Critical Thinking Group 4 - HW1

Sreejaya, Suman, Vuthy

September 12, 2016

Purpose

The purpose of this experiment is to try to predict the amount of wins for a baseball team using the (modified) moneyball dataset. This dataset contains approximately 2200 observations with 17 variables. Each observation represents the performance of a professional baseball team from 1871 to 2006. The statistics have been adjusted to match the performance of a 162 game season.

Dataset:
Moneyball Training Data
Moneyball Evaluation Data

1. Data Exploration

The dependent (response) variable is *TARGET_WINS*. Excluding INDEX, the rest of the variables are the independent variables (predictors). Lets review how each of these independent variables are distributed & how each of these indepdent variable relates to the response variable 'TARGET_WINS'.

1.1 Missing Values

Review the *measure of the center* for the given variables. A quick look at the summary statistics indicate that there are missing values for some of the predictors.

```
##  TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B  TEAM_BATTING_HR
##  Min.   : 891   Min.   : 69.0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:1383   1st Qu.:208.0   1st Qu.: 34.00   1st Qu.: 42.00
##  Median :1454   Median :238.0   Median : 47.00   Median :102.00
##  Mean   :1469   Mean   :241.2   Mean   : 55.25   Mean   : 99.61
##  3rd Qu.:1537   3rd Qu.:273.0   3rd Qu.: 72.00   3rd Qu.:147.00
##  Max.   :2554   Max.   :458.0   Max.   :223.00   Max.   :264.00
##
##  TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB TEAM_BASERUN_CS
##  Min.   :  0.0   Min.   :   0.0   Min.   :  0.0   Min.   :  0.0
##  1st Qu.:451.0   1st Qu.: 548.0   1st Qu.: 66.0   1st Qu.: 38.0
##  Median :512.0   Median : 750.0   Median :101.0   Median : 49.0
##  Mean   :501.6   Mean   : 735.6   Mean   :124.8   Mean   : 52.8
##  3rd Qu.:580.0   3rd Qu.: 930.0   3rd Qu.:156.0   3rd Qu.: 62.0
##  Max.   :878.0   Max.   :1399.0   Max.   :697.0   Max.   :201.0
##                  NA's   :102      NA's   :131     NA's   :772
##  TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##  Min.   :29.00    Min.   : 1137   Min.   :  0.0    Min.   :   0.0
##  1st Qu.:50.50    1st Qu.: 1419   1st Qu.: 50.0    1st Qu.: 476.0
##  Median :58.00    Median : 1518   Median :107.0    Median : 536.5
##  Mean   :59.36    Mean   : 1779   Mean   :105.7    Mean   : 553.0
##  3rd Qu.:67.00    3rd Qu.: 1682   3rd Qu.:150.0    3rd Qu.: 611.0
```

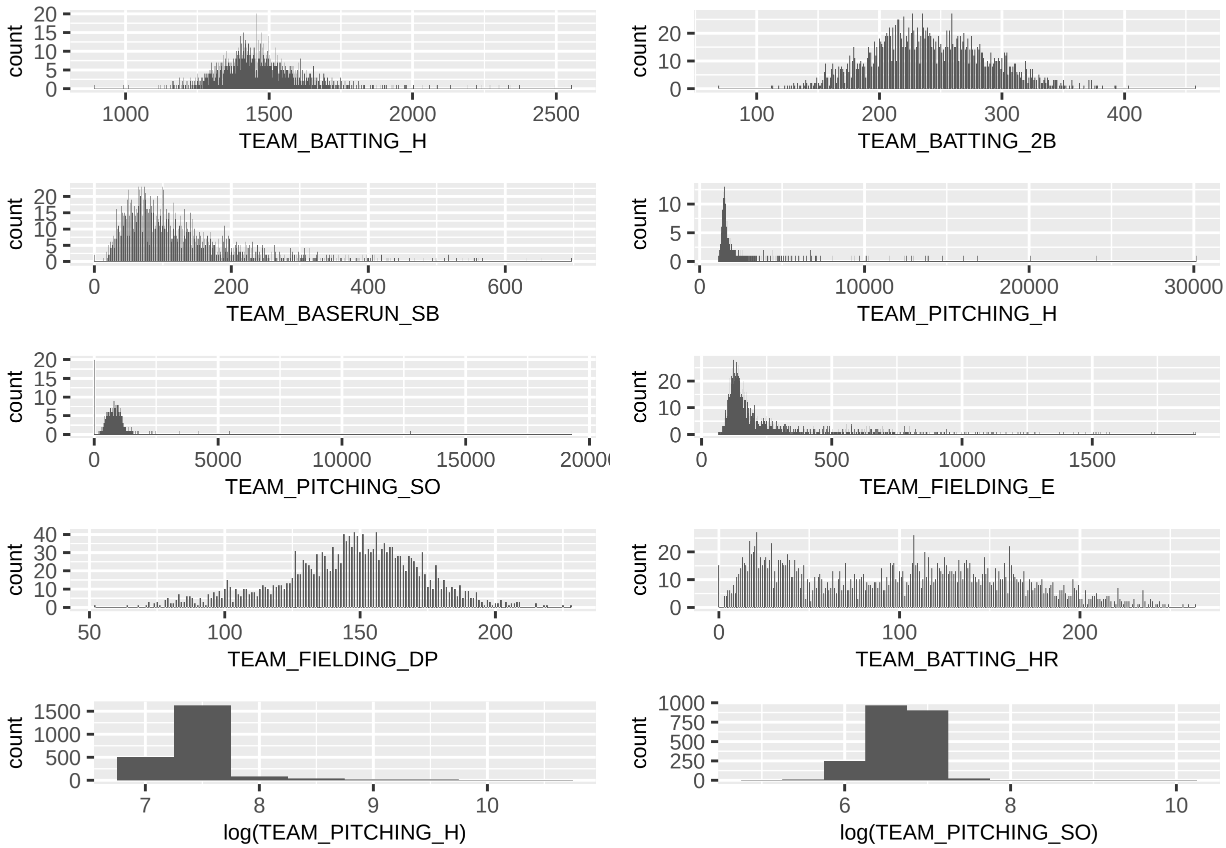
```
## Max.   :95.00    Max.   :30132   Max.   :343.0    Max.   :3645.0
## NA's   :2085
## TEAM_PITCHING_SO  TEAM_FIELDING_E  TEAM_FIELDING_DP
## Min.   :    0.0   Min.   :  65.0   Min.   : 52.0
## 1st Qu.:  615.0   1st Qu.: 127.0   1st Qu.:131.0
## Median :  813.5   Median : 159.0   Median :149.0
## Mean   :  817.7   Mean   : 246.5   Mean   :146.4
## 3rd Qu.:  968.0   3rd Qu.: 249.2   3rd Qu.:164.0
## Max.   :19278.0   Max.   :1898.0   Max.   :228.0
## NA's   :102                        NA's   :286
```

The list of predictor variables with missing data and their counts:

	Missing	Percentage
TEAM_BATTING_H	0	0.0000000
TEAM_BATTING_2B	0	0.0000000
TEAM_BATTING_3B	0	0.0000000
TEAM_BATTING_HR	0	0.0000000
TEAM_BATTING_BB	0	0.0000000
TEAM_BATTING_SO	102	0.0448155
TEAM_BASERUN_SB	131	0.0575571
TEAM_BASERUN_CS	772	0.3391916
TEAM_BATTING_HBP	2085	0.9160808
TEAM_PITCHING_H	0	0.0000000
TEAM_PITCHING_HR	0	0.0000000
TEAM_PITCHING_BB	0	0.0000000
TEAM_PITCHING_SO	102	0.0448155
TEAM_FIELDING_E	0	0.0000000
TEAM_FIELDING_DP	286	0.1256591

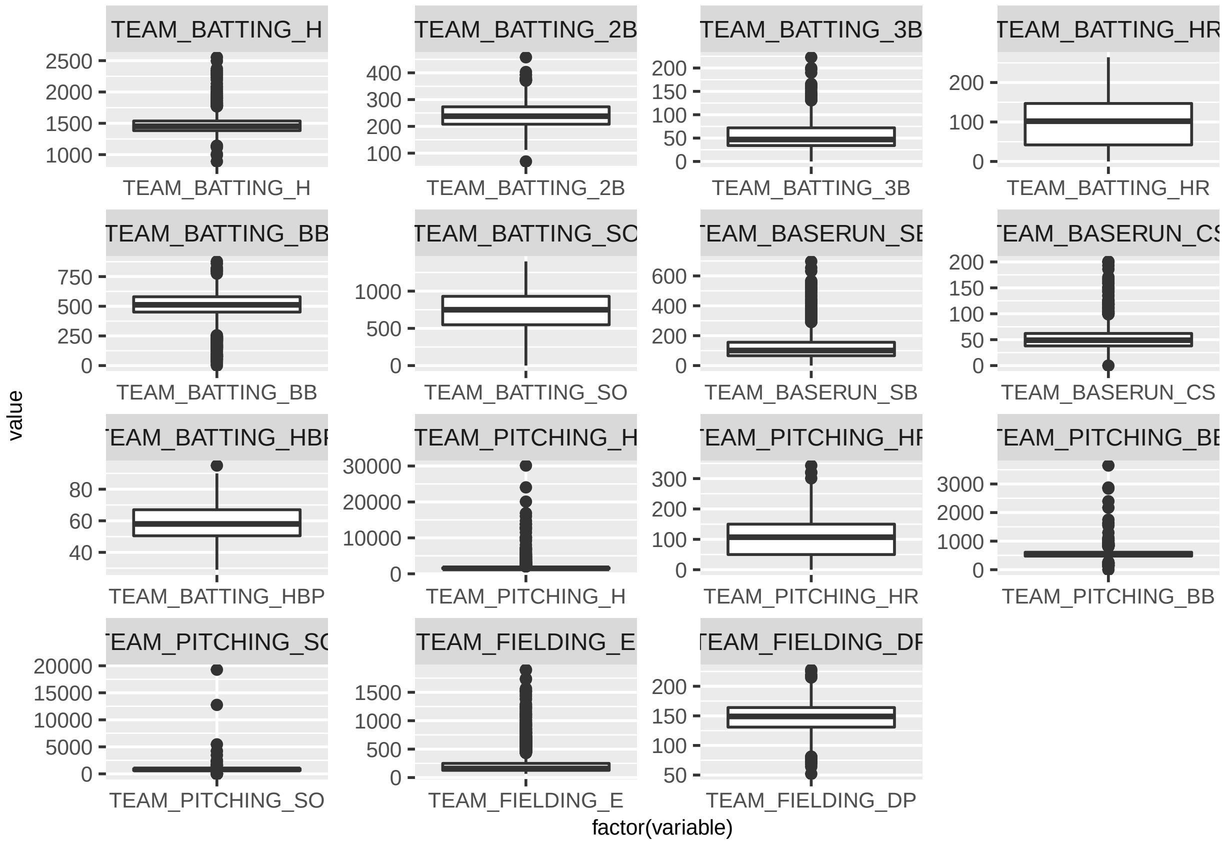
1.2 Distribution of predictors

Review the distributions of the predictors. Here are few histograms of the predictors.



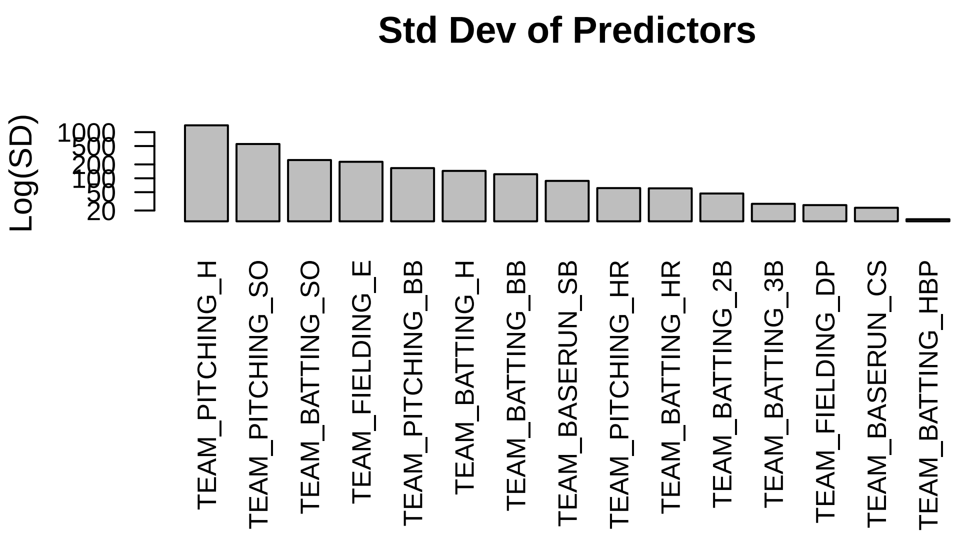
Based on the summary of the data, and the histograms, there are outliers and the distributions of the few of the predictors are skewed. Notice that *TEAM_PITCHING_H* and *TEAM_PICTCHING_SO* distributions are not visible at all in the above diagram, so the log transformation has been applied in the above.

Lets also review the box plot's of the predictors.



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_BATTING_HBP
TEAM_PITCHING_H
TEAM_PITCHING_HR
TEAM_PITCHING_BB
TEAM_PITCHING_SO
TEAM_FIELDING_E
TEAM_FIELDING_DP
value
factor(variable)

1.3 Standard Deviation



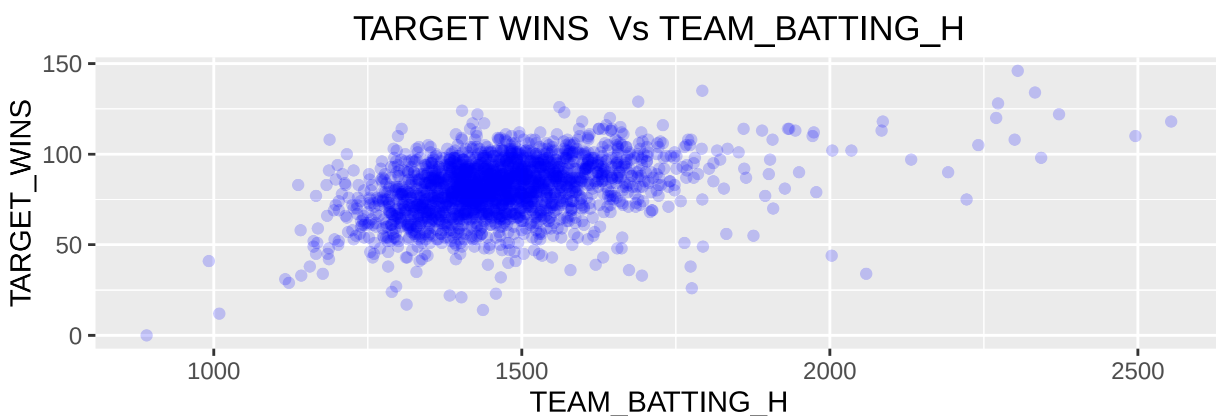
	std
TEAM_BATTING_H	144.59120
TEAM_BATTING_2B	46.80141
TEAM_BATTING_3B	27.93856
TEAM_BATTING_HR	60.54687
TEAM_BATTING_BB	122.67086
TEAM_BATTING_SO	248.52642
TEAM_BASERUN_SB	87.79117
TEAM_BASERUN_CS	22.95634
TEAM_BATTING_HBP	12.96712
TEAM_PITCHING_H	1406.84293
TEAM_PITCHING_HR	61.29875
TEAM_PITCHING_BB	166.35736
TEAM_PITCHING_SO	553.08503
TEAM_FIELDING_E	227.77097
TEAM_FIELDING_DP	26.22639

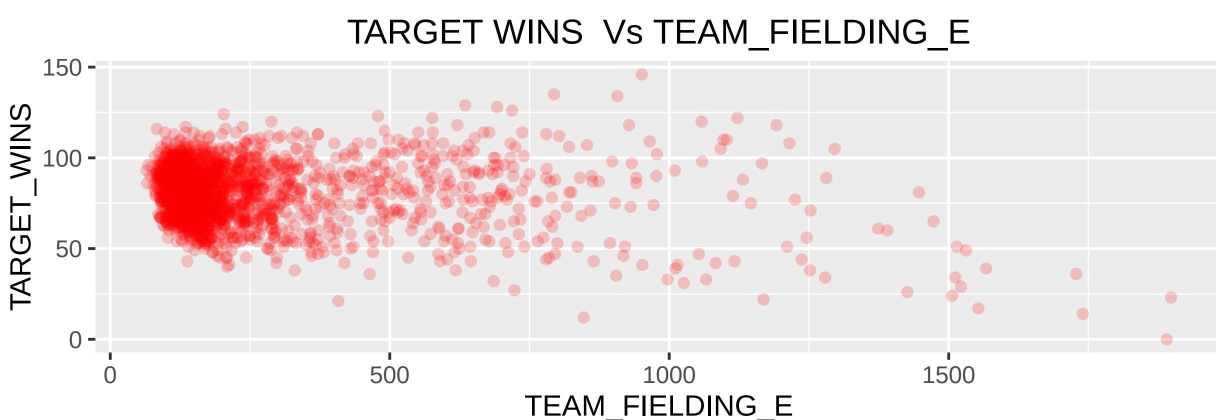
1.4 Correlation

Find correlation of Response variable with predictor variables

Variable	Correlation
TARGET_WINS	1.000
TEAM_BATTING_H	0.389
TEAM_BATTING_2B	0.289
TEAM_BATTING_3B	0.143
TEAM_BATTING_HR	0.176
TEAM_BATTING_BB	0.233
TEAM_BATTING_SO	NA
TEAM_BASERUN_SB	NA
TEAM_BASERUN_CS	NA
TEAM_BATTING_HBP	NA
TEAM_PITCHING_H	-0.110
TEAM_PITCHING_HR	0.189
TEAM_PITCHING_BB	0.124
TEAM_PITCHING_SO	NA
TEAM_FIELDING_E	-0.176
TEAM_FIELDING_DP	NA

From the above we can see that the *TEAM_BATTING_H* is high positively correlated, and *TEAM_FIELDING_E* has the negative corelation with the *TARGET_WINS*. Lets just visualize these two:





2. Data Preparation

2.1.Eliminate variables with most missing data

It is clear from the dataset ‘moneyball.missing’ that ‘TEAM BATTING HBP’ has more than 90% missing items. So we are removing the variable from the dataset.

2.2 Imputation of missing data

We have noticed that there are missing values for predictors, lets impute of missing values with mean.

After imputation, the missing values should not be there.

	mb.imp
TEAM_BATTING_H	0
TEAM_BATTING_2B	0
TEAM_BATTING_3B	0
TEAM_BATTING_HR	0
TEAM_BATTING_BB	0
TEAM_BATTING_SO	0
TEAM_BASERUN_SB	0
TEAM_BASERUN_CS	0
TEAM_PITCHING_H	0
TEAM_PITCHING_HR	0
TEAM_PITCHING_BB	0
TEAM_PITCHING_SO	0
TEAM_FIELDING_E	0
TEAM_FIELDING_DP	0

Correlation of response variable to predictor variable after imputing data

Variable	Correlation
TARGET_WINS	1.000
TEAM_BATTING_H	0.389
TEAM_BATTING_2B	0.289
TEAM_BATTING_3B	0.143
TEAM_BATTING_HR	0.176
TEAM_BATTING_BB	0.233
TEAM_BATTING_SO	-0.031
TEAM_BASERUN_SB	0.123
TEAM_BASERUN_CS	0.016
TEAM_PITCHING_H	-0.110
TEAM_PITCHING_HR	0.189
TEAM_PITCHING_BB	0.124
TEAM_PITCHING_SO	-0.076
TEAM_FIELDING_E	-0.176
TEAM_FIELDING_DP	-0.029

3. Build Models

Lets try to build different models to predict the *TARTGET_WINS*. The first thing we would like to do is to split our given dataset into 'training' and 'test' datasets.

Lets take sample of 75% observations into *training* bucket, which we will use for the model building, and the remaining 25% into *test* bucket, which can be used to compare the model predictions with the actuals.

Number of observations in *training* dataset is 1707 Number of observations in *test* dataset is 409

The below are the few different approaches we will try to build the models:

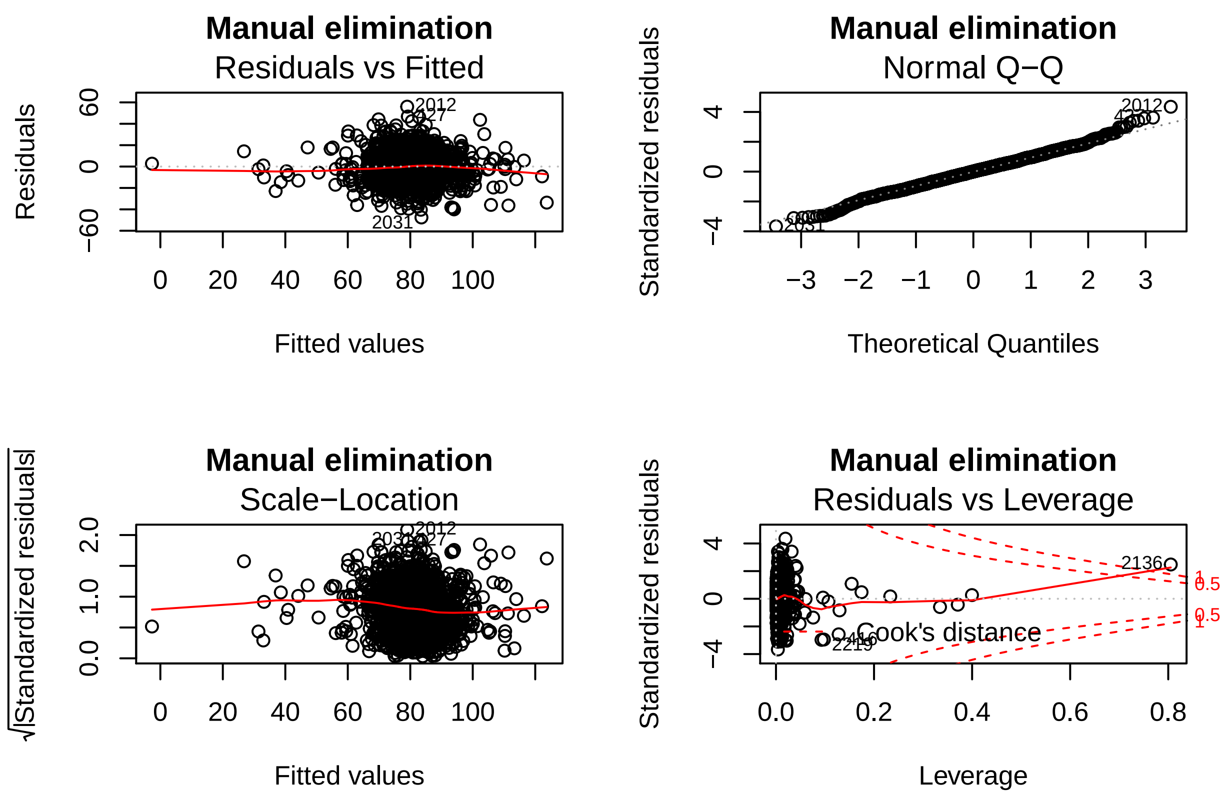
1. Manual elimination
2. Stepwise Regression
3. Stepwise Backward
4. Stepwise Forward
5. High Variance Inflation Factor (VIF) , high p-value predictors elimination.

3.1 Manual elimination

Lets try to fit a multiple linear regression model with TARGET_WINS as the response variable all the other predictors as the explanatory variables except 'TEAM BASERUN CS','TEAM BATTING HBP','TEAM BATTING SO' as they have very low correlation with Wins: (Note: Since we do not need INDEX field, We will be removing INDEX data element from the model building)

The coefficients are:

```
##      (Intercept)   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##     5.826962e-04     1.214980e-35     7.589261e-02     1.278705e-04
##   TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BASERUN_SB  TEAM_PITCHING_H
##     4.651885e-01     1.499784e-01     1.193355e-07     3.816406e-01
## TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##     3.337000e-01     9.055960e-01     9.934444e-01     8.507711e-12
## TEAM_FIELDING_DP
##     3.188035e-12
```



The adjusted r-squared values is 0.3108851

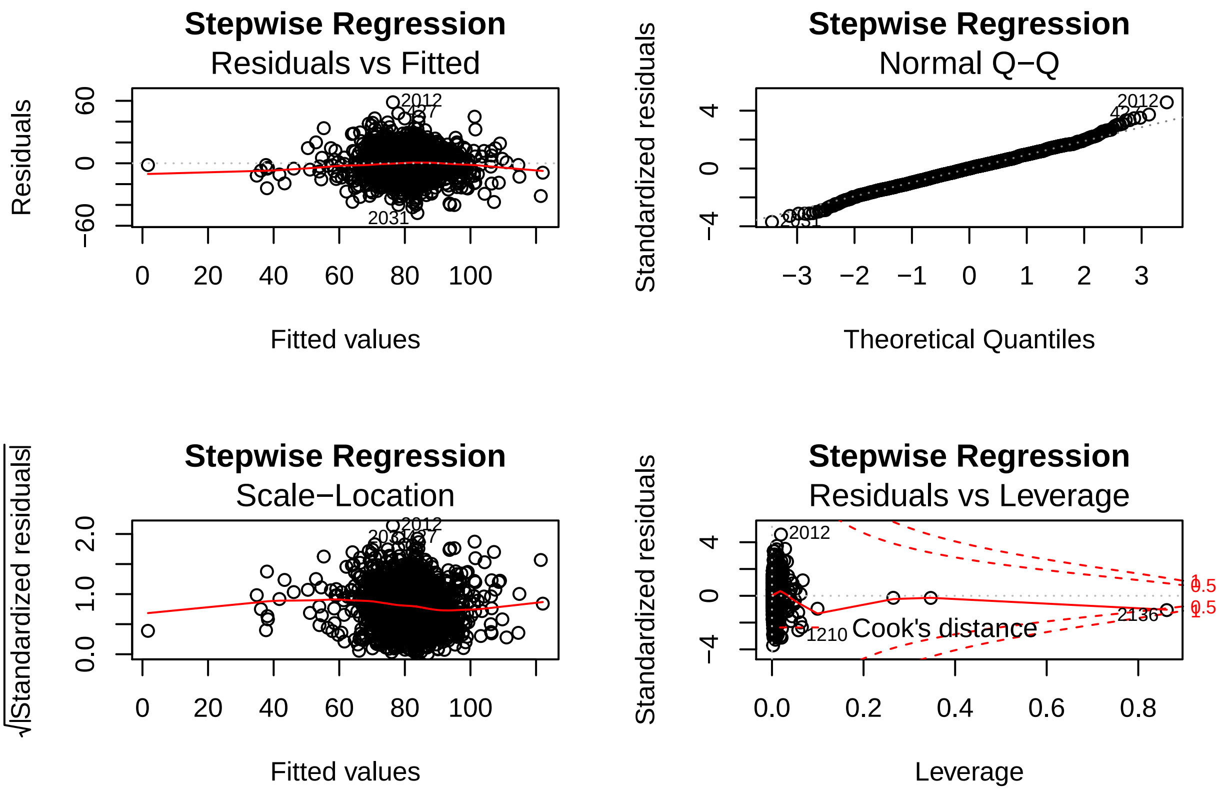
In the residuals Vs Fitted graph, the red line is about flat, which indicates the linearity in residuals is good. In the scale-location graph as well, the red line is about flat, which indicates that residual variance is constant [homo scadasticity assumption]. The Normal Q-Q graph indicates that the most of the residuals are on the straight line.However, the Residual Vs Leverage plot has the redline not alligned with gray dotted line, this indicates that the assumption of standardized residuals centered around zero is NOT true here.

3.2 Stepwise Regression

Here, we will be selecting the predictors based on stepwise regression.

The coefficients we obtained here are:

```
##      (Intercept)   TEAM_BATTING_H  TEAM_BATTING_3B  TEAM_BATTING_HR
##     3.093915e-09     5.706711e-35     3.874854e-04     5.755982e-13
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##     6.395560e-02     7.608175e-06     1.160772e-10     1.295941e-01
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##     8.623681e-02     5.443116e-13     2.175865e-14
```



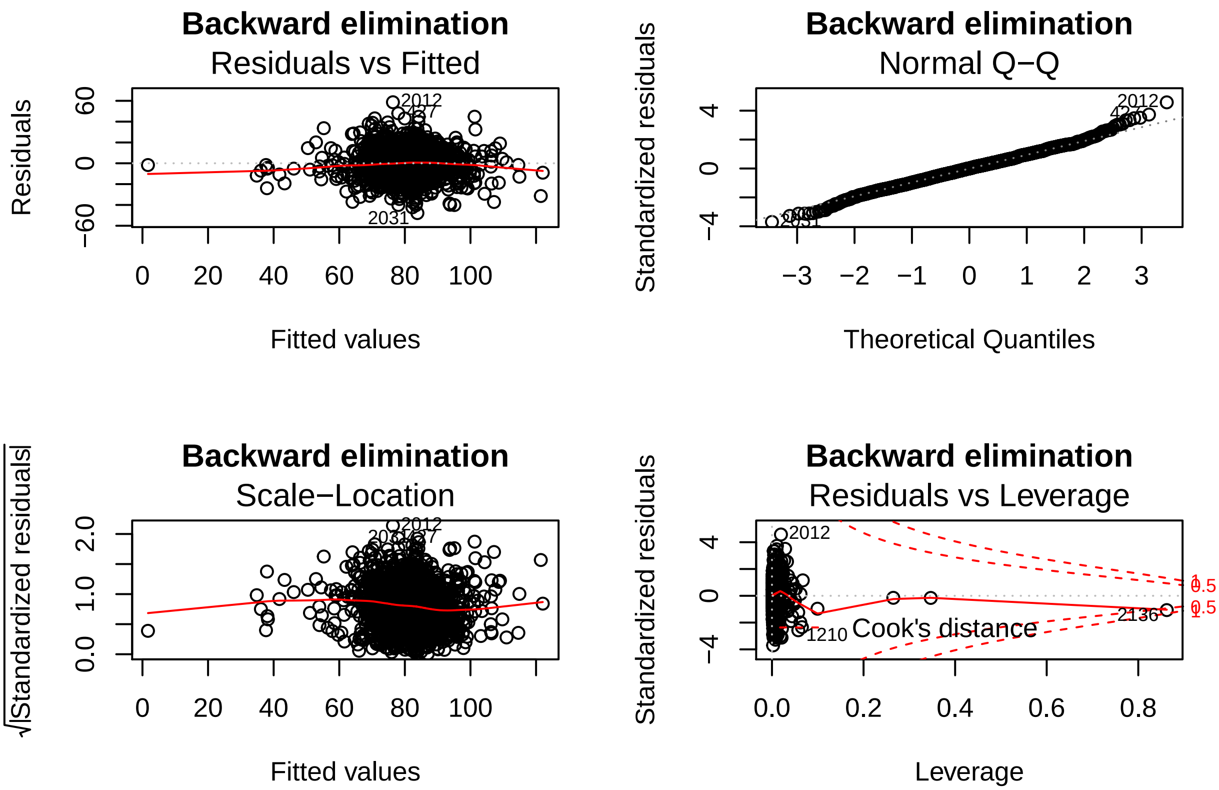
And the adjusted r-squared value is 0.3177756

In the residuals Vs Fitted graph, the red line is about flat, which indicates the linearity in residuals is good. In the scale-location graph as well, the red line is about flat, which indicates that residual variance is constant [homo scadasticity assumption]. The Normal Q-Q graph indicates that the most of the residuals are on the straight line. However, the Residual Vs Leverage plot has the redline not alligned with gray dotted line, this indicates that the assumption of standardized residuals centered around zero is NOT true here.

3.3 Stepwise Backward

The coefficients we obtained here are:

```
##      (Intercept)   TEAM_BATTING_H  TEAM_BATTING_3B  TEAM_BATTING_HR
##     3.093915e-09     5.706711e-35     3.874854e-04     5.755982e-13
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##     6.395560e-02     7.608175e-06     1.160772e-10     1.295941e-01
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##     8.623681e-02     5.443116e-13     2.175865e-14
```



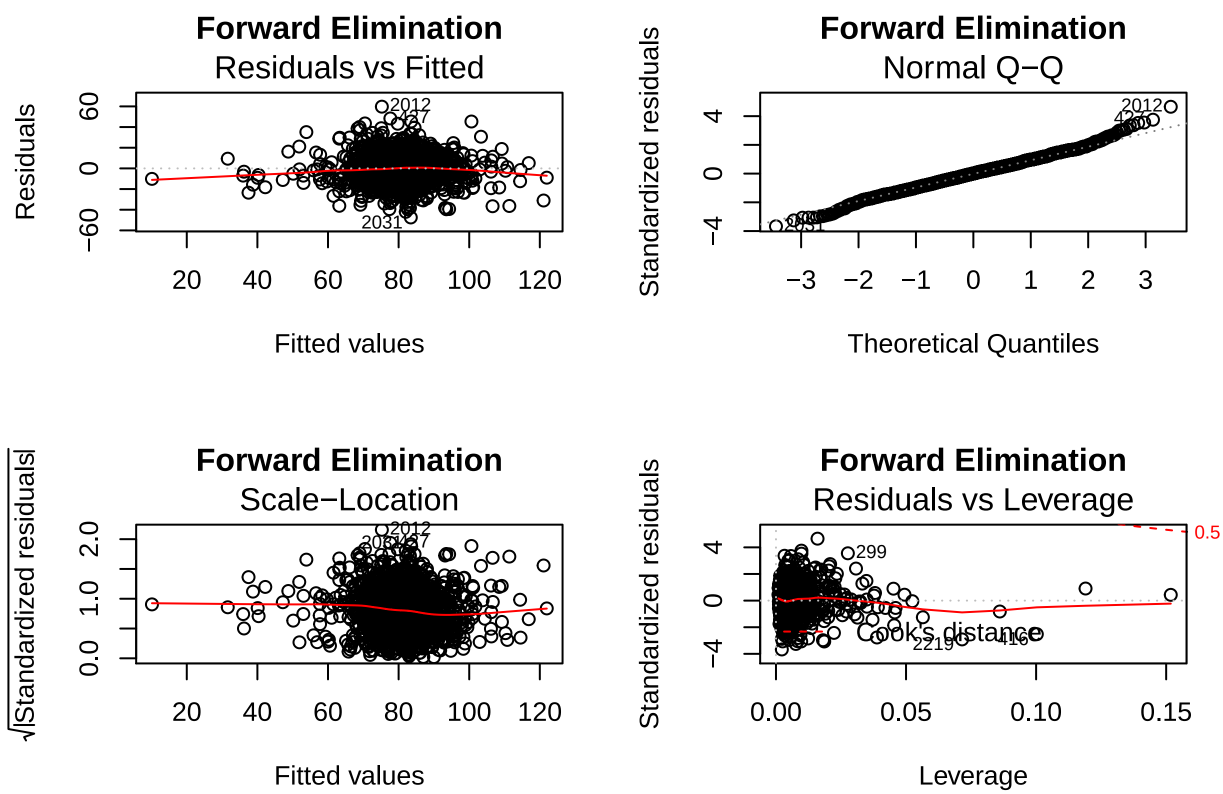
The adjusted r-squared value in the stepwise backward model is 0.3177756

In the residuals Vs Fitted graph, the red line is about flat, which indicates the linearity in residuals is good. In the scale-location graph as well, the red line is about flat, which indicates that residual variance is constant [homo scadasticity assumption]. The Normal Q-Q graph indicates that the most of the residuals are on the straight line. However, the Residual Vs Leverage plot has the redline not alligned with gray dotted line, this indicates that the assumption of standardized residuals centered around zero is NOT true here.

3.4 Stepwise Forward

The coefficients we obtained here are:

```
##      (Intercept)   TEAM_BATTING_H  TEAM_FIELDING_E  TEAM_BASERUN_SB
##     9.932550e-10     3.134153e-33     5.685008e-22     1.994959e-11
## TEAM_FIELDING_DP TEAM_PITCHING_HR  TEAM_BATTING_SO  TEAM_BATTING_3B
##     3.026275e-14     2.168497e-01     3.065198e-05     1.750888e-04
##  TEAM_BATTING_HR  TEAM_BATTING_BB
##     5.722233e-02     6.932204e-02
```



The adjusted r-squared value in the stepwise Forward model is 0.3173093

In the residuals Vs Fitted graph, the red line is about flat, which indicates the linearity in residuals is good. In the scale-location graph as well, the red line is about flat, which indicates that residual variance is constant [homo scadasticity assumption]. The Normal Q-Q graph indicates that the most of the residuals are on the straight line. And notice here in the Residuals Vs Leverage graph, the standardized residuals are some what centered around zero. (the red line stays closer to the horizontal gray dashed line, this indicates that the assumption of standardized residuals centered around zero holds true)

3.5 Remove VIF, and high p value predictors manually.

In this model we would be removing the multi-collinear predictors - basically removing the excessive correlation among the explanatory variables. And then try removing the high p value predictors (> 0.05)

The below is the VIF values, lets get rid of those that has got VIF > 5.

	VIF
TEAM_BATTING_HR	40.344531
TEAM_PITCHING_HR	32.787725
TEAM_BATTING_BB	6.929274
TEAM_PITCHING_BB	6.329970
TEAM_BATTING_SO	5.054896
TEAM_FIELDING_E	4.347904
TEAM_BATTING_H	3.873235
TEAM_PITCHING_H	3.609578
TEAM_BATTING_3B	2.956489

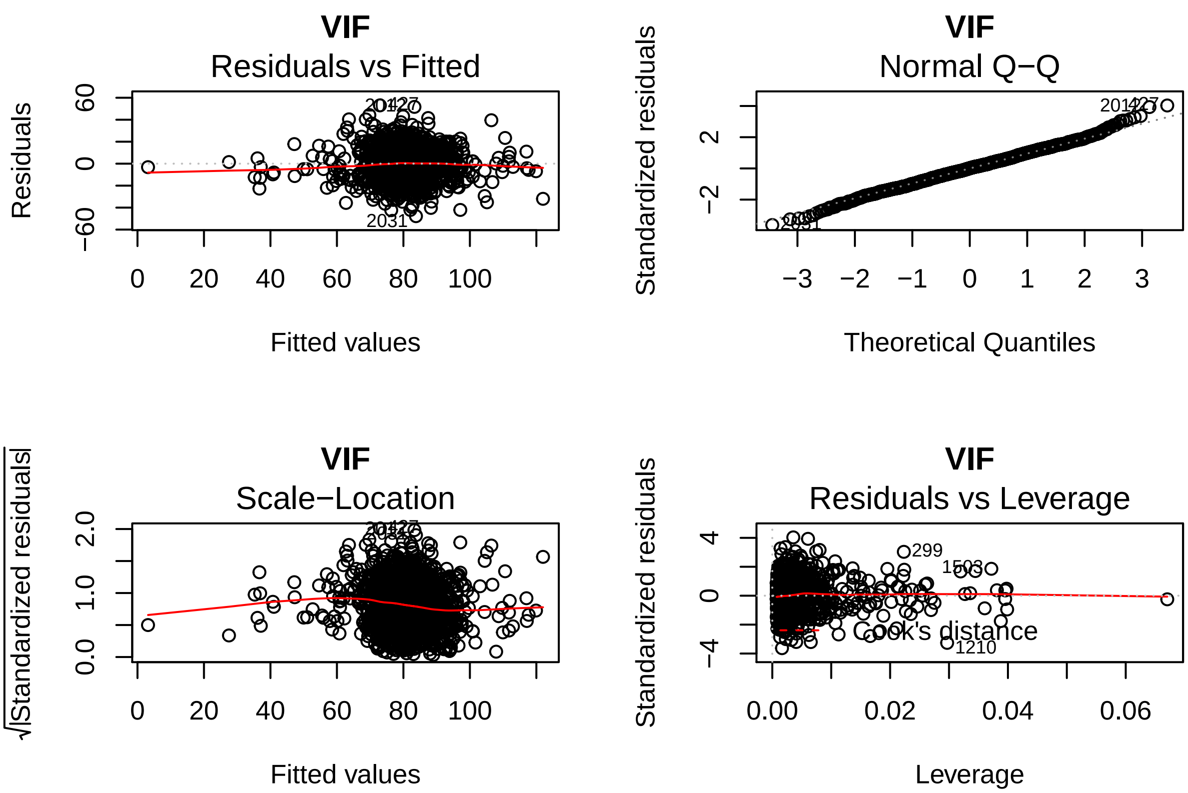
	VIF
TEAM_PITCHING_SO	2.627910
TEAM_BATTING_2B	2.462999
TEAM_BASERUN_SB	1.932370
TEAM_FIELDING_DP	1.373552
TEAM_BASERUN_CS	1.204968

Lets remove *TEAM_BATTING_HR TEAM_PITCHING_HR TEAM_BATTING_BB TEAM_PITCHING_BB* , these highly corelated, which results in multi-colineary among these variables, lets get rid of these from the model building.

These predictors: *TEAM_BATTING_3B, TEAM_BATTING_2B, TEAM_BATTING_SO, TEAM_BATTING_HBP, TEAM_PITCHING_H, TEAM_PITCHING_SO* has got high p value, so, lets try removing and re-build the model:

Here are the final co-efficients we got:

```
##      (Intercept)   TEAM_BATTING_H  TEAM_BASERUN_SB  TEAM_BASERUN_CS
##     6.637961e-07    1.108413e-103     2.109642e-14     2.040959e-02
##  TEAM_FIELDING_E TEAM_FIELDING_DP
##     5.296020e-62     1.058256e-08
```



The adjusted r-squared value we got from the above model is 0.2920768

In the residuals Vs Fitted graph, the red line is about flat, which indicates the linearity in residuals is good. In the scale-location graph as well, the red line is about flat, which indicates that residual variance is constant [homo scadasticity assumption]. The Normal Q-Q graph indicates that the most of the residuals are on the straight line (so, errors are normally distributed). And notice here in the Residuals Vs Leverage graph, the standardized residuals are centered around zero. (the red line stays closer to the horizontal gray dashed line, this indicates that the assumption of standardized residuals centered around zero is good)

4. Selection

Lets now check to see how each model performed, by looking at the adjusted r-sqaured, RMSE values.

4.1. Adjusted R-Square

Here is the adjusted R-Sqaured values from different model above:

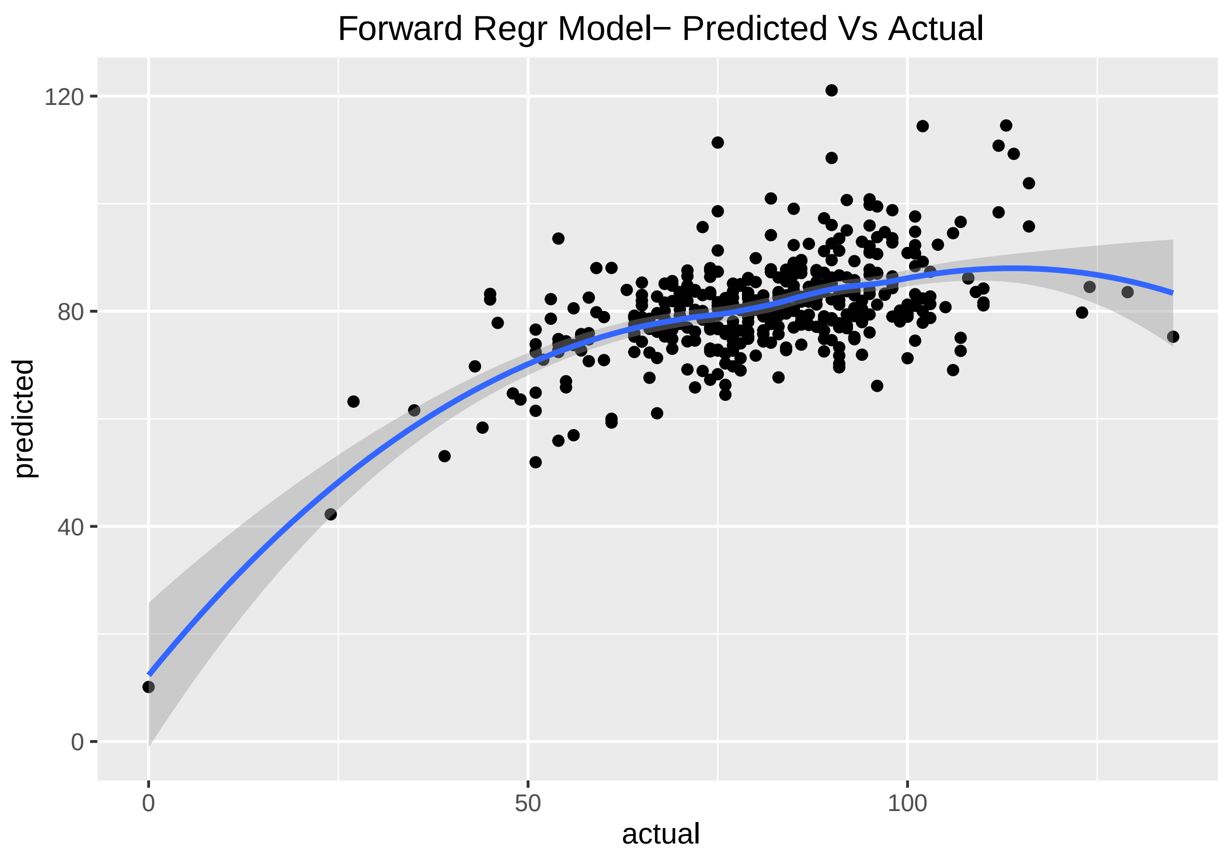
Method	Adj R Squared
Manual Elimination	31.09
Stepwise Regression	31.78
Stepwise Backward	31.78
Stepwise Forward	31.73
VIF Elimination	29.21

Based on our diagnostic observations (assumption of linearity, normality in residuals) from the above models, combined with the above Adj R squared values, we shortlist these 2 model for further validation:

1. Stepwise Forward
2. High Variance Inflation Factor (VIF) , high p-value predictors elimination.

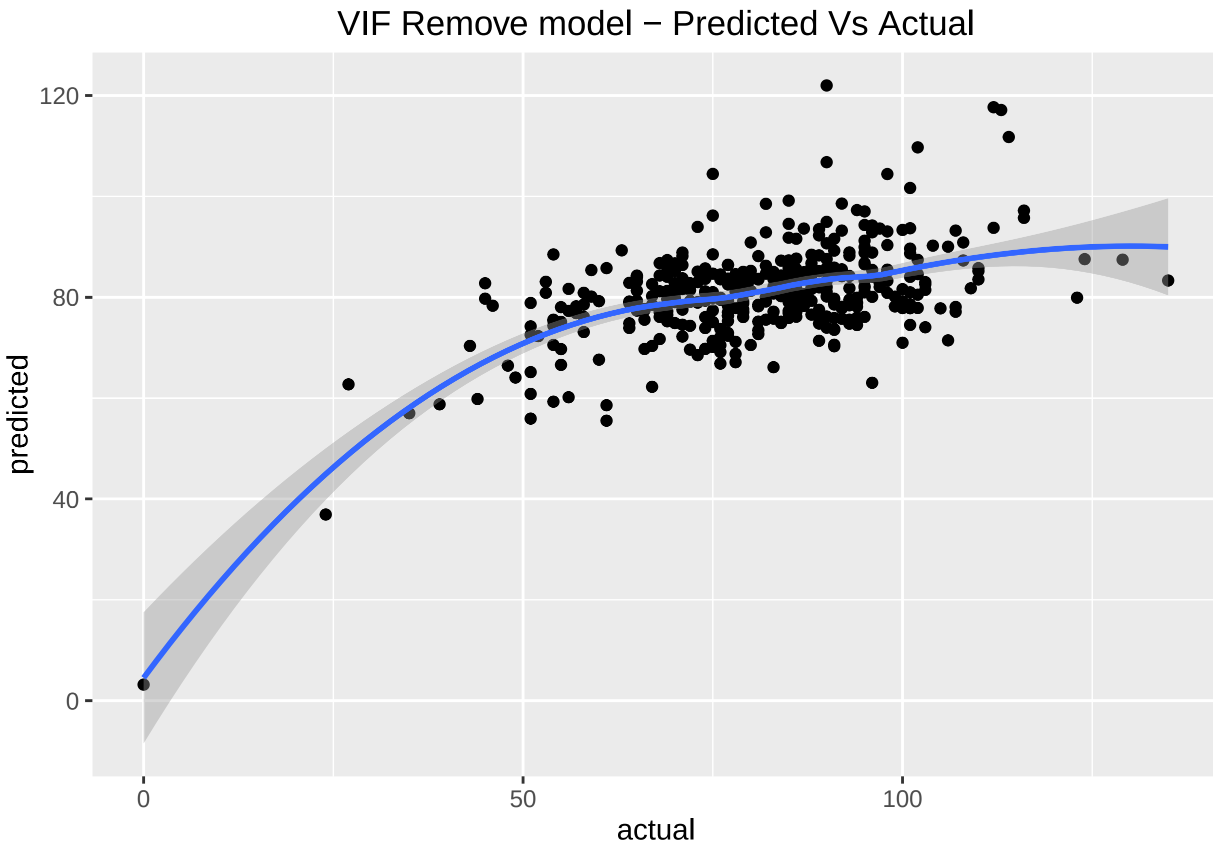
4.2. RMSE - Root Mean Sqared Error (verification with test data)

Lets take our shortlisted models, and apply it on our *test data set*, and compare it with the actuals.



The RMSE in Stepwise Forward Model is 13.8483464

Lets validate the model where we removed the high VIF variables (multicollinearity) :



The RMSE in Stepwise Forward Model is 13.9515691

4.3. Conclusion

Both the models performed similar, however RMSE is slightly lower in the Stepwise Forward model, so we will consider that model as a best fit for our evaluation.

5. Evaluation

We will load the evaluation dataset , and predict the *TARGET_WINS* by applying our final model.

The given evaluation dataset has 259 observations, and the below are the missing values:

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

Lets replace the missing values with column mean & predict.

Click here to view the Predictions for Evaluation File

A. Appendix

```
library(RCurl)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(gridExtra)
library(psych)
library(reshape)
library(MASS)
library(car)
library(recommenderlab)
library(knitr)
# opts_chunk$set(tidy.opts=list(width.cutoff=80),tidy=TRUE)

moneyballTraining <- read.csv("https://raw.githubusercontent.com/Nguyver/DATA621-HW/master/HW1/moneybal
    header = TRUE, sep = ",", stringsAsFactors = FALSE)

summary(moneyballTraining[3:17])

moneyball.NA <- apply(moneyballTraining[3:17], 2, function(x) sum(is.na(x)))
moneyball.missing <- cbind(moneyball.NA, moneyball.NA/nrow(moneyballTraining))
colnames(moneyball.missing) <- c("Missing", "Percentage")
kable(moneyball.missing)

# Explore independent variable TEAM_BATTING_H
g_tbh <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = TEAM_BATTING_H),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

g_b2b <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = TEAM_BATTING_2B),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

g_brsb <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = TEAM_BASERUN_SB),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

g_tph <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = TEAM_PITCHING_H),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

g_tps <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = TEAM_PITCHING_SO),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

g_tfe <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = TEAM_FIELDING_E),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

g_tfd <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = TEAM_FIELDING_DP),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))
```

```r
g_tbhr <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = TEAM_BATTING_HR),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

g_tphLg <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = log(TEAM_PITCHING_H)),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

g_tpsLg <- ggplot(data = moneyballTraining) + geom_histogram(aes(x = log(TEAM_PITCHING_SO)),
    binwidth = 0.5) + theme(axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))

grid.arrange(g_tbh, g_b2b, g_brsb, g_tph, g_tps, g_tfe, g_tfd,
    g_tbhr, g_tphLg, g_tpsLg, ncol = 2)

meltMoneyBallTraining <- melt(moneyballTraining[3:17])
ggplot(meltMoneyBallTraining, aes(factor(variable), value)) +
    geom_boxplot() + facet_wrap(~variable, scale = "free") +
    theme(axis.text = element_text(size = 8), axis.title = element_text(size = 8))

getStandardDev <- function(moneyballTraining) {
    stdDevs <- SD(moneyballTraining[3:17])
    par(mai = c(3, 1.2, 1, 1))

    # transformed the y, due to high variances.
    barplot(stdDevs[order(stdDevs, decreasing = T)], log = "y",
        las = 2, main = "Std Dev of Predictors", xlab = "", ylab = "Log(SD)",
        cex.axis = 0.8, cex.names = 0.8)

    return(stdDevs)
}

std <- getStandardDev(moneyballTraining)
kable(as.data.frame(std))

corData <- round(cor(moneyballTraining), 3)  # rounding makes it easier to look at
t.corData <- t(corData[2, c(2:17)])  # we are only interested on correlation of Team win against all oth
moneyballTraining.cor <- melt(t.corData)  # convert the wide format to long form for ease of read
moneyballTraining.cor <- moneyballTraining.cor[, 2:3]
colnames(moneyballTraining.cor) <- c("Variable", "Correlation")

kable(moneyballTraining.cor)

g1 = ggplot(data = moneyballTraining) + geom_point(aes(x = TEAM_BATTING_H,
    y = TARGET_WINS), alpha = 0.2, color = "blue") + ggtitle("TARGET WINS  Vs TEAM_BATTING_H")

g2 = ggplot(data = moneyballTraining) + geom_point(aes(x = TEAM_FIELDING_E,
    y = TARGET_WINS), alpha = 0.2, color = "red") + ggtitle("TARGET WINS  Vs TEAM_FIELDING_E")

grid.arrange(g1, g2, nrow = 2)
# similarly other specific independent variables Vs target
# wins correlation diagram
```

```
moneyballTraining <- subset(moneyballTraining, select = -TEAM_BATTING_HBP)

# Replacing Missing Values In dataset with column mean
for (i in 1:ncol(moneyballTraining)) {
    moneyballTraining[is.na(moneyballTraining[, i]), i] <- mean(moneyballTraining[,
        i], na.rm = TRUE)
}

mb.imp <- apply(moneyballTraining[3:17], 2, function(x) sum(is.na(x)))
# colnames(mb.imp) <- c('# Missing')
kable(as.data.frame(mb.imp))

corData.imp <- round(cor(moneyballTraining), 3)  # rounding makes it easier to look at
t.corData.imp <- t(corData.imp[2, c(2:17)])  # we are only interested on correlation of Team win agains
moneyballTraining.cor.imp <- melt(t.corData.imp)  # convert the wide format to long form for ease of re
moneyballTraining.cor.imp <- moneyballTraining.cor.imp[, 2:3]

colnames(moneyballTraining.cor.imp) <- c("Variable", "Correlation")
kable(moneyballTraining.cor.imp)

set.seed(11)
samples <- sample(1:nrow(moneyballTraining), 0.75 * nrow(moneyballTraining))
moneyballTraining <- moneyballTraining[samples, ]
moneyballTest <- moneyballTraining[-samples, ]

options(stringsAsFactors = FALSE)
results <- data.frame(character(), numeric())

# Full Model
model.manualElimination <- lm(formula = TARGET_WINS ~ . - INDEX -
    TEAM_BASERUN_CS - TEAM_BATTING_SO, data = moneyballTraining)

summary(model.manualElimination)$coefficients[, 4]

ar1 <- summary(model.manualElimination)$adj.r.squared

results <- rbind(results, c("Manual Elimination", round(ar1 *
    100, 2)))
par(mfrow = c(2, 2))
graphics::plot(model.manualElimination, main = "Manual elimination")

fit <- lm(formula = TARGET_WINS ~ . - INDEX, data = moneyballTraining)

# Stepwise Regression
model.step <- stepAIC(fit, direction = "both", trace = FALSE)
summary(model.step)$coefficients[, 4]

ar2 <- summary(model.step)$adj.r.squared

results <- rbind(results, c("Stepwise Regression", round(ar2 *
    100, 2)))

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

```r
graphics::plot(model.step, main = "Stepwise Regression")

model.step.backward <- step(fit, direction = "backward", trace = FALSE)
summary(model.step.backward)$coefficients[, 4]

ar3 <- summary(model.step.backward)$adj.r.squared

results <- rbind(results, c("Stepwise Backward", round(ar3 *
    100, 2)))

par(mfrow = c(2, 2))
graphics::plot(model.step.backward, main = "Backward elimination")

forward.null <- lm(TARGET_WINS ~ 1, data = moneyballTraining)
forward.full <- lm(TARGET_WINS ~ . - INDEX, data = moneyballTraining)
model.step.forward <- step(forward.null, scope = list(lower = forward.null,
    upper = forward.full), direction = "forward", trace = FALSE)

summary(model.step.forward)$coefficients[, 4]
ar4 <- summary(model.step.forward)$adj.r.squared

results <- rbind(results, c("Stepwise Forward", round(ar4 * 100,
    2)))

par(mfrow = c(2, 2))
graphics::plot(model.step.forward, main = "Forward Elimination")

# Let's consider ALL the variables ( except INDEX).
fit1 <- lm(TARGET_WINS ~ . - INDEX, data = moneyballTraining)

# Lets check for Multi-Collinearity - lets find vif value and
# drop those that has got high vif (>5)
vifFit1 <- vif(fit1)

# sort by descending
vif.df <- as.data.frame(sort(vifFit1, decreasing = T))
names(vif.df) <- c("VIF")
kable(vif.df)

fit2 <- lm(TARGET_WINS ~ . - INDEX - TEAM_BATTING_HR - TEAM_PITCHING_HR -
    TEAM_BATTING_BB - TEAM_PITCHING_BB, data = moneyballTraining)

model.vif <- lm(TARGET_WINS ~ . - INDEX - TEAM_BATTING_HR - TEAM_PITCHING_HR -
    TEAM_BATTING_BB - TEAM_PITCHING_BB - TEAM_BATTING_3B - TEAM_BATTING_2B -
    TEAM_BATTING_SO - TEAM_PITCHING_H - TEAM_PITCHING_SO, data = moneyballTraining)

ar5 <- summary(model.vif)$adj.r.squared
summary(model.vif)$coefficients[, 4]

results <- rbind(results, c("VIF Elimination", round(ar5 * 100,
    2)))

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

```r
graphics::plot(model.vif, main = "VIF")

colnames(results) <- c("Method", "Adj R Squared")
kable(results)

# Lets take our model, and apply it on the test dataset.
predicted.wins <- predict(model.step.forward, newdata = moneyballTest)

# Lets calculate the RMSE
residuals <- moneyballTest$TARGET_WINS - predicted.wins
rmse_forward <- sqrt(mean(residuals^2))

# lets put in ggplot
rmse.df <- data.frame(actual = moneyballTest$TARGET_WINS, predicted = predicted.wins)
ggplot(rmse.df, aes(x = actual, y = predicted)) + geom_point() +
    geom_smooth() + ggtitle("Forward Regr Model- Predicted Vs Actual")

# Lets take our model, and apply it on the test dataset.
predicted.wins2 <- predict(model.vif, newdata = moneyballTest)

# Lets calculate the RMSE
residuals2 <- moneyballTest$TARGET_WINS - predicted.wins2
rmse_vif <- sqrt(mean(residuals2^2))

# lets put in ggplot
rmse.df2 <- data.frame(actual = moneyballTest$TARGET_WINS, predicted = predicted.wins2)
ggplot(rmse.df2, aes(x = actual, y = predicted)) + geom_point() +
    geom_smooth() + ggtitle("VIF Remove model - Predicted Vs Actual")

moneyballEvaluation <- read.csv("https://raw.githubusercontent.com/Nguyver/DATA621-HW/master/HW1/moneyba
    header = TRUE, sep = ",", stringsAsFactors = FALSE)

# check how many na's for each column
apply(moneyballEvaluation, 2, function(x) sum(is.na(x)))

# Replacing Missing Values In dataset with column mean
for (i in 1:ncol(moneyballEvaluation)) {
    moneyballEvaluation[is.na(moneyballEvaluation[, i]), i] <- mean(moneyballEvaluation[,
        i], na.rm = TRUE)
}

moneyballEvaluation$PredictedWins <- round(predict(model.step.forward,
    newdata = moneyballEvaluation))

# write.csv(file='moneyballEvaluation_Predictions.csv',
# moneyballEvaluation)
```