 1281 90 80.65960  
## 1282 91 74.69654  
## 1283 88 75.32830  
## 1284 72 81.12020  
## 1285 82 70.93756  
## 1286 85 70.80932  
## 1287 73 74.57660  
## 1288 95 85.39189  
## 1289 75 86.64623  
## 1290 53 84.02929  
## 1291 69 79.94081  
## 1292 94 78.06523  
## 1293 74 84.11563  
## 1294 80 88.72233  
## 1295 88 86.22934  
## 1296 92 79.80524  
## 1297 98 86.74477  
## 1298 57 87.62938  
## 1299 95 92.83543  
## 1300 98 94.36469  
## 1301 75 86.01547  
## 1302 73 80.54337  
## 1303 72 73.29776  
## 1304 67 78.68654  
## 1305 63 82.39342  
## 1306 60 78.33648  
## 1307 81 79.95985  
## 1308 92 81.96134  
## 1309 89 81.59286  
## 1310 91 79.05989  
## 1311 74 77.86516  
## 1312 66 82.34401  
## 1313 88 75.16981  
## 1314 57 80.07663  
## 1315 67 82.51050  
## 1316 73 82.61926  
## 1317 83 96.35798  
## 1318 88 91.78498  
## 1319 76 91.83003  
## 1320 116 96.02732  
## 1321 78 86.45282  
## 1322 103 82.53612  
## 1323 88 86.96077  
## 1324 79 87.22632  
## 1325 92 84.18313  
## 1326 97 83.26233  
## 1327 58 89.16195  
## 1328 70 81.82784  
## 1329 83 80.15164  
## 1330 73 76.91678  
## 1331 89 72.78148  
## 1332 90 70.53999  
## 1333 80 78.85112  
## 1334 75 81.40689  
## 1335 86 83.17569  
## 1336 90 87.87173  
## 1337 92 75.63225  
## 1338 42 78.58231  
## 1339 68 73.46579  
## 1340 61 78.71896  
## 1341 58 82.70960  
## 1342 82 73.33920  
## 1343 82 77.90761  
## 1344 97 79.98687  
## 1345 106 70.52942  
## 1346 76 84.46694  
## 1347 87 98.40246  
## 1348 101 86.63365  
## 1349 97 78.47395  
## 1350 103 85.92126  
## 1351 112 83.71443  
## 1352 89 87.85776  
## 1353 83 94.71885  
## 1354 93 90.06946  
## 1355 80 81.89370  
## 1356 83 71.95118  
## 1357 97 75.62014  
## 1358 69 88.90679  
## 1359 72 95.62316  
## 1360 86 72.47442  
## 1361 57 97.62852  
## 1362 83 88.88781  
## 1363 88 73.04341  
## 1364 72 74.20625  
## 1365 86 74.85814  
## 1366 96 75.56058  
## 1367 87 67.96074  
## 1368 89 64.83668  
## 1369 86 66.12139  
## 1370 80 77.36962  
## 1371 71 86.11549  
## 1372 76 88.08887  
## 1373 86 84.97467  
## 1374 82 79.91790  
## 1375 65 82.47619  
## 1376 67 76.35949  
## 1377 68 78.68785  
## 1378 82 76.76146  
## 1379 80 90.75052  
## 1380 85 83.07884  
## 1381 78 90.01429  
## 1382 87 88.37454  
## 1383 81 92.99445  
## 1384 91 90.64963  
## 1385 87 90.30104  
## 1386 85 83.17853  
## 1387 70 79.07535  
## 1388 82 77.46767  
## 1389 75 84.16275  
## 1390 92 79.55429  
## 1391 92 80.16002  
## 1392 98 80.89242  
## 1393 51 83.19119  
## 1394 111 81.77316  
## 1395 86 88.11476  
## 1396 55 78.96010  
## 1397 74 82.05215  
## 1398 66 78.88763  
## 1399 70 79.95473  
## 1400 63 87.94550  
## 1401 81 84.21109  
## 1402 87 82.94389  
## 1403 90 81.63757  
## 1404 80 82.86589  
## 1405 87 82.07041  
## 1406 84 88.16516  
## 1407 88 83.92208  
## 1408 86 82.40573  
## 1409 81 82.94323  
## 1410 76 79.72355  
## 1411 70 80.43830  
## 1412 67 81.25349  
## 1413 82 77.38370  
## 1414 65 78.72661  
## 1415 95 78.92403  
## 1416 90 73.09722  
## 1417 83 76.51427  
## 1418 88 75.89040  
## 1419 66 75.09768  
## 1420 56 69.03655  
## 1421 66 73.11790  
## 1422 45 80.32345  
## 1423 69 81.62448  
## 1424 61 72.46322  
## 1425 97 73.05070  
## 1426 84 82.60617  
## 1427 97 76.80164  
## 1428 54 74.71730  
## 1429 76 72.53451  
## 1430 85 79.24098  
## 1431 74 80.59521  
## 1432 104 77.15610  
## 1433 94 74.10392  
## 1434 107 79.46361  
## 1435 100 81.59949  
## 1436 64 92.84525  
## 1437 94 86.74162  
## 1438 45 89.00812  
## 1439 68 83.81009  
## 1440 78 82.45990  
## 1441 87 86.81278  
## 1442 80 87.08329  
## 1443 92 81.02582  
## 1444 85 81.79036  
## 1445 84 76.09764  
## 1446 98 79.58062  
## 1447 84 70.85242  
## 1448 88 73.77270  
## 1449 97 78.06725  
## 1450 78 82.56246  
## 1451 85 82.22541  
## 1452 95 82.97301  
## 1453 75 77.64168  
## 1454 62 79.54539  
## 1455 114 68.24060  
## 1456 110 70.24206  
## 1457 98 67.33869  
## 1458 86 80.95609  
## 1459 88 77.96052  
## 1460 87 70.39466  
## 1461 64 75.42098  
## 1462 68 79.98365  
## 1463 65 72.26114  
## 1464 89 80.33432  
## 1465 75 72.40334  
## 1466 81 76.28209  
## 1467 79 77.74120  
## 1468 80 76.98433  
## 1469 71 74.78655  
## 1470 76 84.57421  
## 1471 77 77.67434  
## 1472 82 84.70901  
## 1473 65 78.88817  
## 1474 79 86.51928  
## 1475 93 94.59262  
## 1476 83 83.68060  
## 1477 83 90.09426  
## 1478 63 86.45699  
## 1479 53 85.18612  
## 1480 56 73.40551  
## 1481 79 73.27707  
## 1482 90 91.99847  
## 1483 71 86.52118  
## 1484 77 92.42330  
## 1485 78 84.04087  
## 1486 54 83.86156  
## 1487 94 85.52067  
## 1488 99 80.89763  
## 1489 86 86.58041  
## 1490 99 91.77189  
## 1491 92 82.89224  
## 1492 95 78.98014  
## 1493 67 87.14506  
## 1494 79 89.37809  
## 1495 75 74.04740  
## 1496 64 75.81540  
## 1497 90 84.01966  
## 1498 74 88.51943  
## 1499 71 72.64024  
## 1500 87 79.09215  
## 1501 71 70.49611  
## 1502 77 79.19742  
## 1503 80 74.80890  
## 1504 75 70.05598  
## 1505 95 79.93610  
## 1506 93 71.64301  
## 1507 69 81.23458  
## 1508 77 84.56800  
## 1509 63 81.05502  
## 1510 64 77.11747  
## 1511 111 78.69532  
## 1512 62 76.56764  
## 1513 62 74.31725  
## 1514 78 79.24925  
## 1515 102 77.95751  
## 1516 93 73.35033  
## 1517 94 82.50775  
## 1518 64 77.38719  
## 1519 77 79.51796  
## 1520 73 67.76209  
## 1521 79 80.89661

mean (apply(predicted\_values, 1, min)/apply(predicted\_values, 1, max)) # calculate accuracy

## [1] 0.8625792

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  
## 2 82 77.58814  
## 4 80 77.10002  
## 5 85 80.31237  
## 7 78 74.88453  
## 8 87 84.25671  
## 13 81 79.89083  
## 18 91 85.86964  
## 21 87 88.63322  
## 23 85 84.05879  
## 24 70 73.26257  
## 25 82 83.95873  
## 26 75 77.57260  
## 29 92 90.53511  
## 34 92 90.74698  
## 35 98 95.14071  
## 37 51 59.85753  
## 39 111 100.90621  
## 42 86 83.50799  
## 52 55 63.91588  
## 53 74 76.70358  
## 57 66 70.21566  
## 59 70 72.75631  
## 60 63 68.30186  
## 65 81 78.72088  
## 66 87 81.23349  
## 78 90 87.88129  
## 82 80 81.42072  
## 86 87 85.90553  
## 87 84 81.24384  
## 88 88 85.21943  
## 89 86 82.31661  
## 92 81 78.41031  
## 93 76 78.60334  
## 98 70 71.75496  
## 101 67 70.62713  
## 102 82 78.35821  
## 106 65 69.17566  
## 114 95 91.76035  
## 117 90 87.88443  
## 122 83 80.39256  
## 124 88 81.32096  
## 125 66 66.10715  
## 128 56 62.61049  
## 136 66 72.00023  
## 139 45 58.94235  
## 144 69 70.99145  
## 148 61 68.40904  
## 159 97 90.49464  
## 167 84 80.46060  
## 168 97 93.05296  
## 182 54 61.61181  
## 184 76 78.96327  
## 187 85 85.15068  
## 196 74 76.89193  
## 199 104 99.35933  
## 202 94 91.14382  
## 204 107 97.67330  
## 205 100 90.70641  
## 208 64 65.45619  
## 210 94 87.06168  
## 225 45 55.57485  
## 226 68 73.31081  
## 229 78 81.33232  
## 237 87 86.69332  
## 238 80 79.13771  
## 243 92 86.67803  
## 246 85 83.06708  
## 248 84 82.52686  
## 250 98 92.34159  
## 252 84 82.72442  
## 256 88 85.71802  
## 261 97 94.10501  
## 263 78 80.04023  
## 266 85 85.46132  
## 268 95 94.99322  
## 271 75 78.15942  
## 272 62 67.95918  
## 275 114 99.69396  
## 277 110 96.71489  
## 280 98 93.68179  
## 286 86 83.19599  
## 291 88 87.55503  
## 304 87 84.99691  
## 307 64 67.56353  
## 308 68 70.96448  
## 309 65 69.26030  
## 322 89 86.20832  
## 324 75 76.87183  
## 325 81 80.40220  
## 326 79 79.52186  
## 327 80 79.39506  
## 331 71 74.69175  
## 332 76 78.46869  
## 335 77 78.21737  
## 338 82 82.97846  
## 343 65 72.89054  
## 346 79 80.00084  
## 353 93 85.04770  
## 355 83 79.06732  
## 357 83 78.97431  
## 361 63 68.83993  
## 364 53 61.99379  
## 366 56 65.52321  
## 367 79 81.83210  
## 369 90 87.79088  
## 370 71 75.21541  
## 375 77 77.08779  
## 376 78 76.75046  
## 378 54 63.06412  
## 380 94 90.35000  
## 381 99 94.57933  
## 385 86 84.50757  
## 386 99 93.39128  
## 387 92 89.54919  
## 391 95 88.10821  
## 392 67 67.14836  
## 394 79 78.69729  
## 397 75 77.21986  
## 398 64 69.10124  
## 399 90 89.05717  
## 401 74 76.07051  
## 402 71 71.48790  
## 403 87 85.28027  
## 409 71 71.63595  
## 413 77 83.09511  
## 415 80 82.09815  
## 416 75 78.60410  
## 417 95 92.34197  
## 427 93 87.39177  
## 429 69 70.63887  
## 431 77 75.10014  
## 434 63 67.23094  
## 435 64 68.32685  
## 438 111 100.60597  
## 442 62 68.55769  
## 445 62 64.64742  
## 448 78 78.23057  
## 451 102 94.36657  
## 453 93 86.61589  
## 455 94 86.68250  
## 456 64 67.28791  
## 458 77 78.89907  
## 463 73 74.94612  
## 464 79 81.00199  
## 466 80 81.73902  
## 468 70 73.75221  
## 471 86 82.27548  
## 475 87 82.20550  
## 480 108 101.36544  
## 484 91 88.19970  
## 488 86 84.97441  
## 494 81 82.75519  
## 495 76 78.43185  
## 498 85 85.76494  
## 499 66 70.57798  
## 501 69 71.30254  
## 506 92 86.07157  
## 508 94 88.32014  
## 518 99 96.44675  
## 521 93 88.45163  
## 527 87 87.29920  
## 533 82 80.71250  
## 534 72 73.79423  
## 539 97 92.03838  
## 541 93 89.83518  
## 545 79 77.26149  
## 546 79 78.39392  
## 547 81 78.25079  
## 550 62 66.76745  
## 551 76 75.82642  
## 557 80 83.23144  
## 559 61 69.12355  
## 563 57 62.78660  
## 566 113 106.42571  
## 568 87 88.46492  
## 577 21 38.95215  
## 581 77 79.37503  
## 582 72 77.49115  
## 589 76 78.99462  
## 594 95 87.06096  
## 595 70 75.55117  
## 604 82 83.07090  
## 609 88 88.71450  
## 610 85 86.50145  
## 611 95 94.47437  
## 612 79 80.77058  
## 621 72 74.90374  
## 622 82 81.31000  
## 629 103 94.10193  
## 630 79 79.64710  
## 634 58 62.92561  
## 635 74 74.35784  
## 643 87 87.29672  
## 644 98 95.39103  
## 645 88 86.11955  
## 647 79 80.20395  
## 649 75 78.17972  
## 653 79 81.59405  
## 658 95 89.62915  
## 659 64 68.57931  
## 660 76 79.38772  
## 663 54 62.12229  
## 664 64 69.13166  
## 665 79 80.30159  
## 669 65 68.72162  
## 673 72 73.66237  
## 676 81 80.85653  
## 683 80 79.83631  
## 687 80 79.60955  
## 688 83 80.83230  
## 691 65 68.61621  
## 695 82 82.23927  
## 696 102 97.25261  
## 700 89 86.31379  
## 704 65 68.42147  
## 706 80 79.33094  
## 709 91 86.89104  
## 713 92 88.76536  
## 717 72 74.75530  
## 720 83 83.52297  
## 726 94 90.89906  
## 735 75 74.78864  
## 740 75 76.53695  
## 741 82 82.71854  
## 742 84 81.91350  
## 746 70 72.74042  
## 747 66 70.03425  
## 756 99 95.69741  
## 757 88 86.72710  
## 759 94 92.31084  
## 768 86 84.23802  
## 769 94 90.35474  
## 770 100 94.61573  
## 775 85 82.97080  
## 776 88 86.77427  
## 777 89 84.85451  
## 779 96 89.73534  
## 780 88 86.66407  
## 787 95 89.59599  
## 788 73 73.73932  
## 789 95 87.32776  
## 790 78 76.56196  
## 794 90 85.81143  
## 795 88 86.68017  
## 797 77 80.51344  
## 799 86 84.40557  
## 806 64 69.53342  
## 807 69 70.12851  
## 814 96 92.81228  
## 818 71 73.53953  
## 825 78 77.71461  
## 830 56 64.07635  
## 832 67 71.58380  
## 833 81 80.45667  
## 834 75 76.29805  
## 841 45 52.28863  
## 848 73 74.94793  
## 849 89 88.42641  
## 855 70 75.68097  
## 859 69 74.64608  
## 860 67 71.67173  
## 862 67 72.64780  
## 864 67 70.48897  
## 874 66 69.78993  
## 875 77 77.73683  
## 881 75 76.65285  
## 883 81 81.83649  
## 886 85 84.87760  
## 887 77 76.10391  
## 891 81 79.84921  
## 894 80 81.36571  
## 902 69 74.50646  
## 904 95 91.64217  
## 912 50 54.86170  
## 913 66 66.82942  
## 919 86 83.23011  
## 920 65 68.07593  
## 921 90 85.22906  
## 922 98 92.27276  
## 927 91 88.31299  
## 931 71 73.33092  
## 947 61 67.12971  
## 948 84 82.23334  
## 951 93 87.85509  
## 953 99 94.02444  
## 954 95 92.25779  
## 956 116 109.15701  
## 957 106 99.79448  
## 958 93 91.73027  
## 959 99 96.44072  
## 961 93 91.27314  
## 963 108 99.14686  
## 970 103 96.29935  
## 972 106 99.20626  
## 978 72 71.27987  
## 986 99 93.60541  
## 987 100 95.27237  
## 988 99 93.17939  
## 989 103 97.16364  
## 990 79 80.60452  
## 991 91 88.40028  
## 995 89 87.89961  
## 996 86 84.97801  
## 997 74 76.19600  
## 1001 92 91.85754  
## 1002 96 94.05218  
## 1006 96 92.66122  
## 1008 101 97.45470  
## 1014 87 82.29999  
## 1016 87 82.61996  
## 1019 101 92.13341  
## 1020 110 99.35549  
## 1021 108 100.27908  
## 1024 66 67.60624  
## 1026 56 64.38594  
## 1028 74 75.48517  
## 1030 104 97.98325  
## 1035 63 70.03174  
## 1036 56 65.17295  
## 1038 56 67.36491  
## 1042 52 58.97439  
## 1047 74 77.76308  
## 1049 66 69.90360  
## 1052 59 65.61664  
## 1054 62 66.19679  
## 1055 82 79.07380  
## 1060 54 61.32659  
## 1062 77 78.90117  
## 1063 76 78.22970  
## 1070 78 81.02123  
## 1072 74 76.72428  
## 1074 102 95.77534  
## 1075 69 73.40292  
## 1076 91 88.08616  
## 1079 100 95.19529  
## 1086 87 81.70110  
## 1088 83 82.03596  
## 1089 84 82.62498  
## 1092 78 78.73064  
## 1094 93 86.52565  
## 1101 46 58.32518  
## 1105 82 83.45294  
## 1106 68 70.30356  
## 1112 45 55.32082  
## 1121 92 87.89394  
## 1124 49 57.99122  
## 1125 82 80.67792  
## 1129 76 73.49735  
## 1130 63 67.44821  
## 1132 61 65.73689  
## 1134 80 78.68736  
## 1147 70 73.44992  
## 1152 77 81.88798  
## 1153 65 72.31074  
## 1162 105 97.85936  
## 1163 92 86.60282  
## 1166 103 94.03893  
## 1168 91 87.05159  
## 1170 85 84.21651  
## 1171 73 73.54876  
## 1176 89 89.88359  
## 1179 99 93.93935  
## 1184 90 86.69156  
## 1187 91 88.91604  
## 1188 88 87.27198  
## 1191 72 75.12200  
## 1192 82 82.66817  
## 1193 85 83.74225  
## 1194 73 75.01250  
## 1196 95 91.59964  
## 1201 75 77.30408  
## 1203 53 62.21462  
## 1204 69 70.73690  
## 1207 94 86.91624  
## 1208 74 72.57387  
## 1213 80 77.50738  
## 1214 88 84.08865  
## 1219 92 87.23719  
## 1220 98 92.92086  
## 1221 57 64.41999  
## 1224 95 90.83643  
## 1225 98 93.51719  
## 1227 75 77.47774  
## 1228 73 76.39512  
## 1231 72 73.04201  
## 1233 67 70.24594  
## 1235 63 67.18192  
## 1238 60 63.60912  
## 1243 81 80.39053  
## 1244 92 87.22138  
## 1247 89 84.70466  
## 1253 91 88.63550  
## 1256 74 77.09069  
## 1258 66 70.15209  
## 1260 88 87.46766  
## 1262 57 64.65424  
## 1263 67 72.85011  
## 1268 73 74.21985  
## 1270 83 82.20183  
## 1273 88 89.46885  
## 1276 76 80.63727  
## 1279 116 108.39614  
## 1284 78 80.11958  
## 1287 103 93.96849  
## 1289 88 85.53023  
## 1296 79 78.30483  
## 1300 92 91.52416  
## 1303 97 88.58794  
## 1313 58 64.74848  
## 1314 70 73.99430  
## 1315 83 83.21693  
## 1323 73 74.88670  
## 1327 89 87.17632  
## 1333 90 86.12250  
## 1338 80 80.20127  
## 1348 75 77.89806  
## 1352 86 87.38507  
## 1354 90 88.65523  
## 1358 92 89.80546  
## 1360 42 52.90818  
## 1363 68 69.92670  
## 1366 61 66.16366  
## 1370 58 65.79814  
## 1372 82 83.33219  
## 1381 82 83.38778  
## 1385 97 94.01207  
## 1386 106 97.33014  
## 1387 76 77.18003  
## 1388 87 84.68333  
## 1389 101 94.48236  
## 1394 97 92.38572  
## 1396 103 95.29474  
## 1397 112 101.49698  
## 1401 89 87.91192  
## 1403 83 82.64675  
## 1415 93 88.40713  
## 1417 80 79.20350  
## 1418 83 78.95669  
## 1440 97 91.80469  
## 1445 69 72.49412  
## 1456 72 71.71197  
## 1458 86 83.90352  
## 1461 57 63.91965  
## 1469 83 81.29703  
## 1474 88 88.35827  
## 1478 72 77.42839  
## 1483 86 86.05525  
## 1484 96 93.06745  
## 1485 87 85.68841  
## 1486 89 86.21777  
## 1487 86 86.40816  
## 1496 80 80.55728  
## 1501 71 71.88875  
## 1506 76 75.13397  
## 1508 86 85.43600  
## 1509 82 82.82999  
## 1513 65 68.49871  
## 1515 67 71.38261  
## 1516 68 71.45643

Lets calculate the accuracy of using Model6 for our predictions

mean (apply(predicted\_values, 1, min)/apply(predicted\_values, 1, max)) # calculate accuracy

## [1] 0.9547124

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.