Homework Assignment #1

Alain Kuiete

9/10/2020

## INTRODUCTION

Study of 2276 professionals baseball teams from 1871 to 2006. There are 16 variables where 15 are predictors. ## DATA EXPLORATION

We can use the command read.csv to import the dataset and view the first six row with the command head().

moneyball <- read.csv("/home/alain/Documents/DATA621/Assignment1/moneyball-training-data.csv")  
moneyball\_eval <- read.csv("/home/alain/Documents/DATA621/Assignment1/moneyball-evaluation-data.csv")  
head(moneyball)

## INDEX TARGET\_WINS TEAM\_BATTING\_H TEAM\_BATTING\_2B TEAM\_BATTING\_3B  
## 1 1 39 1445 194 39  
## 2 2 70 1339 219 22  
## 3 3 86 1377 232 35  
## 4 4 70 1387 209 38  
## 5 5 82 1297 186 27  
## 6 6 75 1279 200 36  
## TEAM\_BATTING\_HR TEAM\_BATTING\_BB TEAM\_BATTING\_SO TEAM\_BASERUN\_SB  
## 1 13 143 842 NA  
## 2 190 685 1075 37  
## 3 137 602 917 46  
## 4 96 451 922 43  
## 5 102 472 920 49  
## 6 92 443 973 107  
## TEAM\_BASERUN\_CS TEAM\_BATTING\_HBP TEAM\_PITCHING\_H TEAM\_PITCHING\_HR  
## 1 NA NA 9364 84  
## 2 28 NA 1347 191  
## 3 27 NA 1377 137  
## 4 30 NA 1396 97  
## 5 39 NA 1297 102  
## 6 59 NA 1279 92  
## TEAM\_PITCHING\_BB TEAM\_PITCHING\_SO TEAM\_FIELDING\_E TEAM\_FIELDING\_DP  
## 1 927 5456 1011 NA  
## 2 689 1082 193 155  
## 3 602 917 175 153  
## 4 454 928 164 156  
## 5 472 920 138 168  
## 6 443 973 123 149

All the variables are numeric. The summary and describe function gives the univariate statistic of each variable. For each variable there are computation of minimun, maximun, mean, median, first and third quantiles. The describe function also include the standard deviation, the degree of skweness and the degree of kurtosis. For a quick univariate statistics of the datasets, the function summary is convenient.

summary(moneyball[,-1])

## TARGET\_WINS TEAM\_BATTING\_H TEAM\_BATTING\_2B TEAM\_BATTING\_3B   
## Min. : 0.00 Min. : 891 Min. : 69.0 Min. : 0.00   
## 1st Qu.: 71.00 1st Qu.:1383 1st Qu.:208.0 1st Qu.: 34.00   
## Median : 82.00 Median :1454 Median :238.0 Median : 47.00   
## Mean : 80.79 Mean :1469 Mean :241.2 Mean : 55.25   
## 3rd Qu.: 92.00 3rd Qu.:1537 3rd Qu.:273.0 3rd Qu.: 72.00   
## Max. :146.00 Max. :2554 Max. :458.0 Max. :223.00   
##   
## TEAM\_BATTING\_HR TEAM\_BATTING\_BB TEAM\_BATTING\_SO TEAM\_BASERUN\_SB  
## Min. : 0.00 Min. : 0.0 Min. : 0.0 Min. : 0.0   
## 1st Qu.: 42.00 1st Qu.:451.0 1st Qu.: 548.0 1st Qu.: 66.0   
## Median :102.00 Median :512.0 Median : 750.0 Median :101.0   
## Mean : 99.61 Mean :501.6 Mean : 735.6 Mean :124.8   
## 3rd Qu.:147.00 3rd Qu.:580.0 3rd Qu.: 930.0 3rd Qu.:156.0   
## Max. :264.00 Max. :878.0 Max. :1399.0 Max. :697.0   
## NA's :102 NA's :131   
## TEAM\_BASERUN\_CS TEAM\_BATTING\_HBP TEAM\_PITCHING\_H TEAM\_PITCHING\_HR  
## Min. : 0.0 Min. :29.00 Min. : 1137 Min. : 0.0   
## 1st Qu.: 38.0 1st Qu.:50.50 1st Qu.: 1419 1st Qu.: 50.0   
## Median : 49.0 Median :58.00 Median : 1518 Median :107.0   
## Mean : 52.8 Mean :59.36 Mean : 1779 Mean :105.7   
## 3rd Qu.: 62.0 3rd Qu.:67.00 3rd Qu.: 1682 3rd Qu.:150.0   
## Max. :201.0 Max. :95.00 Max. :30132 Max. :343.0   
## NA's :772 NA's :2085   
## TEAM\_PITCHING\_BB TEAM\_PITCHING\_SO TEAM\_FIELDING\_E TEAM\_FIELDING\_DP  
## Min. : 0.0 Min. : 0.0 Min. : 65.0 Min. : 52.0   
## 1st Qu.: 476.0 1st Qu.: 615.0 1st Qu.: 127.0 1st Qu.:131.0   
## Median : 536.5 Median : 813.5 Median : 159.0 Median :149.0   
## Mean : 553.0 Mean : 817.7 Mean : 246.5 Mean :146.4   
## 3rd Qu.: 611.0 3rd Qu.: 968.0 3rd Qu.: 249.2 3rd Qu.:164.0   
## Max. :3645.0 Max. :19278.0 Max. :1898.0 Max. :228.0   
## NA's :102 NA's :286

describe(moneyball[,-1])

## vars n mean sd median trimmed mad min max  
## TARGET\_WINS 1 2276 80.79 15.75 82.0 81.31 14.83 0 146  
## TEAM\_BATTING\_H 2 2276 1469.27 144.59 1454.0 1459.04 114.16 891 2554  
## TEAM\_BATTING\_2B 3 2276 241.25 46.80 238.0 240.40 47.44 69 458  
## TEAM\_BATTING\_3B 4 2276 55.25 27.94 47.0 52.18 23.72 0 223  
## TEAM\_BATTING\_HR 5 2276 99.61 60.55 102.0 97.39 78.58 0 264  
## TEAM\_BATTING\_BB 6 2276 501.56 122.67 512.0 512.18 94.89 0 878  
## TEAM\_BATTING\_SO 7 2174 735.61 248.53 750.0 742.31 284.66 0 1399  
## TEAM\_BASERUN\_SB 8 2145 124.76 87.79 101.0 110.81 60.79 0 697  
## TEAM\_BASERUN\_CS 9 1504 52.80 22.96 49.0 50.36 17.79 0 201  
## TEAM\_BATTING\_HBP 10 191 59.36 12.97 58.0 58.86 11.86 29 95  
## TEAM\_PITCHING\_H 11 2276 1779.21 1406.84 1518.0 1555.90 174.95 1137 30132  
## TEAM\_PITCHING\_HR 12 2276 105.70 61.30 107.0 103.16 74.13 0 343  
## TEAM\_PITCHING\_BB 13 2276 553.01 166.36 536.5 542.62 98.59 0 3645  
## TEAM\_PITCHING\_SO 14 2174 817.73 553.09 813.5 796.93 257.23 0 19278  
## TEAM\_FIELDING\_E 15 2276 246.48 227.77 159.0 193.44 62.27 65 1898  
## TEAM\_FIELDING\_DP 16 1990 146.39 26.23 149.0 147.58 23.72 52 228  
## range skew kurtosis se  
## TARGET\_WINS 146 -0.40 1.03 0.33  
## TEAM\_BATTING\_H 1663 1.57 7.28 3.03  
## TEAM\_BATTING\_2B 389 0.22 0.01 0.98  
## TEAM\_BATTING\_3B 223 1.11 1.50 0.59  
## TEAM\_BATTING\_HR 264 0.19 -0.96 1.27  
## TEAM\_BATTING\_BB 878 -1.03 2.18 2.57  
## TEAM\_BATTING\_SO 1399 -0.30 -0.32 5.33  
## TEAM\_BASERUN\_SB 697 1.97 5.49 1.90  
## TEAM\_BASERUN\_CS 201 1.98 7.62 0.59  
## TEAM\_BATTING\_HBP 66 0.32 -0.11 0.94  
## TEAM\_PITCHING\_H 28995 10.33 141.84 29.49  
## TEAM\_PITCHING\_HR 343 0.29 -0.60 1.28  
## TEAM\_PITCHING\_BB 3645 6.74 96.97 3.49  
## TEAM\_PITCHING\_SO 19278 22.17 671.19 11.86  
## TEAM\_FIELDING\_E 1833 2.99 10.97 4.77  
## TEAM\_FIELDING\_DP 176 -0.39 0.18 0.59

All the variables are numeric There are missing values with variables TEAM\_BATTING\_SO, TEAM\_BASERUN\_SB, TEAM\_BASERUN\_CS, TEAM\_BATTING\_HBP, TEAM\_PITCHING\_SO, TEAM\_FIELDING\_DP.

In This train dataset, the target variable, TARGET\_WINS, varies from 0 to 146.

The median and the mean are closed in values or in the same magnitude except TEAM\_PITCHING\_H where the mean is 200 time bigger than the median, TEAM\_FIELDING\_E where mean is also larger than median.

str(moneyball)

## '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 ...

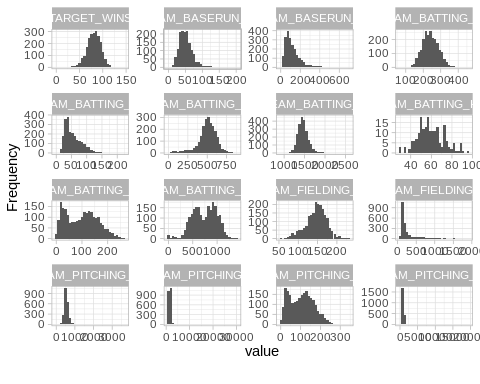
The str function explain the structure of data frame. The data frame has 15 variables of type integer with 2276 observations

dim(moneyball)

## [1] 2276 17

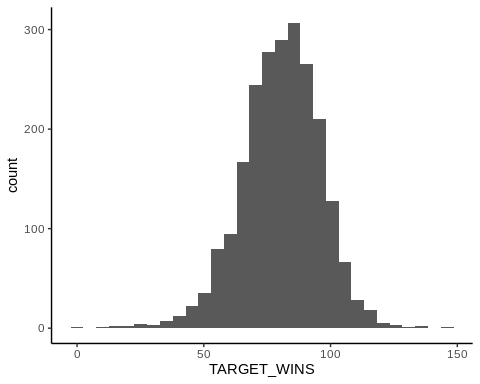
The histograms below allow the visualization of the distribution of each variable.

plot\_histogram(moneyball[,-1], ggtheme=theme\_light())

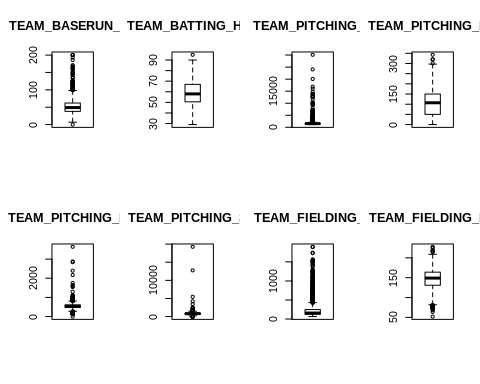
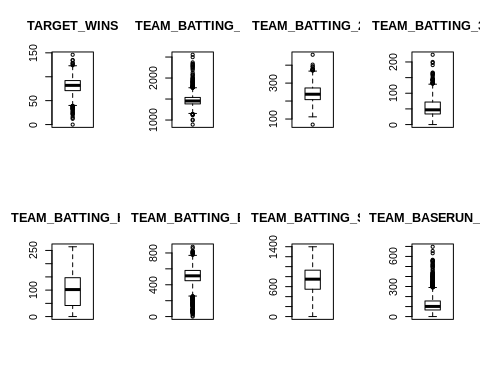
 The TARGET\_WINS which is the target variable present a normal distribution.The variables TEAM\_PITCHING\_H, TEAM\_PITCHING\_BB, and TEAM\_PITCHING\_SO have high degrees of skweness and kurtosis. These variable need to be log transformed before introducting in a model.

There are three bimodal distributions TEAM\_BATTING\_HR, TEAM\_BATING\_SO, AND REAM\_PITCHING\_HR

ggplot(moneyball, aes(TARGET\_WINS)) + geom\_histogram(bins=30) + theme\_classic()

 Normal distribution of the target variable.

par(mfrow = c(2,4))  
for (x in colnames(moneyball)[-1]){  
 boxplot(moneyball[,x],main=x)  
}

 The boxplots of different variables add some visual information about the outliers. Some variable distributions are skewed by to much outlierS in one side as TEAM\_FIELDING\_E, TEAM\_PITCHING\_H, TEAM\_BASERUN\_CS, and TEAM\_BATING\_HR.

library(e1071)  
apply(moneyball[,-1], 2, skewness)

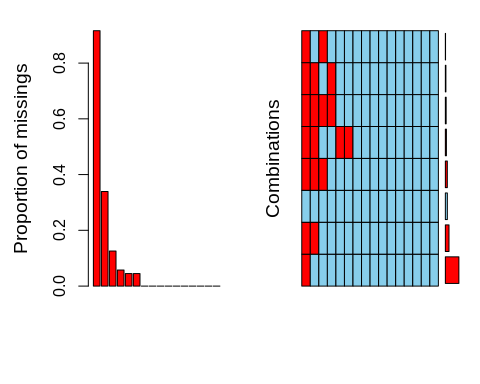
## TARGET\_WINS TEAM\_BATTING\_H TEAM\_BATTING\_2B TEAM\_BATTING\_3B   
## -0.3987232 1.5713335 0.2151018 1.1094652   
## TEAM\_BATTING\_HR TEAM\_BATTING\_BB TEAM\_BATTING\_SO TEAM\_BASERUN\_SB   
## 0.1860421 -1.0257599 NA NA   
## TEAM\_BASERUN\_CS TEAM\_BATTING\_HBP TEAM\_PITCHING\_H TEAM\_PITCHING\_HR   
## NA NA 10.3295111 0.2877877   
## TEAM\_PITCHING\_BB TEAM\_PITCHING\_SO TEAM\_FIELDING\_E TEAM\_FIELDING\_DP   
## 6.7438995 NA 2.9904656 NA

#### 

The aggr function in the VIM package plots and calculates the amount of missing values in each variable. The dply function is useful for wrangling data into aggregate summaries and is used to find the pattern of missing data related to the classes.

aggr(moneyball[,-1], prop = c(TRUE, TRUE), bars=TRUE, numbers=TRUE, sortVars=TRUE)

## Warning in plot.aggr(res, ...): not enough horizontal space to display  
## frequencies

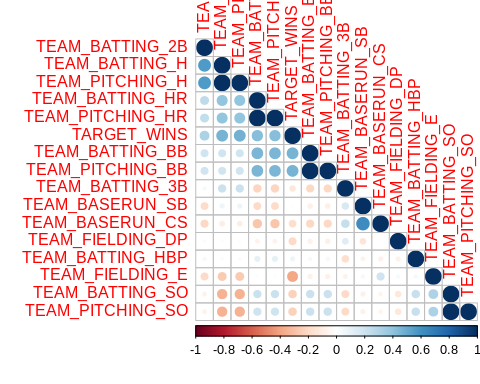


##   
## Variables sorted by number of missings:   
## Variable Count  
## TEAM\_BATTING\_HBP 0.91608084  
## TEAM\_BASERUN\_CS 0.33919156  
## TEAM\_FIELDING\_DP 0.12565905  
## TEAM\_BASERUN\_SB 0.05755712  
## TEAM\_BATTING\_SO 0.04481547  
## TEAM\_PITCHING\_SO 0.04481547  
## TARGET\_WINS 0.00000000  
## TEAM\_BATTING\_H 0.00000000  
## TEAM\_BATTING\_2B 0.00000000  
## TEAM\_BATTING\_3B 0.00000000  
## TEAM\_BATTING\_HR 0.00000000  
## TEAM\_BATTING\_BB 0.00000000  
## TEAM\_PITCHING\_H 0.00000000  
## TEAM\_PITCHING\_HR 0.00000000  
## TEAM\_PITCHING\_BB 0.00000000  
## TEAM\_FIELDING\_E 0.00000000

TEAM\_BATTING\_HBP and TEAM\_BASERUN\_CS have respectively 91.6% and 34% fo missing values in their respective column. Including those variable in the model imply an imputation of massive data in the model. We will exclude those variables from the model.

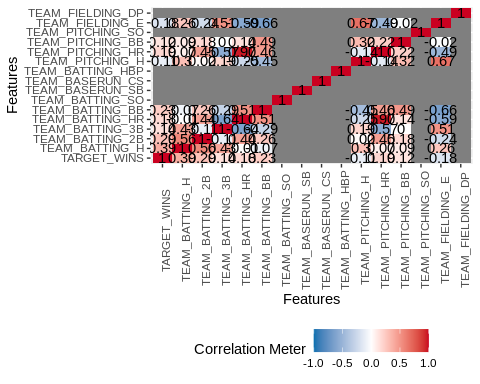
The correlations between variables in our training dataset are below.

cor\_moneyball <- cor(moneyball[,-1], use = "na.or.complete")  
corrplot(cor\_moneyball, order = 'hclust', type = 'lower')



plot\_correlation(moneyball[,-1])

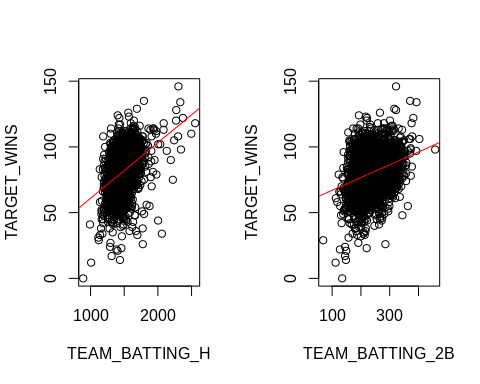
## Warning: Removed 150 rows containing missing values (geom\_text).

 There is no strong correlation between the target variable with other predictors.

### Divers Correlations with TARGET\_WINS

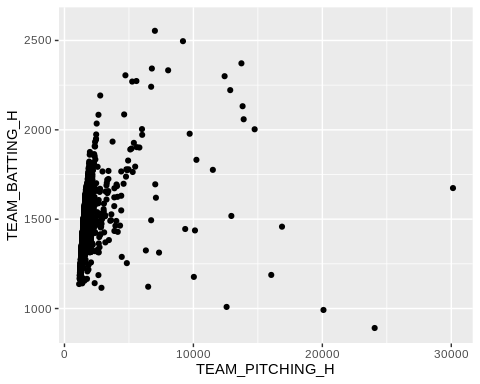
plot\_cor <- function(x, ...){  
 plot(TARGET\_WINS~x, moneyball, ...)  
 abline(lm(TARGET\_WINS~x, moneyball), col="red")  
}

par(mfrow=c(1,2))  
attach(moneyball)  
plot\_cor(TEAM\_BATTING\_H, xlab="TEAM\_BATTING\_H")  
plot\_cor(TEAM\_BATTING\_2B, xlab="TEAM\_BATTING\_2B")



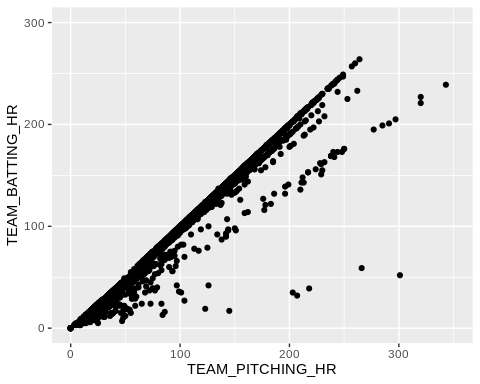
### High correlated predictors

ggplot(moneyball, aes(x=TEAM\_PITCHING\_H, y=TEAM\_BATTING\_H)) +   
 geom\_point() + coord\_cartesian(xlim=c(1000,30200), ylim=c(890, 2600))

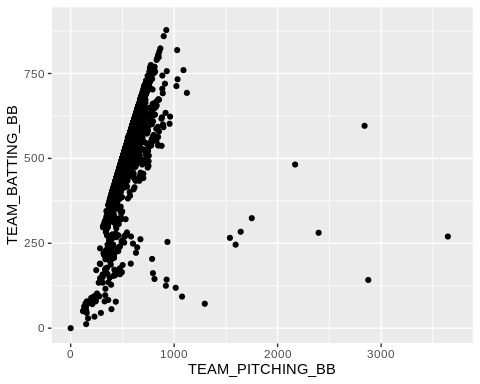


There exist a trend in the relationship between TEAM\_BATTING\_H and TEAM\_PITCHING\_H

ggplot(moneyball, aes(x=TEAM\_PITCHING\_HR, y=TEAM\_BATTING\_HR)) +   
 geom\_point() + coord\_cartesian(xlim=c(0,350), ylim=c(0, 300))

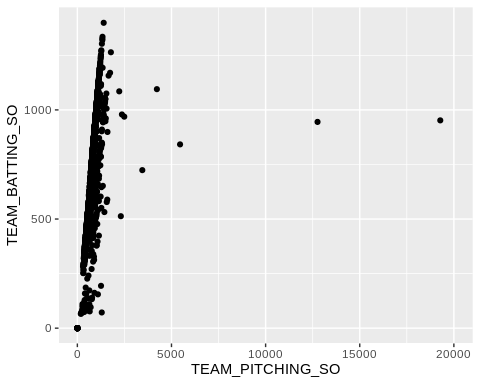
 The relation between TEAM\_BATTING\_HR and TEAM\_PITCHING\_H is strong enough even though there are multiple layers of linearities.

ggplot(moneyball, aes(x=TEAM\_PITCHING\_BB, y=TEAM\_BATTING\_BB)) +   
 geom\_point() + coord\_cartesian(xlim=c(0,3700), ylim=c(0, 900))

 TEAM\_BATTING\_BB and TEAM\_PITCHING\_BB could be collinear if we remove some outliers that leverage the relationship.

ggplot(moneyball, aes(x=TEAM\_PITCHING\_SO, y=TEAM\_BATTING\_SO)) +   
 geom\_point() + coord\_cartesian(xlim=c(0,20000), ylim=c(0, 1400))

## Warning: Removed 102 rows containing missing values (geom\_point).



TEAM\_BATTING\_SO AND TEAM\_PITCHING\_SO are colinear at some levels.

#moneyball.predictor <- moneyball[,-c(1,2)]  
#ggpairs(as.data.frame(moneyball.predictor), showStrips = FALSE)

The table above and the histograms show that the predictors hail, sever, seed.tmt, and lodging have around 18% of missing data. Other variables that are more likely to be missing are germ(16% of missing values), leaf.mild(16%),fruiting.bodies(15%), fruits.spots(15%), seed.discolor(15%), and shriveling(15%).

The grid shows the combination of all with 82% of data not missing in accordance with the problem description (18% missing). The remainder of the grid shows missing data for variable combinations with each row highlighting the missing values for the group of variables detailed in the x-axis. The non-graphical output of the function shows on top the exact proportion of missing values per variable.

## DATA PREPARATION

### Remove the two variables with lot of missing data

moneyball\_train <- moneyball[, -c(1,2,10,11)]

# MoneyBall <- moneyball[,-c(1,10,11)]  
# MoneyBall <-mice(MoneyBall, method="pmm", printFlag=FALSE, seed=6)   
#   
# aggr(complete(MoneyBall), prop = c(TRUE, TRUE), bars=TRUE, numbers=TRUE, sortVars=TRUE)

# moneyball\_train2 <- MoneyBall[[1]]  
# describe(moneyball\_train2)

# moneyball\_train2 <- cbind(TARGET\_WINS=moneyball[,"TARGET\_WINS"],moneyball\_train2)  
# moneyball\_train2 <- moneyball\_train2[1:n,]  
# moneyball\_test2 <- moneyball\_train2[n:m,]  
#   
# head(moneyball\_train2)

### Imputing the median in place of missing data

library(dplyr)  
moneyball\_train <- moneyball %>%   
 mutate\_all(~ifelse(is.na(.), median(., na.rm = TRUE), .))  
colSums(is.na(moneyball\_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 0   
## TEAM\_BASERUN\_SB TEAM\_BASERUN\_CS TEAM\_BATTING\_HBP TEAM\_PITCHING\_H   
## 0 0 0 0   
## TEAM\_PITCHING\_HR TEAM\_PITCHING\_BB TEAM\_PITCHING\_SO TEAM\_FIELDING\_E   
## 0 0 0 0   
## TEAM\_FIELDING\_DP   
## 0

#### Splitting into train test dataset

nrow <- dim(moneyball\_train)[1]  
nsplit <- as.integer(nrow\*.8)  
moneyball\_train1 <- moneyball\_train[1:nsplit,-c(1,10,11)]  
moneyball\_test1 <- moneyball\_train[nsplit:nrow,-c(10,11)]

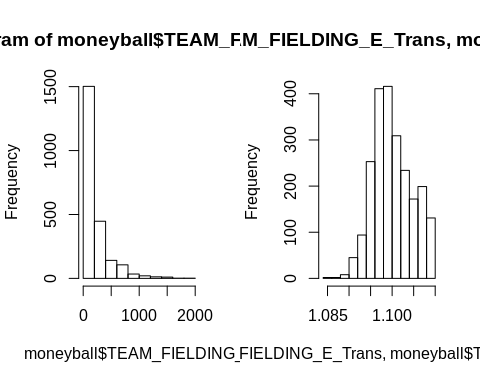
### Transforming the skewedvariables

#### Look for lambda transformation

library(caret)  
TEAM\_FIELDING\_E\_Trans <- BoxCoxTrans(moneyball$TEAM\_FIELDING\_E)  
TEAM\_FIELDING\_E\_Trans

## Box-Cox Transformation  
##   
## 2276 data points used to estimate Lambda  
##   
## Input data summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 65.0 127.0 159.0 246.5 249.2 1898.0   
##   
## Largest/Smallest: 29.2   
## Sample Skewness: 2.99   
##   
## Estimated Lambda: -0.9

#The original data  
par(mfrow=c(1,2))  
hist(moneyball$TEAM\_FIELDING\_E)  
# After transformation  
hist(predict(TEAM\_FIELDING\_E\_Trans, moneyball$TEAM\_FIELDING\_E))

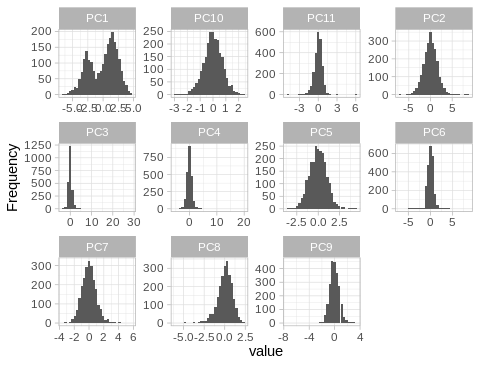


trans <- preProcess(moneyball\_train,method = c("BoxCox", "center", "scale", "pca"))  
trans

## Created from 2276 samples and 17 variables  
##   
## Pre-processing:  
## - Box-Cox transformation (7)  
## - centered (17)  
## - ignored (0)  
## - principal component signal extraction (17)  
## - scaled (17)  
##   
## Lambda estimates for Box-Cox transformation:  
## 0.7, -1.3, 0.6, 0.4, -2, -0.9, 1.8  
## PCA needed 11 components to capture 95 percent of the variance

# Apply the transformations:  
transformed <- predict(trans, moneyball\_train)

plot\_histogram(transformed, ggtheme=theme\_light())



baseball\_train <-cbind(moneyball[,2], transformed)

?merge

## Help on topic 'merge' was found in the following packages:  
##   
## Package Library  
## base /usr/lib/R/library  
## lava /home/alain/R/x86\_64-pc-linux-gnu-library/3.6  
## sp /home/alain/R/x86\_64-pc-linux-gnu-library/3.6  
## data.table /home/alain/R/x86\_64-pc-linux-gnu-library/3.6  
##   
##   
## Using the first match ...

head(baseball\_train)

## moneyball[, 2] PC1 PC2 PC3 PC4 PC5  
## 1 39 -1.6077922 -1.1719114 5.64426824 7.4100674 0.09615263  
## 2 70 3.0691749 -1.0499726 0.53511064 0.5932772 -1.90541052  
## 3 86 1.7792841 -0.5512956 -0.07011525 0.2485807 -1.57886869  
## 4 70 0.7338135 -1.5622378 -0.85211911 0.9274248 -1.40218521  
## 5 82 1.3572663 -2.1587298 -0.61502486 0.1515135 -1.65495497  
## 6 75 0.9350596 -2.5850492 -0.25872530 -0.5360716 -1.00937426  
## PC6 PC7 PC8 PC9 PC10 PC11  
## 1 -0.2150172 3.1562756 -2.80296573 0.08221636 -1.58775439 0.3542679  
## 2 -1.4802892 0.9318745 0.63595926 0.61700252 0.15727630 -0.6500371  
## 3 -1.5202266 1.2307636 0.78512869 -0.15085783 -0.08859832 0.1711517  
## 4 -1.2495558 1.2001744 0.18846020 0.22071705 -0.40866418 0.4791926  
## 5 -1.2487793 1.0114017 0.09829747 -0.54080924 -1.15211830 0.4009828  
## 6 -0.9297980 1.4684596 -0.68469292 0.06281676 -0.74328911 0.4627860

transformed[1:6,1:5]

## PC1 PC2 PC3 PC4 PC5  
## 1 -1.6077922 -1.1719114 5.64426824 7.4100674 0.09615263  
## 2 3.0691749 -1.0499726 0.53511064 0.5932772 -1.90541052  
## 3 1.7792841 -0.5512956 -0.07011525 0.2485807 -1.57886869  
## 4 0.7338135 -1.5622378 -0.85211911 0.9274248 -1.40218521  
## 5 1.3572663 -2.1587298 -0.61502486 0.1515135 -1.65495497  
## 6 0.9350596 -2.5850492 -0.25872530 -0.5360716 -1.00937426

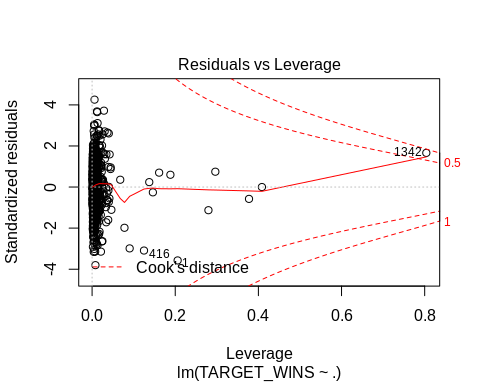
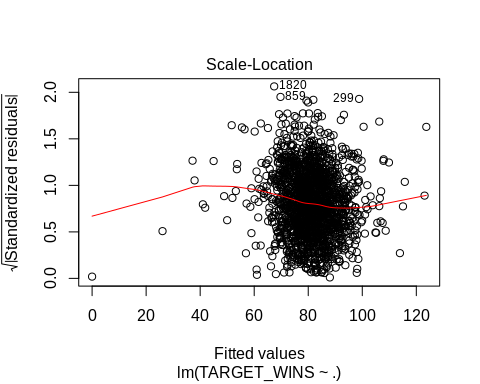
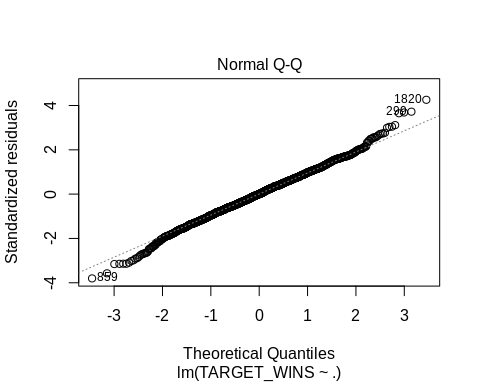
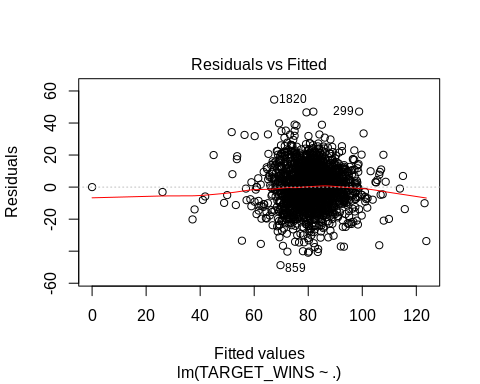
#colSums(is.na(moneyballp))

## BUILD MODELS

lm01 <- lm(TARGET\_WINS~., moneyball\_train1)  
summary(lm01)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ ., data = moneyball\_train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -48.734 -8.124 0.001 8.288 54.604   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 26.4877901 5.8872286 4.499 7.26e-06 \*\*\*  
## TEAM\_BATTING\_H 0.0448682 0.0040650 11.038 < 2e-16 \*\*\*  
## TEAM\_BATTING\_2B -0.0307027 0.0101105 -3.037 0.00243 \*\*   
## TEAM\_BATTING\_3B 0.0968992 0.0181473 5.340 1.05e-07 \*\*\*  
## TEAM\_BATTING\_HR 0.0467460 0.0313119 1.493 0.13563   
## TEAM\_BATTING\_BB 0.0190663 0.0069012 2.763 0.00579 \*\*   
## TEAM\_BATTING\_SO -0.0136139 0.0032407 -4.201 2.79e-05 \*\*\*  
## TEAM\_BASERUN\_SB 0.0278951 0.0048189 5.789 8.35e-09 \*\*\*  
## TEAM\_PITCHING\_H -0.0002383 0.0004089 -0.583 0.56017   
## TEAM\_PITCHING\_HR 0.0396934 0.0277145 1.432 0.15225   
## TEAM\_PITCHING\_BB -0.0058496 0.0050952 -1.148 0.25110   
## TEAM\_PITCHING\_SO 0.0072298 0.0016350 4.422 1.04e-05 \*\*\*  
## TEAM\_FIELDING\_E -0.0200219 0.0026864 -7.453 1.41e-13 \*\*\*  
## TEAM\_FIELDING\_DP -0.1258433 0.0140361 -8.966 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.87 on 1806 degrees of freedom  
## Multiple R-squared: 0.3195, Adjusted R-squared: 0.3146   
## F-statistic: 65.22 on 13 and 1806 DF, p-value: < 2.2e-16

plot(lm01)



#### We remove the predictor with the highest p-value

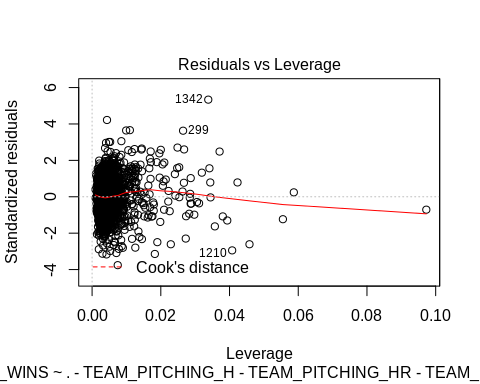
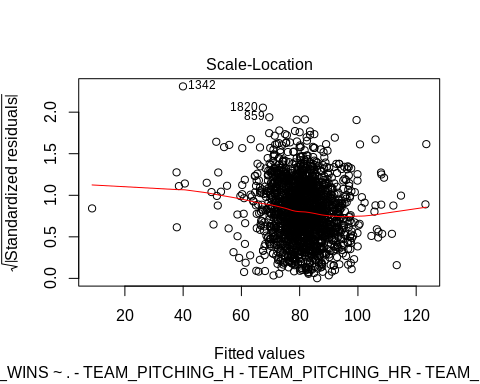
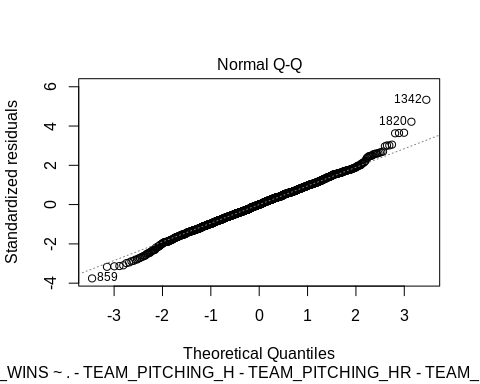
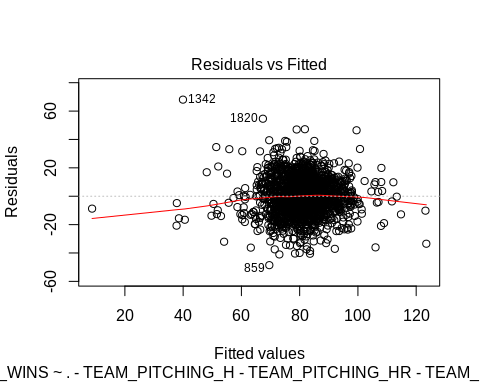
lm02 <- lm(TARGET\_WINS~.-TEAM\_BATTING\_SO, moneyball\_train1)  
summary(lm02)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ . - TEAM\_BATTING\_SO, data = moneyball\_train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -46.680 -8.466 -0.020 8.395 52.873   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 12.4297738 4.8658784 2.554 0.010716 \*   
## TEAM\_BATTING\_H 0.0511463 0.0037976 13.468 < 2e-16 \*\*\*  
## TEAM\_BATTING\_2B -0.0360255 0.0100769 -3.575 0.000359 \*\*\*  
## TEAM\_BATTING\_3B 0.1036119 0.0181599 5.706 1.35e-08 \*\*\*  
## TEAM\_BATTING\_HR 0.0400851 0.0314154 1.276 0.202131   
## TEAM\_BATTING\_BB 0.0116223 0.0067004 1.735 0.082989 .   
## TEAM\_BASERUN\_SB 0.0215647 0.0045982 4.690 2.94e-06 \*\*\*  
## TEAM\_PITCHING\_H -0.0003091 0.0004104 -0.753 0.451437   
## TEAM\_PITCHING\_HR 0.0216824 0.0275067 0.788 0.430648   
## TEAM\_PITCHING\_BB 0.0031008 0.0046497 0.667 0.504935   
## TEAM\_PITCHING\_SO 0.0030405 0.0013017 2.336 0.019608 \*   
## TEAM\_FIELDING\_E -0.0187112 0.0026805 -6.981 4.12e-12 \*\*\*  
## TEAM\_FIELDING\_DP -0.1186085 0.0139941 -8.476 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.93 on 1807 degrees of freedom  
## Multiple R-squared: 0.3128, Adjusted R-squared: 0.3083   
## F-statistic: 68.55 on 12 and 1807 DF, p-value: < 2.2e-16

lm11 <- lm(TARGET\_WINS~.-TEAM\_PITCHING\_H-TEAM\_PITCHING\_HR-TEAM\_PITCHING\_BB-TEAM\_PITCHING\_SO, moneyball\_train1)  
summary(lm11)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ . - TEAM\_PITCHING\_H - TEAM\_PITCHING\_HR -   
## TEAM\_PITCHING\_BB - TEAM\_PITCHING\_SO, data = moneyball\_train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -48.583 -8.284 -0.017 8.285 68.063   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 25.415618 5.870164 4.330 1.58e-05 \*\*\*  
## TEAM\_BATTING\_H 0.043405 0.004022 10.793 < 2e-16 \*\*\*  
## TEAM\_BATTING\_2B -0.023340 0.010099 -2.311 0.02094 \*   
## TEAM\_BATTING\_3B 0.100211 0.017733 5.651 1.85e-08 \*\*\*  
## TEAM\_BATTING\_HR 0.080643 0.010851 7.432 1.64e-13 \*\*\*  
## TEAM\_BATTING\_BB 0.012287 0.003774 3.255 0.00115 \*\*   
## TEAM\_BATTING\_SO -0.004257 0.002580 -1.650 0.09909 .   
## TEAM\_BASERUN\_SB 0.026545 0.004736 5.605 2.40e-08 \*\*\*  
## TEAM\_FIELDING\_E -0.017943 0.002203 -8.145 7.00e-16 \*\*\*  
## TEAM\_FIELDING\_DP -0.122708 0.014141 -8.678 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.98 on 1810 degrees of freedom  
## Multiple R-squared: 0.3062, Adjusted R-squared: 0.3028   
## F-statistic: 88.77 on 9 and 1810 DF, p-value: < 2.2e-16

plot(lm11)



lm2 <- lm(TARGET\_WINS~TEAM\_BATTING\_2B+TEAM\_BATTING\_H+TEAM\_PITCHING\_H+  
 TEAM\_BATTING\_HR+TEAM\_PITCHING\_HR+  
 TEAM\_PITCHING\_BB+TEAM\_FIELDING\_E , moneyball)  
summary(lm2)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ TEAM\_BATTING\_2B + TEAM\_BATTING\_H +   
## TEAM\_PITCHING\_H + TEAM\_BATTING\_HR + TEAM\_PITCHING\_HR + TEAM\_PITCHING\_BB +   
## TEAM\_FIELDING\_E, data = moneyball)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -53.298 -8.868 0.110 8.799 51.667   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.3576809 3.3445960 0.406 0.684830   
## TEAM\_BATTING\_2B -0.0334941 0.0090405 -3.705 0.000217 \*\*\*  
## TEAM\_BATTING\_H 0.0585216 0.0028059 20.856 < 2e-16 \*\*\*  
## TEAM\_PITCHING\_H -0.0018772 0.0003147 -5.965 2.83e-09 \*\*\*  
## TEAM\_BATTING\_HR 0.0164810 0.0240922 0.684 0.493995   
## TEAM\_PITCHING\_HR -0.0047054 0.0226647 -0.208 0.835554   
## TEAM\_PITCHING\_BB 0.0128687 0.0020225 6.363 2.39e-10 \*\*\*  
## TEAM\_FIELDING\_E -0.0137590 0.0022656 -6.073 1.47e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 13.59 on 2268 degrees of freedom  
## Multiple R-squared: 0.2582, Adjusted R-squared: 0.2559   
## F-statistic: 112.8 on 7 and 2268 DF, p-value: < 2.2e-16

lm3 <- lm(TARGET\_WINS~TEAM\_BATTING\_2B+TEAM\_BATTING\_H+  
 TEAM\_BATTING\_HR+TEAM\_BATTING\_SO+  
 TEAM\_BATTING\_BB, moneyball)  
summary(lm3)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ TEAM\_BATTING\_2B + TEAM\_BATTING\_H +   
## TEAM\_BATTING\_HR + TEAM\_BATTING\_SO + TEAM\_BATTING\_BB, data = moneyball)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -59.904 -8.595 0.573 8.982 53.284   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -9.937644 5.012318 -1.983 0.04753 \*   
## TEAM\_BATTING\_2B -0.022361 0.009297 -2.405 0.01625 \*   
## TEAM\_BATTING\_H 0.052027 0.003378 15.404 < 2e-16 \*\*\*  
## TEAM\_BATTING\_HR 0.025820 0.008622 2.995 0.00278 \*\*   
## TEAM\_BATTING\_SO 0.002612 0.002260 1.156 0.24777   
## TEAM\_BATTING\_BB 0.029388 0.002772 10.601 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 13.58 on 2168 degrees of freedom  
## (102 observations deleted due to missingness)  
## Multiple R-squared: 0.2419, Adjusted R-squared: 0.2401   
## F-statistic: 138.4 on 5 and 2168 DF, p-value: < 2.2e-16

lm3 <- lm(TARGET\_WINS~TEAM\_BATTING\_2B+TEAM\_PITCHING\_H+TEAM\_PITCHING\_HR+TEAM\_PITCHING\_SO+TEAM\_PITCHING\_BB, moneyball)  
summary(lm3)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ TEAM\_BATTING\_2B + TEAM\_PITCHING\_H +   
## TEAM\_PITCHING\_HR + TEAM\_PITCHING\_SO + TEAM\_PITCHING\_BB, data = moneyball)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -62.118 -9.519 0.245 9.378 67.184   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 54.6413858 1.8825343 29.025 < 2e-16 \*\*\*  
## TEAM\_BATTING\_2B 0.0827264 0.0075317 10.984 < 2e-16 \*\*\*  
## TEAM\_PITCHING\_H -0.0012829 0.0002392 -5.363 9.05e-08 \*\*\*  
## TEAM\_PITCHING\_HR 0.0219313 0.0060492 3.625 0.000295 \*\*\*  
## TEAM\_PITCHING\_SO -0.0048607 0.0006606 -7.357 2.65e-13 \*\*\*  
## TEAM\_PITCHING\_BB 0.0176132 0.0022130 7.959 2.77e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.47 on 2168 degrees of freedom  
## (102 observations deleted due to missingness)  
## Multiple R-squared: 0.1385, Adjusted R-squared: 0.1365   
## F-statistic: 69.71 on 5 and 2168 DF, p-value: < 2.2e-16

lm2 <- lm(TARGET\_WINS~TEAM\_BATTING\_2B+TEAM\_BATTING\_H+TEAM\_PITCHING\_H+  
 TEAM\_BATTING\_HR+TEAM\_PITCHING\_HR+TEAM\_BATTING\_SO+TEAM\_PITCHING\_SO+  
 TEAM\_BATTING\_BB+TEAM\_PITCHING\_BB, moneyball)  
summary(lm2)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ TEAM\_BATTING\_2B + TEAM\_BATTING\_H +   
## TEAM\_PITCHING\_H + TEAM\_BATTING\_HR + TEAM\_PITCHING\_HR + TEAM\_BATTING\_SO +   
## TEAM\_PITCHING\_SO + TEAM\_BATTING\_BB + TEAM\_PITCHING\_BB, data = moneyball)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -51.641 -8.660 0.346 9.026 49.760   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4.7512713 5.0667223 -0.938 0.34848   
## TEAM\_BATTING\_2B -0.0259231 0.0092470 -2.803 0.00510 \*\*   
## TEAM\_BATTING\_H 0.0562882 0.0034454 16.337 < 2e-16 \*\*\*  
## TEAM\_PITCHING\_H -0.0026602 0.0003322 -8.007 1.9e-15 \*\*\*  
## TEAM\_BATTING\_HR 0.0329398 0.0270819 1.216 0.22400   
## TEAM\_PITCHING\_HR 0.0065137 0.0246717 0.264 0.79179   
## TEAM\_BATTING\_SO -0.0041868 0.0025209 -1.661 0.09689 .   
## TEAM\_PITCHING\_SO 0.0027962 0.0009324 2.999 0.00274 \*\*   
## TEAM\_BATTING\_BB 0.0149152 0.0057096 2.612 0.00906 \*\*   
## TEAM\_PITCHING\_BB 0.0049287 0.0041793 1.179 0.23841   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 13.35 on 2164 degrees of freedom  
## (102 observations deleted due to missingness)  
## Multiple R-squared: 0.2685, Adjusted R-squared: 0.2655   
## F-statistic: 88.26 on 9 and 2164 DF, p-value: < 2.2e-16

### Tuning Linear Model

metric = 'RMSE'  
  
  
# Train control  
  
customTrainControl <- trainControl(method = "repeatedcv",   
 number = 10,   
 repeats = 5 ,  
 verboseIter = F)  
#Linear Model  
lmg <- train(TARGET\_WINS ~ .,  
 moneyball\_train1,  
 method= 'lm',  
 trControl = customTrainControl  
 )

lmg$results

## intercept RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 TRUE 13.00564 0.3011865 10.20265 0.5283102 0.06376927 0.3811096

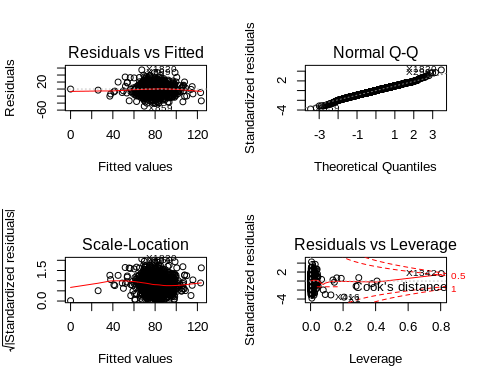
lmg # 2055 , 32 predictors,

## Linear Regression   
##   
## 1820 samples  
## 13 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 5 times)   
## Summary of sample sizes: 1638, 1637, 1639, 1638, 1638, 1639, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 13.00564 0.3011865 10.20265  
##   
## Tuning parameter 'intercept' was held constant at a value of TRUE

summary(lmg)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -48.734 -8.124 0.001 8.288 54.604   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 26.4877901 5.8872286 4.499 7.26e-06 \*\*\*  
## TEAM\_BATTING\_H 0.0448682 0.0040650 11.038 < 2e-16 \*\*\*  
## TEAM\_BATTING\_2B -0.0307027 0.0101105 -3.037 0.00243 \*\*   
## TEAM\_BATTING\_3B 0.0968992 0.0181473 5.340 1.05e-07 \*\*\*  
## TEAM\_BATTING\_HR 0.0467460 0.0313119 1.493 0.13563   
## TEAM\_BATTING\_BB 0.0190663 0.0069012 2.763 0.00579 \*\*   
## TEAM\_BATTING\_SO -0.0136139 0.0032407 -4.201 2.79e-05 \*\*\*  
## TEAM\_BASERUN\_SB 0.0278951 0.0048189 5.789 8.35e-09 \*\*\*  
## TEAM\_PITCHING\_H -0.0002383 0.0004089 -0.583 0.56017   
## TEAM\_PITCHING\_HR 0.0396934 0.0277145 1.432 0.15225   
## TEAM\_PITCHING\_BB -0.0058496 0.0050952 -1.148 0.25110   
## TEAM\_PITCHING\_SO 0.0072298 0.0016350 4.422 1.04e-05 \*\*\*  
## TEAM\_FIELDING\_E -0.0200219 0.0026864 -7.453 1.41e-13 \*\*\*  
## TEAM\_FIELDING\_DP -0.1258433 0.0140361 -8.966 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.87 on 1806 degrees of freedom  
## Multiple R-squared: 0.3195, Adjusted R-squared: 0.3146   
## F-statistic: 65.22 on 13 and 1806 DF, p-value: < 2.2e-16

par(mfrow=c(2,2))  
plot(lmg$finalModel)



### Model for pca

pcaModel <- lm(TARGET\_WINS~.-PC3-PC7,transformed)  
summary(pcaModel)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ . - PC3 - PC7, data = transformed)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -14.544 -2.710 -0.108 2.548 33.052   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 80.79086 0.08995 898.133 <2e-16 \*\*\*  
## PC1 0.61162 0.03975 15.388 <2e-16 \*\*\*  
## PC2 5.45299 0.05569 97.918 <2e-16 \*\*\*  
## PC4 -5.91745 0.07803 -75.838 <2e-16 \*\*\*  
## PC5 3.35846 0.08608 39.015 <2e-16 \*\*\*  
## PC6 -1.50375 0.09045 -16.626 <2e-16 \*\*\*  
## PC8 4.47131 0.09465 47.239 <2e-16 \*\*\*  
## PC9 -9.25670 0.11644 -79.496 <2e-16 \*\*\*  
## PC10 -5.97985 0.12250 -48.814 <2e-16 \*\*\*  
## PC11 1.21020 0.13840 8.744 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.291 on 2266 degrees of freedom  
## Multiple R-squared: 0.9261, Adjusted R-squared: 0.9258   
## F-statistic: 3154 on 9 and 2266 DF, p-value: < 2.2e-16

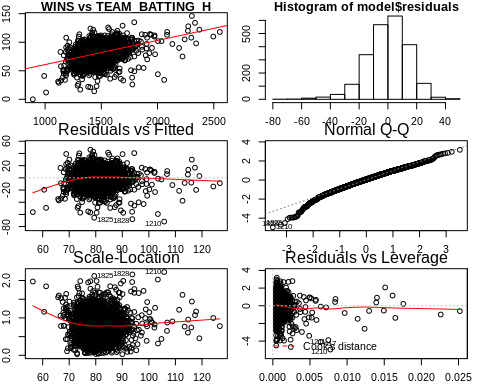
#names(summary(model))

### Foward Selection

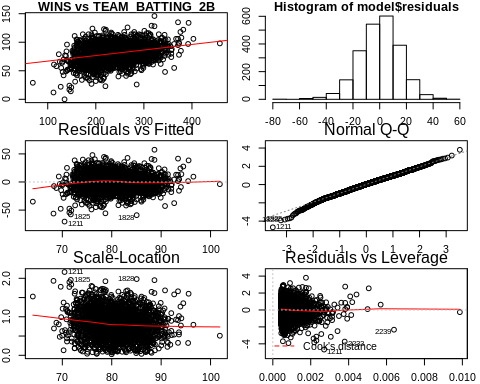
#### Impact of each predictor on the outcome

predictors\_validation <- function(x, ...){  
 par(mar=c(2,2,1,1), mfrow=c(3,2))  
 plot(TARGET\_WINS~x, moneyball, ...)  
 model <- lm(TARGET\_WINS~x, moneyball)  
 abline(model, col="red")  
 hist(model$residuals)  
 plot(model)  
}

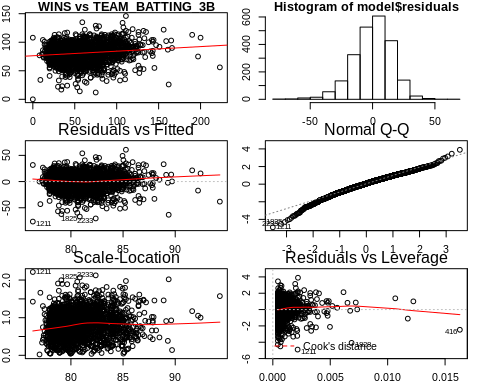
predictors\_validation(TEAM\_BATTING\_H, main ="WINS vs TEAM\_BATTING\_H")



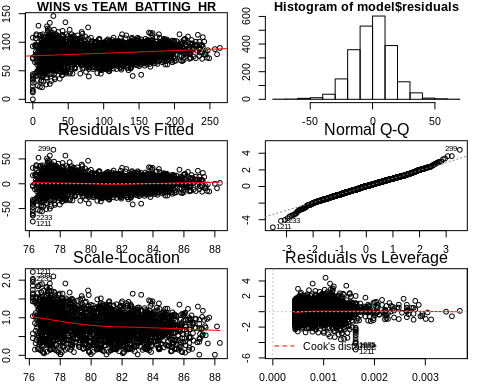
predictors\_validation(TEAM\_BATTING\_2B, main ="WINS vs TEAM\_BATTING\_2B")



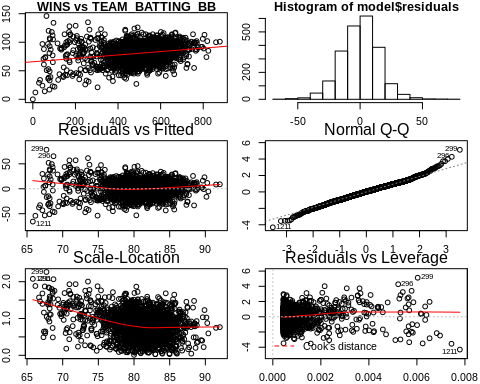
predictors\_validation(TEAM\_BATTING\_3B, main ="WINS vs TEAM\_BATTING\_3B")



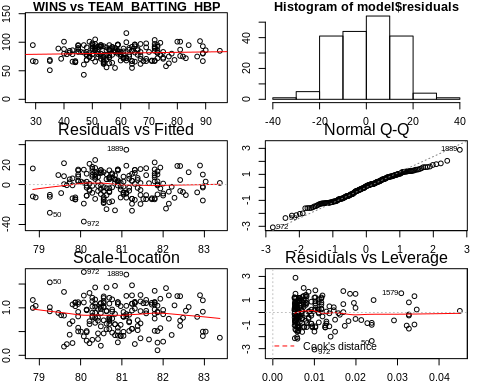
predictors\_validation(TEAM\_BATTING\_HR, main ="WINS vs TEAM\_BATTING\_HR")



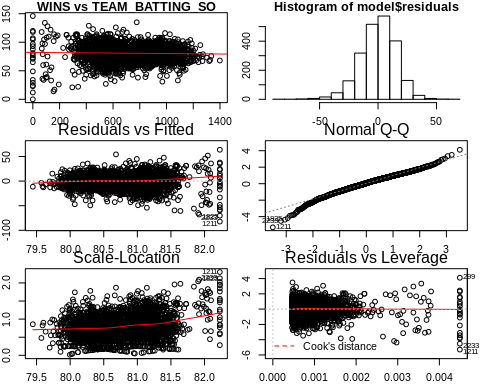
predictors\_validation(TEAM\_BATTING\_BB, main ="WINS vs TEAM\_BATTING\_BB")



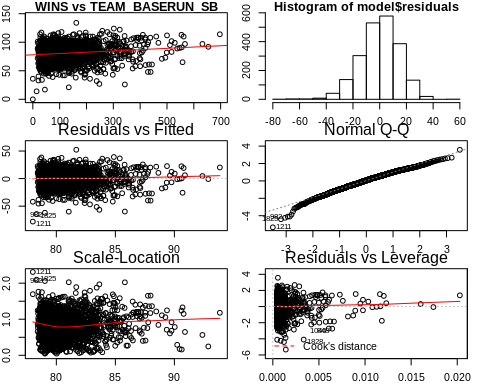
predictors\_validation(TEAM\_BATTING\_HBP, main ="WINS vs TEAM\_BATTING\_HBP")



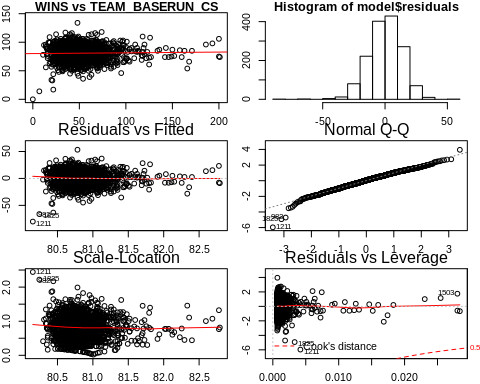
predictors\_validation(TEAM\_BATTING\_SO, main ="WINS vs TEAM\_BATTING\_SO")



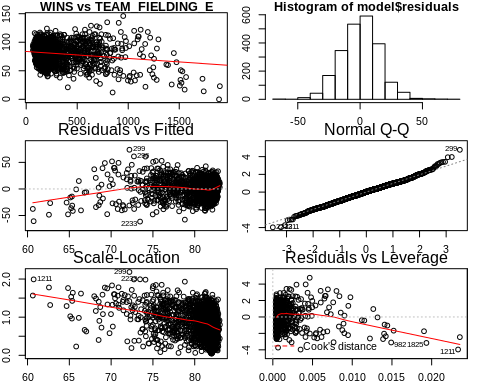
predictors\_validation(TEAM\_BASERUN\_SB, main ="WINS vs TEAM\_BASERUN\_SB")



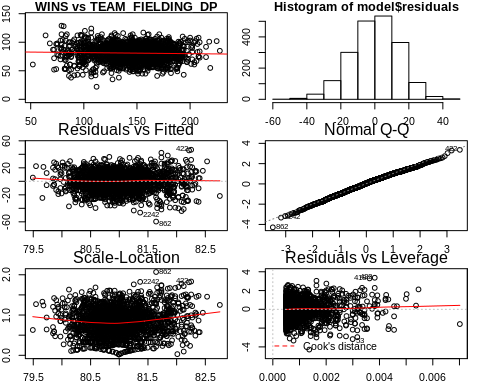
predictors\_validation(TEAM\_BASERUN\_CS, main ="WINS vs TEAM\_BASERUN\_CS")



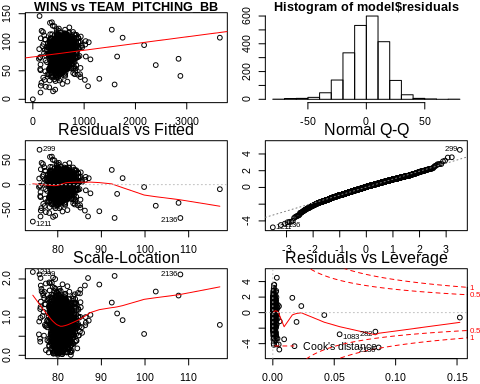
predictors\_validation(TEAM\_FIELDING\_E, main ="WINS vs TEAM\_FIELDING\_E")



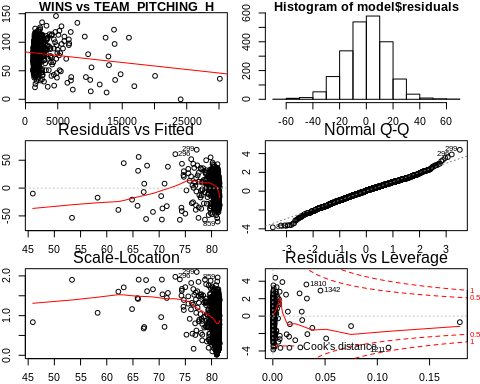
predictors\_validation(TEAM\_FIELDING\_DP, main ="WINS vs TEAM\_FIELDING\_DP")



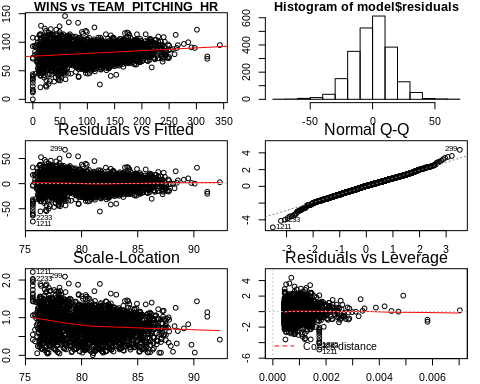
predictors\_validation(TEAM\_PITCHING\_BB, main ="WINS vs TEAM\_PITCHING\_BB")



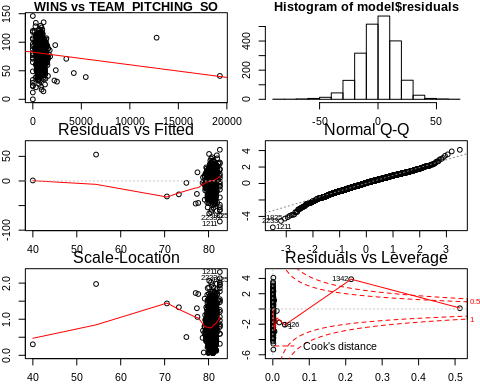
predictors\_validation(TEAM\_PITCHING\_H, main ="WINS vs TEAM\_PITCHING\_H")



predictors\_validation(TEAM\_PITCHING\_HR, main ="WINS vs TEAM\_PITCHING\_HR")



predictors\_validation(TEAM\_PITCHING\_SO, main ="WINS vs TEAM\_PITCHING\_SO")



adj.rsq <- NULL  
for (x in colnames(moneyball\_train)) {  
 model <- lm(TARGET\_WINS~moneyball\_train[,x], moneyball\_train)  
 adj.rsq <- c(adj.rsq, summary(model)$adj.r.squared)  
}  
names(adj.rsq) <- colnames(moneyball\_train)   
adj.rsq

## INDEX TARGET\_WINS TEAM\_BATTING\_H TEAM\_BATTING\_2B   
## 3.814687e-06 1.000000e+00 1.507669e-01 8.317792e-02   
## TEAM\_BATTING\_3B TEAM\_BATTING\_HR TEAM\_BATTING\_BB TEAM\_BATTING\_SO   
## 1.990635e-02 3.060384e-02 5.366812e-02 4.958767e-04   
## TEAM\_BASERUN\_SB TEAM\_BASERUN\_CS TEAM\_BATTING\_HBP TEAM\_PITCHING\_H   
## 1.484661e-02 -1.849260e-04 -1.668419e-04 1.165172e-02   
## TEAM\_PITCHING\_HR TEAM\_PITCHING\_BB TEAM\_PITCHING\_SO TEAM\_FIELDING\_E   
## 3.530215e-02 1.498634e-02 5.308364e-03 3.072081e-02   
## TEAM\_FIELDING\_DP   
## 4.658299e-04

#### Foward selection

We add one predictor at the time and observe the change in adjusted r\_squared. If the r\_squared increases, we keep the predictor, otherwise we remove that predictor.

foward.selection.model <- lm(TARGET\_WINS~TEAM\_BATTING\_H+TEAM\_BATTING\_2B+TEAM\_BATTING\_BB+TEAM\_FIELDING\_E+  
 TEAM\_BATTING\_HR+TEAM\_PITCHING\_HR+TEAM\_BATTING\_3B+TEAM\_BASERUN\_SB+  
 TEAM\_PITCHING\_H+TEAM\_BATTING\_SO+TEAM\_PITCHING\_BB+TEAM\_PITCHING\_SO+  
 TEAM\_FIELDING\_DP, moneyball\_train)

foward.selection.model <- lm(TARGET\_WINS~TEAM\_BATTING\_H+TEAM\_BATTING\_2B+TEAM\_BATTING\_BB+TEAM\_PITCHING\_HR+  
 TEAM\_FIELDING\_E+TEAM\_BATTING\_3B+TEAM\_BASERUN\_SB+  
 TEAM\_PITCHING\_H+TEAM\_PITCHING\_SO+TEAM\_FIELDING\_DP, moneyball\_train)

summary(foward.selection.model)$adj.r.squared

## [1] 0.307881

summary(foward.selection.model)

##   
## Call:  
## lm(formula = TARGET\_WINS ~ TEAM\_BATTING\_H + TEAM\_BATTING\_2B +   
## TEAM\_BATTING\_BB + TEAM\_PITCHING\_HR + TEAM\_FIELDING\_E + TEAM\_BATTING\_3B +   
## TEAM\_BASERUN\_SB + TEAM\_PITCHING\_H + TEAM\_PITCHING\_SO + TEAM\_FIELDING\_DP,   
## data = moneyball\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -51.016 -8.545 0.080 8.434 55.883   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.1840241 3.9184730 2.854 0.00435 \*\*   
## TEAM\_BATTING\_H 0.0540533 0.0033680 16.049 < 2e-16 \*\*\*  
## TEAM\_BATTING\_2B -0.0266846 0.0089866 -2.969 0.00302 \*\*   
## TEAM\_BATTING\_BB 0.0139419 0.0033243 4.194 2.85e-05 \*\*\*  
## TEAM\_PITCHING\_HR 0.0416183 0.0069515 5.987 2.48e-09 \*\*\*  
## TEAM\_FIELDING\_E -0.0187712 0.0023813 -7.883 4.93e-15 \*\*\*  
## TEAM\_BATTING\_3B 0.0663052 0.0159486 4.157 3.34e-05 \*\*\*  
## TEAM\_BASERUN\_SB 0.0206002 0.0040528 5.083 4.02e-07 \*\*\*  
## TEAM\_PITCHING\_H -0.0006610 0.0003127 -2.114 0.03462 \*   
## TEAM\_PITCHING\_SO 0.0020188 0.0006208 3.252 0.00116 \*\*   
## TEAM\_FIELDING\_DP -0.1146370 0.0128170 -8.944 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 13.1 on 2265 degrees of freedom  
## Multiple R-squared: 0.3109, Adjusted R-squared: 0.3079   
## F-statistic: 102.2 on 10 and 2265 DF, p-value: < 2.2e-16

## SELECT MODELS

# Train data Test set   
X\_test <- moneyball\_test1[,-1] # Dropped TARGET\_WINS  
Y\_test <- moneyball\_test1[,1] # Only TARGET\_WINS  
  
test\_model <- function(modelName,predData){  
options(warn=-1) #turn off warnings  
predicted\_result <- predict(modelName, predData)  
options(warn=1)   
  
#We can collect the observed and predicted values into a data frame, then use  
# the caret function defaultSummary to estimate the test set performance  
DT\_model\_lm\_pred <- data.frame(obs=Y\_test,pred=predicted\_result)  
res\_sum <- defaultSummary(DT\_model\_lm\_pred)  
mape\_score <- MLmetrics::MAPE(predicted\_result,Y\_test)  
return(cbind(res\_sum,mape\_score))  
}  
  
kable(list(test\_model(lmg,X\_test)  
 ))

res\_sum

mape\_score

RMSE

2211.2628816

0.964447

Rsquared

0.0057971

0.964447

MAE

2206.6991364

0.964447

mape\_score <- MLmetrics::MAPE(predict(lmg, moneyball\_train1),moneyball\_test1[,1])

## Warning in y\_true - y\_pred: longer object length is not a multiple of shorter  
## object length

## Warning in (y\_true - y\_pred)/y\_true: longer object length is not a multiple of  
## shorter object length

test\_model(lmg,moneyball\_test1)

## res\_sum mape\_score  
## RMSE 2.211263e+03 0.964447  
## Rsquared 5.797079e-03 0.964447  
## MAE 2.206699e+03 0.964447