Moneyball

Jorge Fernandes

MSDS 411

24 April 2018

Introduction

The moneyball dataset has sparked many companies, teams, and organizations to understand and utilize the data they generate/gather. This project highlights many pitfalls that those same individuals fall into simply because they forget to do the due diligence and prepare the data before modeling.

This paper will focus on;
1. Data Exploration
2. Data Transformation
3. Model Building
4. How to select the best model

Data Exploration

Step 1: Can we find outliers in our Independent and Dependent variables?

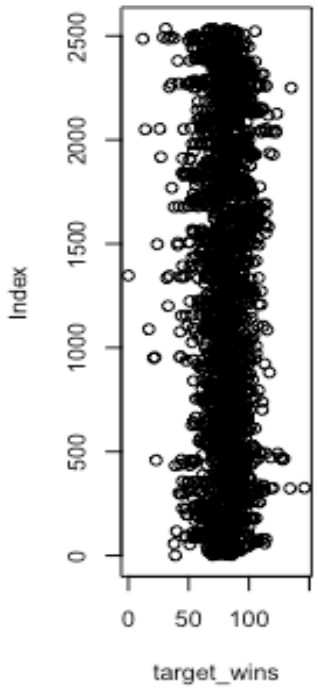
Outliers can cause our model to produce the wrong output by influencing its fit. Creating boxplots will aid in identifying those outliers. We can also use the cleveland dotplot to understand the outliers better. This technique uses the row number against actual value to quickly point out any patterns of outliers. This plot will easilly allow us to check the raw data for errors such as typos during the data collection phase. Points on the far right side, or on the far left side, are observed values that are considerably larger, or smaller, than the majority of the observations, and require further investigation. When we use this chart, together with the box plot and histogram, we can easily identify patterns at to where in the data we're seeing outliers.

```
library(e1071) # to understand skewness
library(dplyr)
library(stringr) # Used to rename the columns by removing the word team
from the column header
library(VIM) # To understand NAs
library(caret)
```

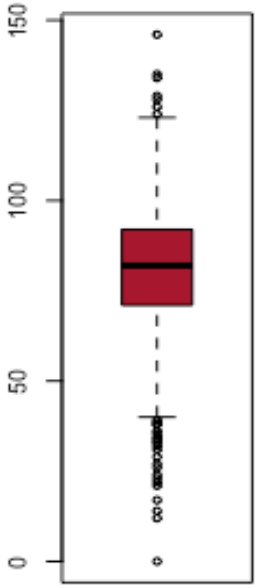
```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone
'zone/tz/2018c.
## 1.0/zoneinfo/America/New_York'

library(mice)
library(MASS) # to use for robust Linear Regression.

# browse to the data
moneyball = read.csv('/Users/legs_jorge/Documents/Data Science
Projects/MSDS_Northwestern/MSDS 411/Unit 01 Moneyball Baseball
Problem/Data/moneyball.csv', header = T)
colnames(moneyball) <- str_replace_all(colnames(moneyball),"TEAM_","")
%>%
  tolower() # Fixing column names
```

```
par(mfrow = c(1, 3))
i = 2
while (i %in% c(2:17)) {

plot(moneyball[,i], moneyball$index, xlab = colnames(moneyball)[i] ,
ylab = "Index", main = paste("cleveland dotplot of
",colnames(moneyball)[i]))

boxplot(moneyball[,i], col = "#A71930", main = paste("Boxplot of
",colnames(moneyball)[i]))

hist(
  moneyball[,i],
  col = "#A71930",
  xlab = colnames(moneyball)[i],
  main = paste("Histogram of ",colnames(moneyball)[i])
)
  i = i + 1
}
```

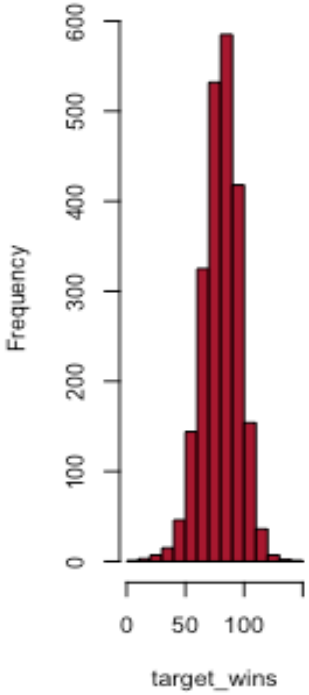
leveland dotplot of target_



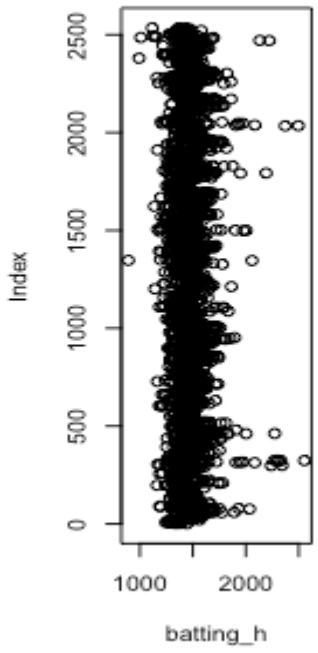
Boxplot of target_wins



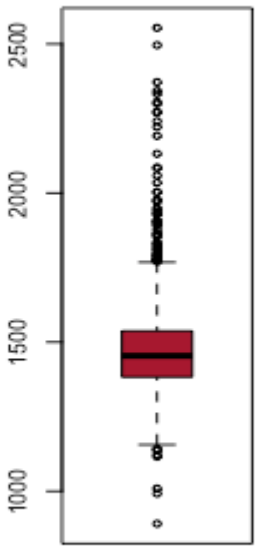
Histogram of target_wir



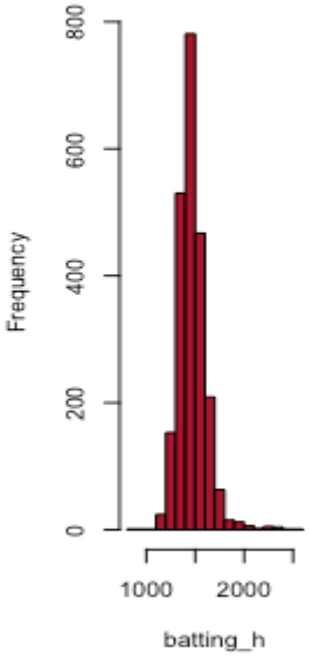
cleveland dotplot of battin

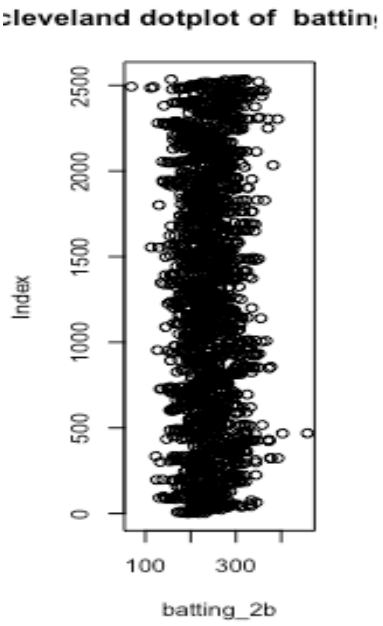


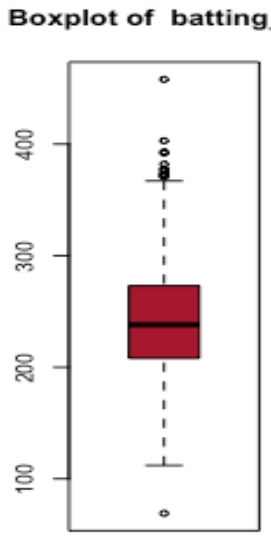
Boxplot of batting_h

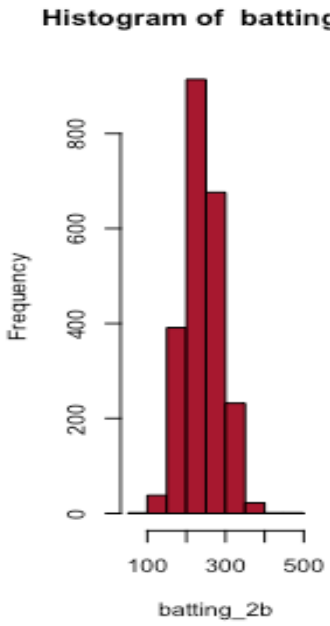


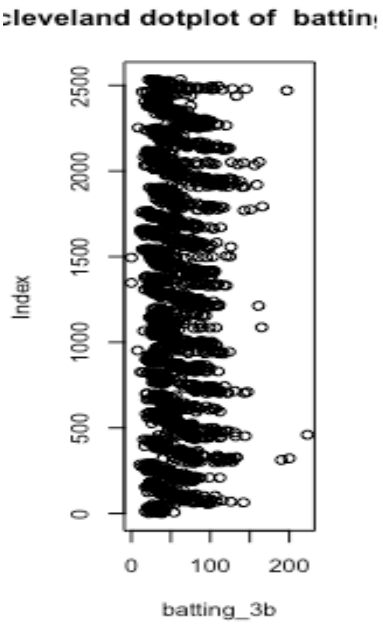
Histogram of batting_h

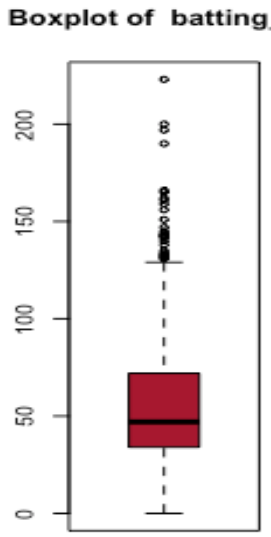


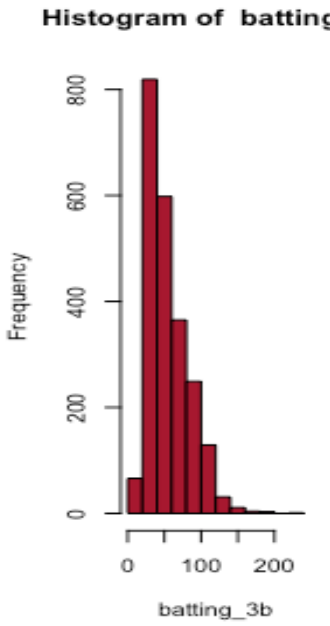

cleveland dotplot of batting
Index
0
500
1000
1500
2000
2500
100
300
batting_2b

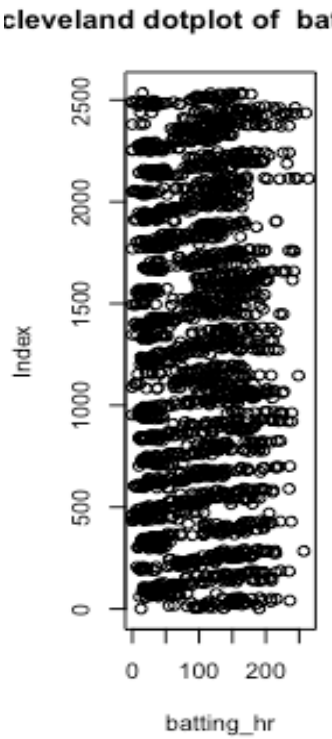

Boxplot of batting_2b
100
200
300
400

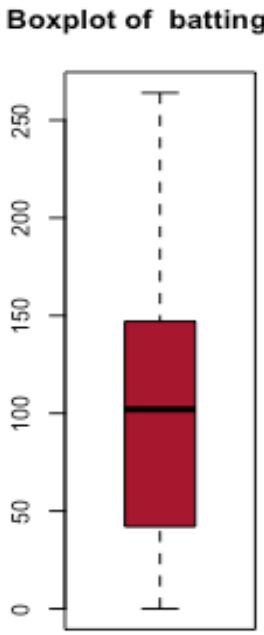

Histogram of batting_2
Frequency
0
200
400
600
800
100
300
500
batting_2b

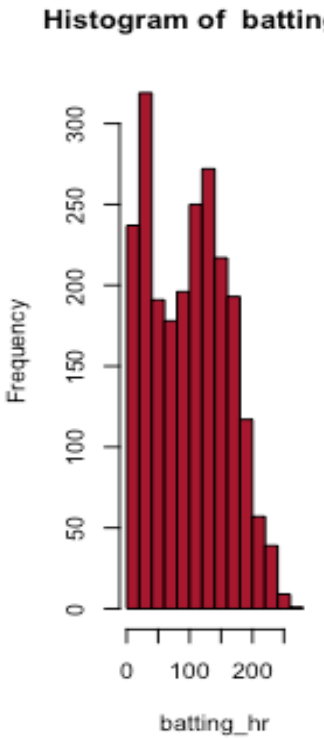

cleveland dotplot of batting
Index
0
500
1000
1500
2000
2500
0
100
200
batting_3b

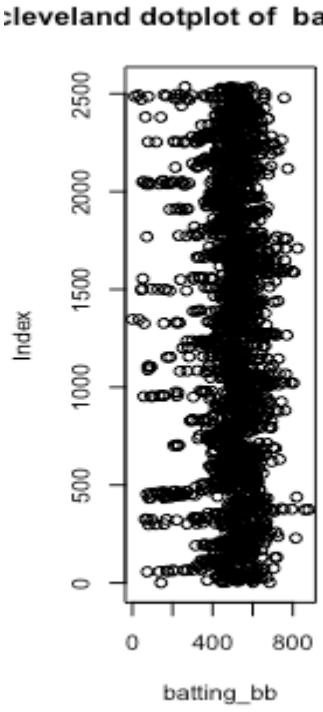

Boxplot of batting_3b
0
50
100
150
200

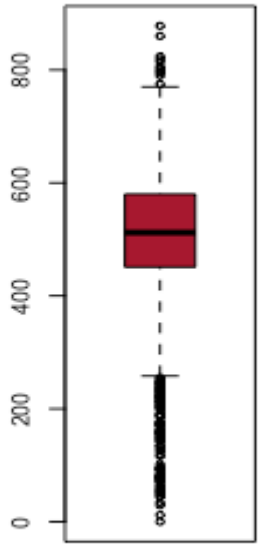

Histogram of batting_3
Frequency
0
200
400
600
800
0
100
200
batting_3b

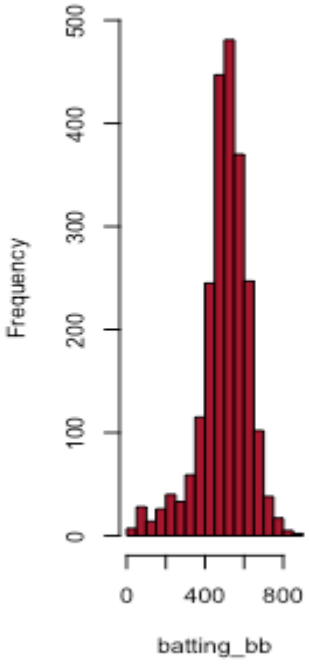

cleveland dotplot of battin
Index
0
500
1000
1500
2000
2500
0
100
200
batting_hr

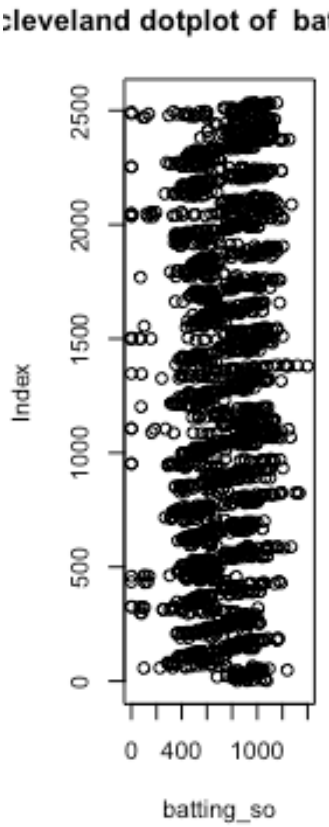

Boxplot of batting_hr
0
50
100
150
200
250

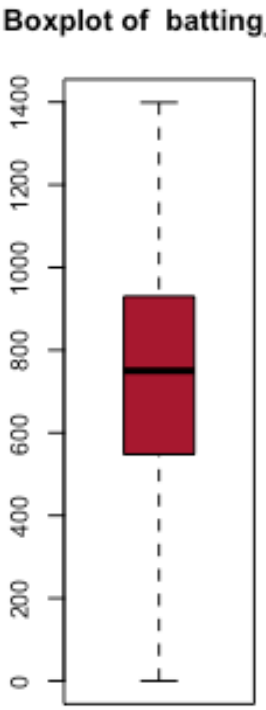

Histogram of batting_h
Frequency
0
50
100
150
200
250
300
0
100
200
batting_hr

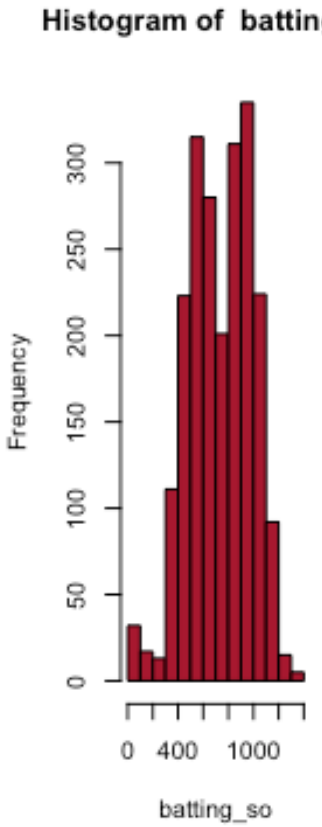

cleveland dotplot of battin
Index
0
500
1000
1500
2000
2500
0
400
800
batting_bb

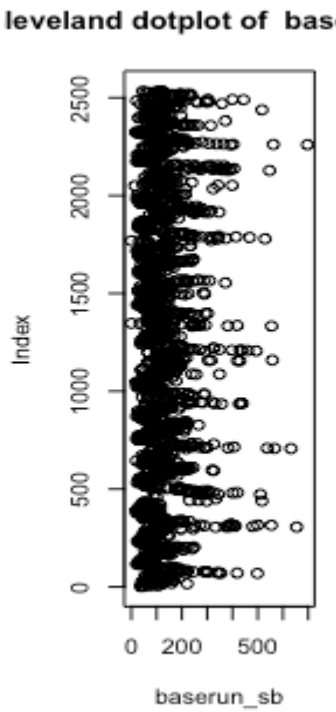

Boxplot of batting_bb
0
200
400
600
800

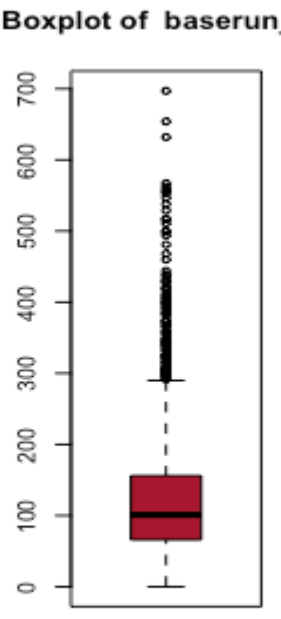

Histogram of batting_b
Frequency
0
100
200
300
400
500
0
400
800
batting_bb

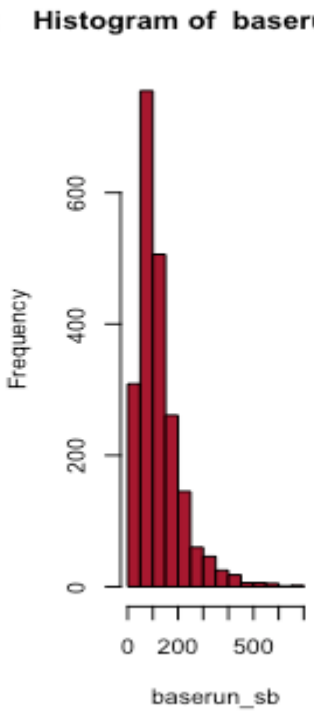

cleveland dotplot of batting
Index
0
500
1000
1500
2000
2500
0
400
1000
batting_so


Boxplot of batting_so
0
200
400
600
800
1000
1200
1400

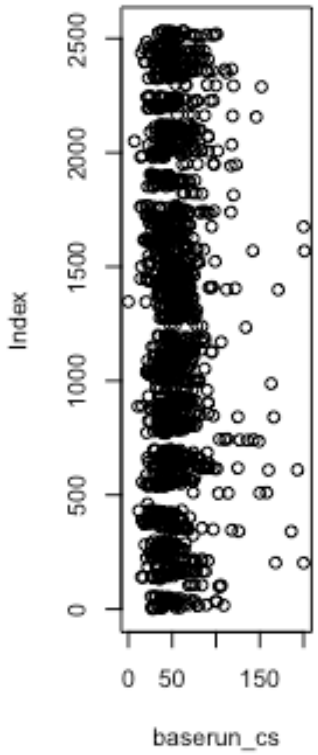

Histogram of batting_s
Frequency
0
50
100
150
200
250
300
0
400
1000
batting_so


leveland dotplot of baseru
Index
0
500
1000
1500
2000
2500
0
200
500
baserun_sb

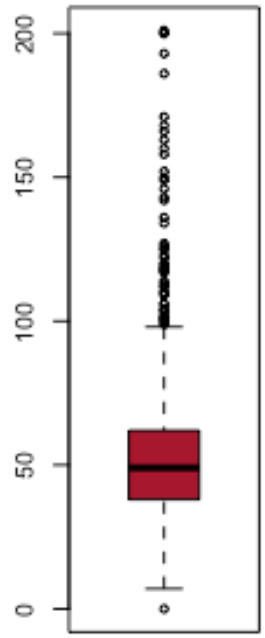

Boxplot of baserun_st
0
100
200
300
400
500
600
700


Histogram of baserun_s
Frequency
0
200
400
600
0
200
500
baserun_sb

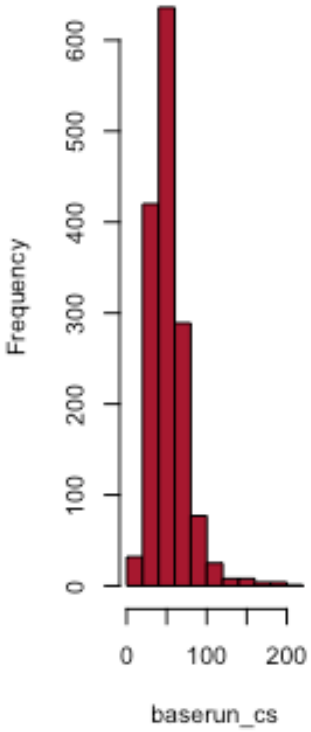
leveland dotplot of baseru



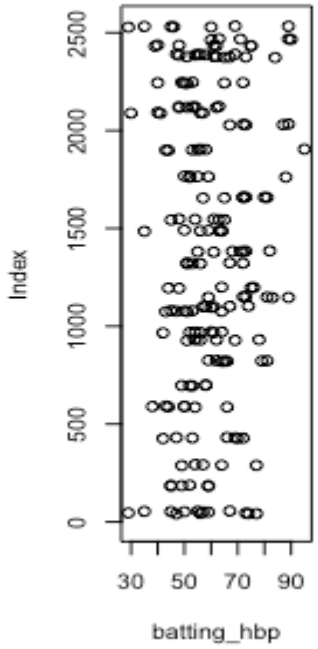
Boxplot of baserun_cs



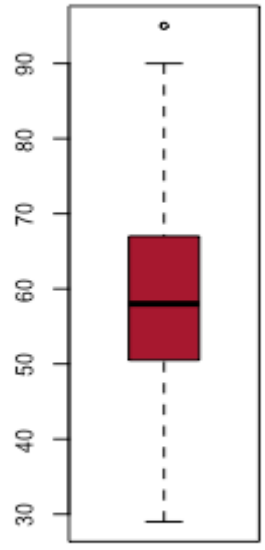
Histogram of baserun_c



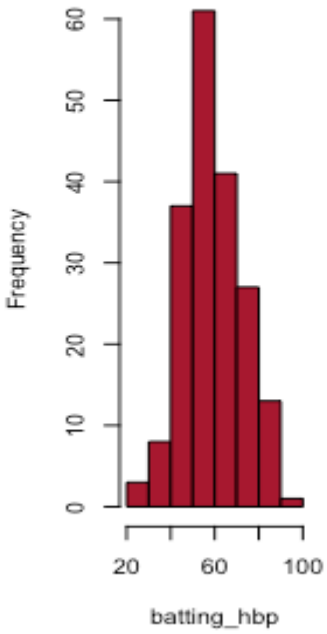
leveland dotplot of batting

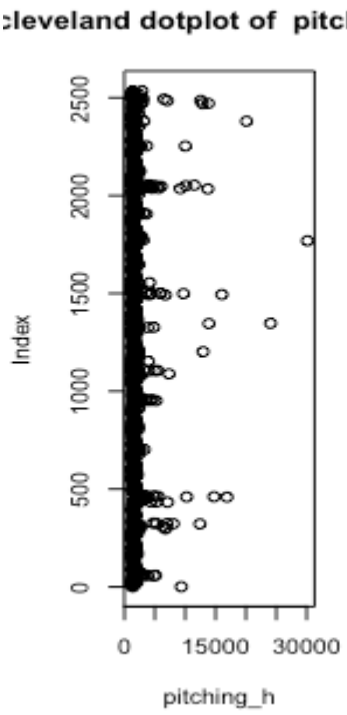


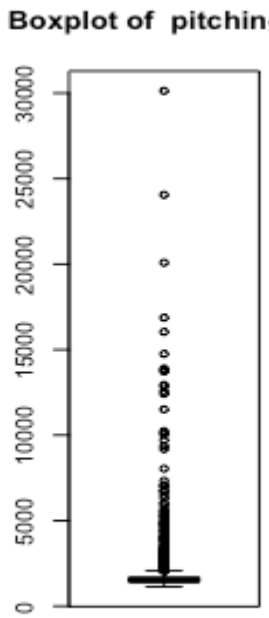
Boxplot of batting_hbp

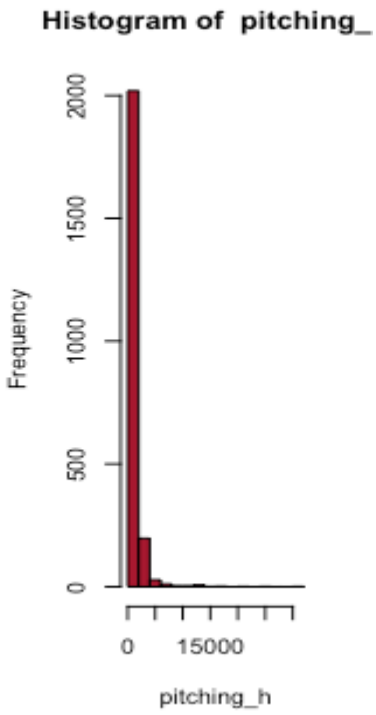


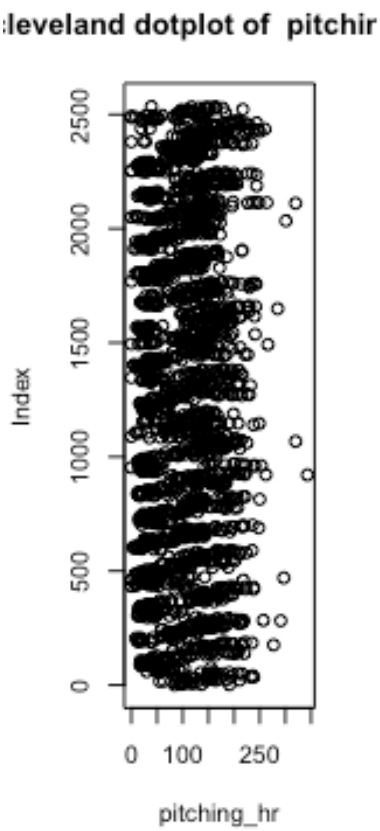
Histogram of batting_hb

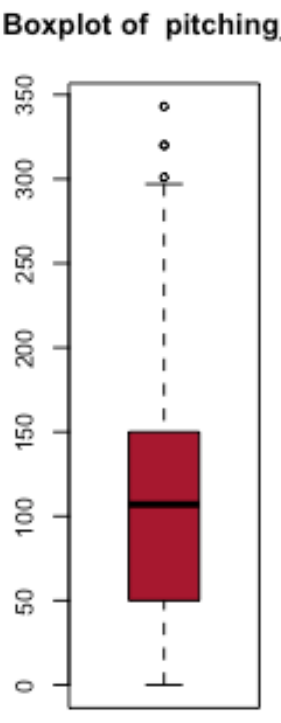


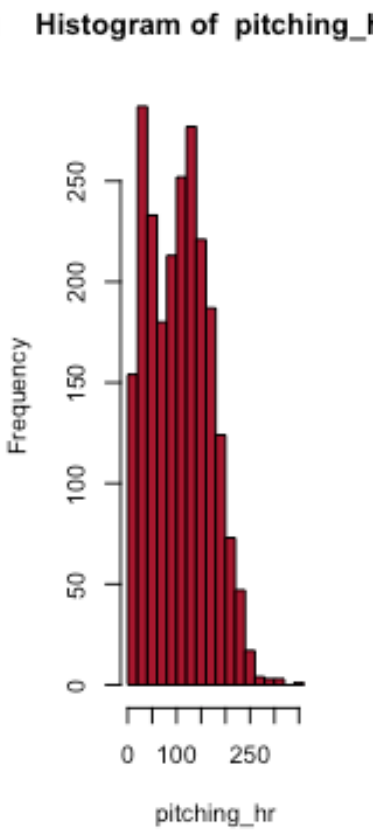

cleveland dotplot of pitchi
Index
0
500
1000
1500
2000
2500
0
15000
30000
pitching_h

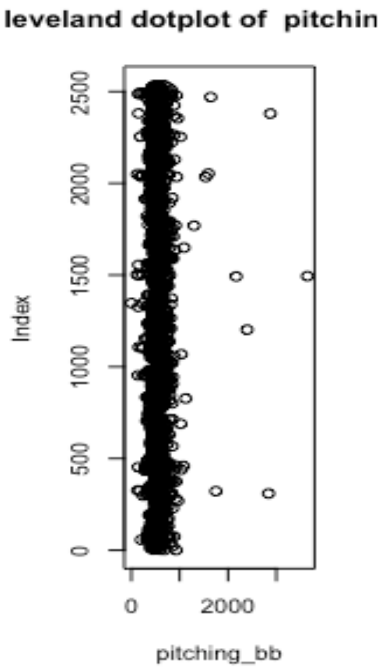

Boxplot of pitching_h
0
5000
10000
15000
20000
25000
30000

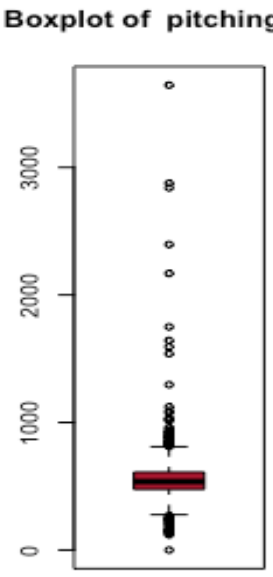

Histogram of pitching_
Frequency
0
500
1000
1500
2000
0
15000
pitching_h

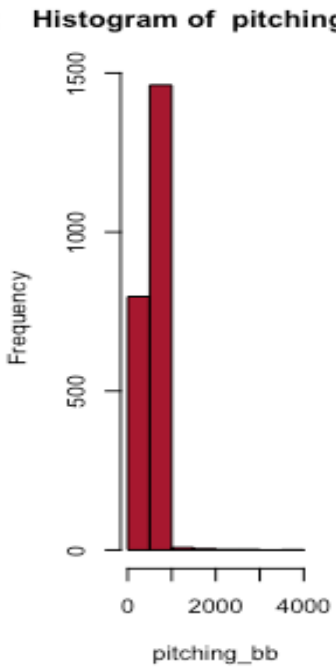

cleveland dotplot of pitchir
Index
0
500
1000
1500
2000
2500
0
100
250
pitching_hr

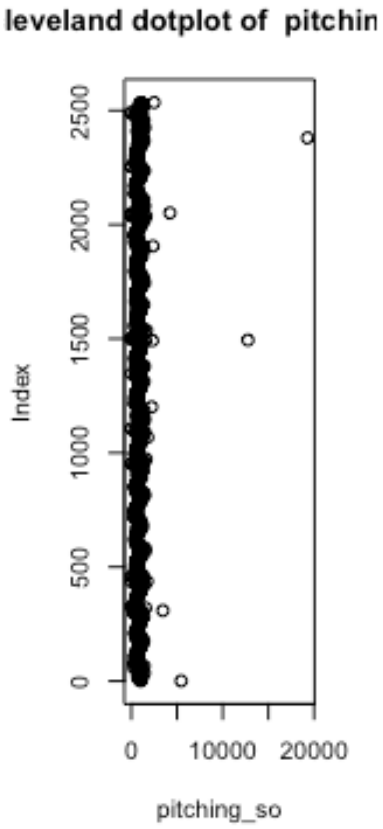

Boxplot of pitching_hr
0
50
100
150
200
250
300
350

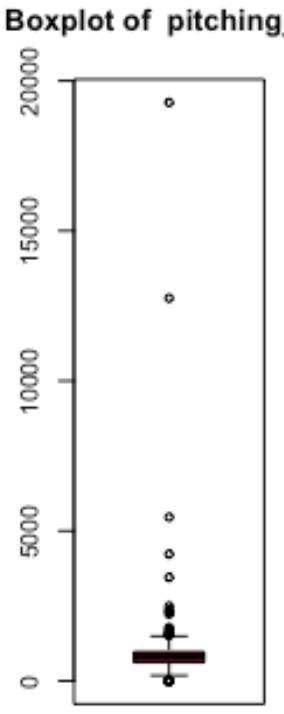

Histogram of pitching_h
Frequency
0
50
100
150
200
250
0
100
250
pitching_hr

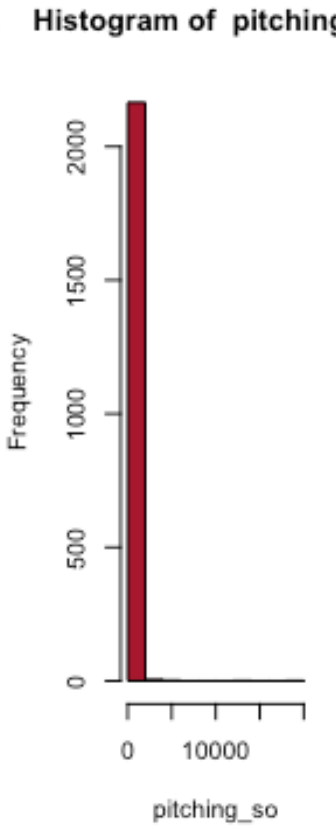

leveland dotplot of pitchin
Index
0
500
1000
1500
2000
2500
0
2000
pitching_bb

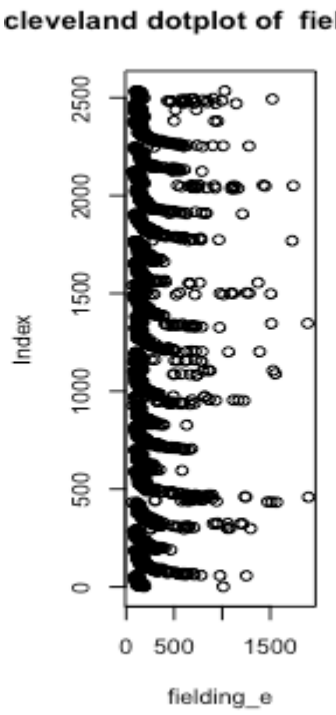

Boxplot of pitching_bb
0
1000
2000
3000

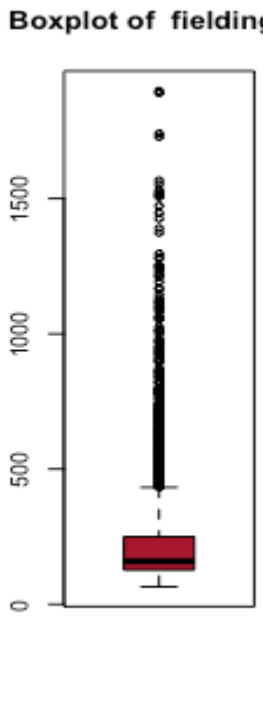

Histogram of pitching_b
Frequency
0
500
1000
1500
0
2000
4000
pitching_bb

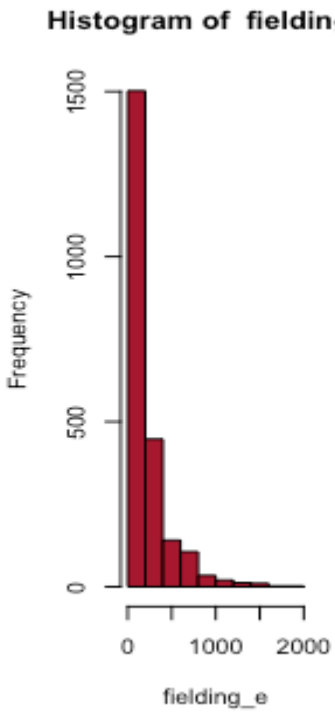

leveland dotplot of pitchin
Index
0
500
1000
1500
2000
2500
0
10000
20000
pitching_so

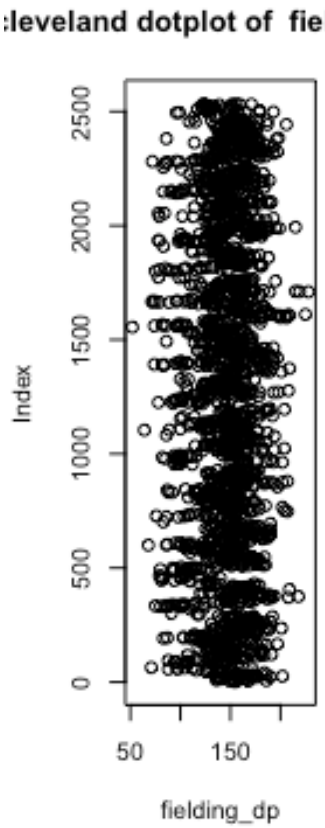

Boxplot of pitching_so
0
5000
10000
15000
20000

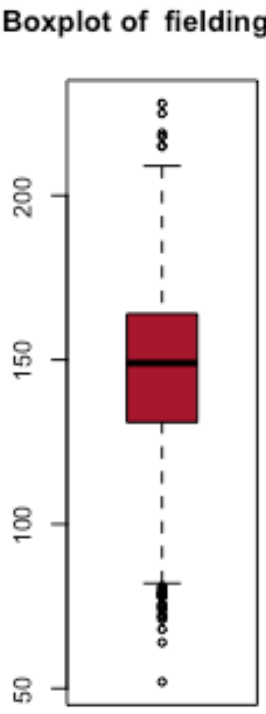

Histogram of pitching_s
Frequency
0
500
1000
1500
2000
0
10000
pitching_so

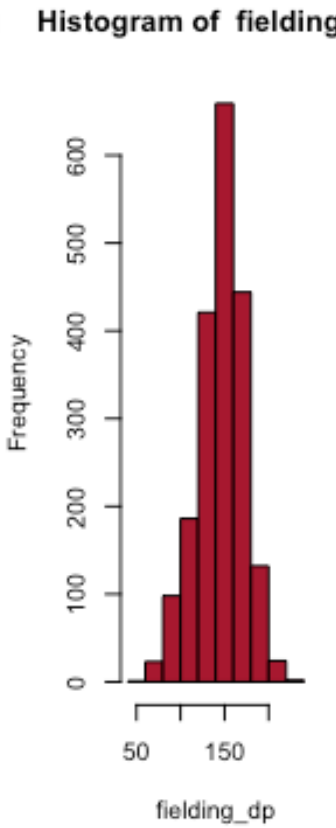

cleveland dotplot of fieldir
Index
0
500
1000
1500
2000
2500
0
500
1500
fielding_e


Boxplot of fielding_e
0
500
1000
1500


Histogram of fielding_
Frequency
0
500
1000
1500
0
1000
2000
fielding_e


cleveland dotplot of fieldin
Index
0
500
1000
1500
2000
2500
50
150
fielding_dp


Boxplot of fielding_dp
50
100
150
200


Histogram of fielding_d
Frequency
0
100
200
300
400
500
600
50
150
fielding_dp

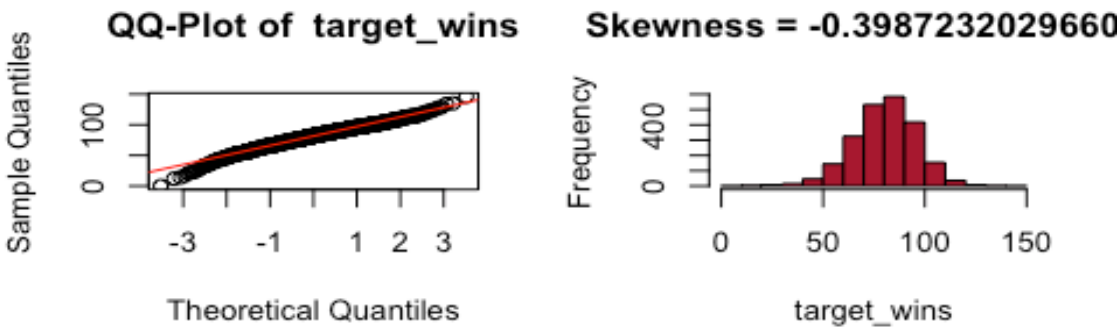
It looks like the outliers are legitmate and we will try Spatial Sign transformation to deal with them.

Now that step one is done, let's look at step 2.

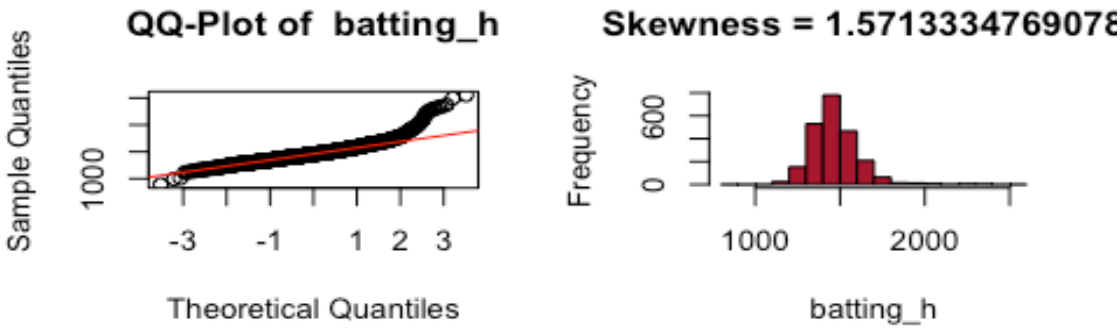
Step 2: Are the data normally distributed?

From the historgram above we can clearly see that the data is not normal, with the exception of some that seems to sort of follow a normal distribution. Let's use QQ-plot to test each column for normality, while adding a histogram and a Skewness number.
- If skewness is less than −1 or greater than +1, the distribution is highly skewed.
- If skewness is between −1 and −½ or between +½ and +1, the distribution is moderately skewed.
- If skewness is between −½ and +½, the distribution is approximately symmetric.

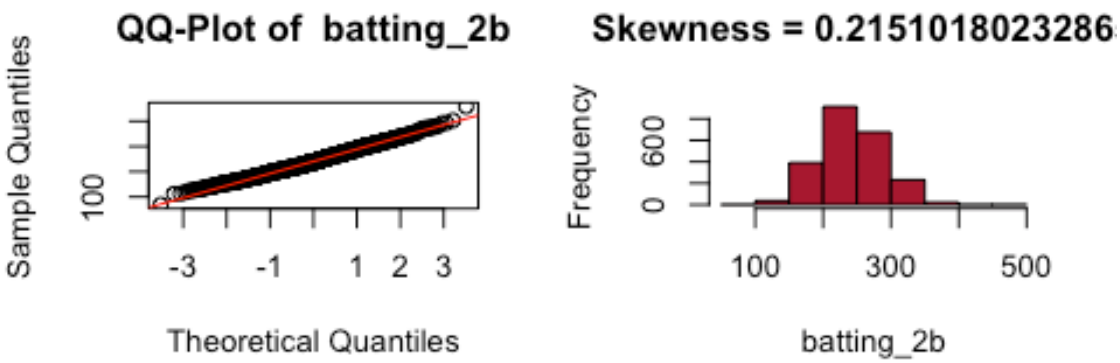
```
par(mfrow = c(2, 2))
i = 2
while (i %in% c(2:17)) {
  qqnorm(moneyball[,i], main = paste("QQ-Plot of
",colnames(moneyball)[i]));qqline(moneyball[,i], col = 2)

  hist(
  moneyball[,i],
  col = "#A71930",
  xlab = colnames(moneyball)[i],
  main = paste0("Skewness = ",skewness(moneyball[,i]))
)

  i = i + 1

}
```


QQ-Plot of target_wins
Sample Quantiles
0 100
-3 -1 1 2 3
Theoretical Quantiles

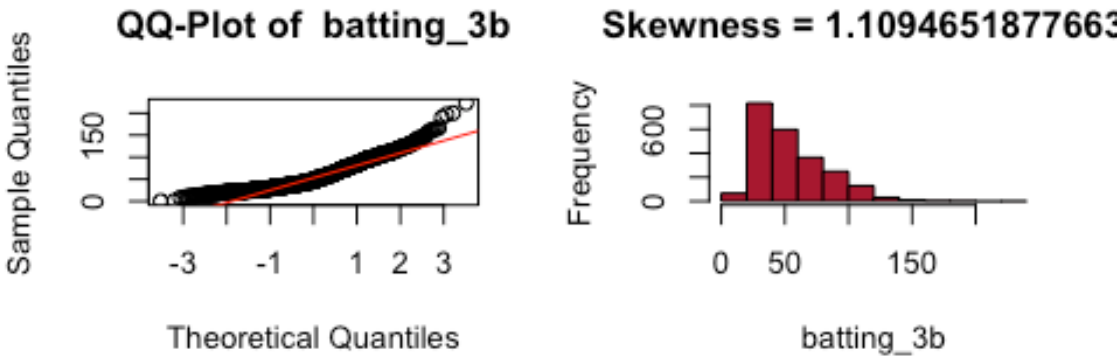
Skewness = -0.3987232029660
Frequency
0 400
0 50 100 150
target_wins


QQ-Plot of batting_h
Sample Quantiles
1000
-3 -1 1 2 3
Theoretical Quantiles

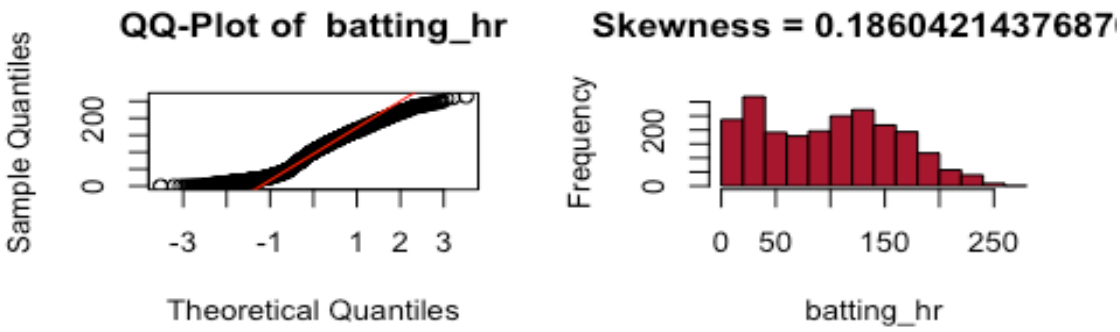
Skewness = 1.5713334769078
Frequency
0 600
1000 2000
batting_h


QQ-Plot of batting_2b
Sample Quantiles
100
-3 -1 1 2 3
Theoretical Quantiles

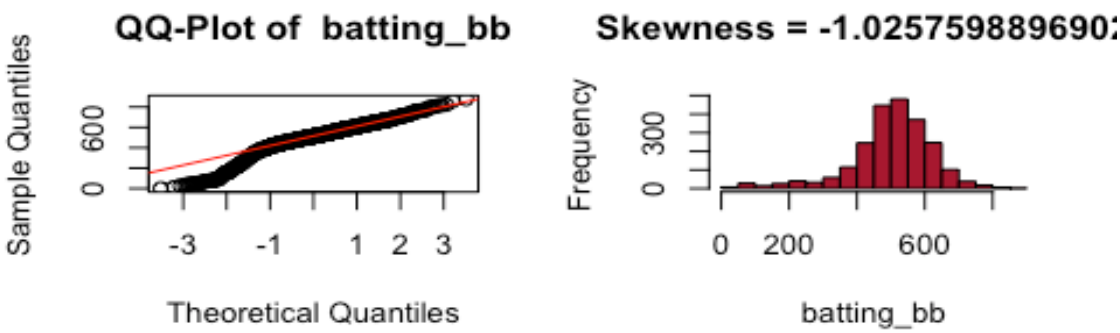
Skewness = 0.2151018023286
Frequency
0 600
100 300 500
batting_2b


QQ-Plot of batting_3b
Sample Quantiles
0 150
-3 -1 1 2 3
Theoretical Quantiles

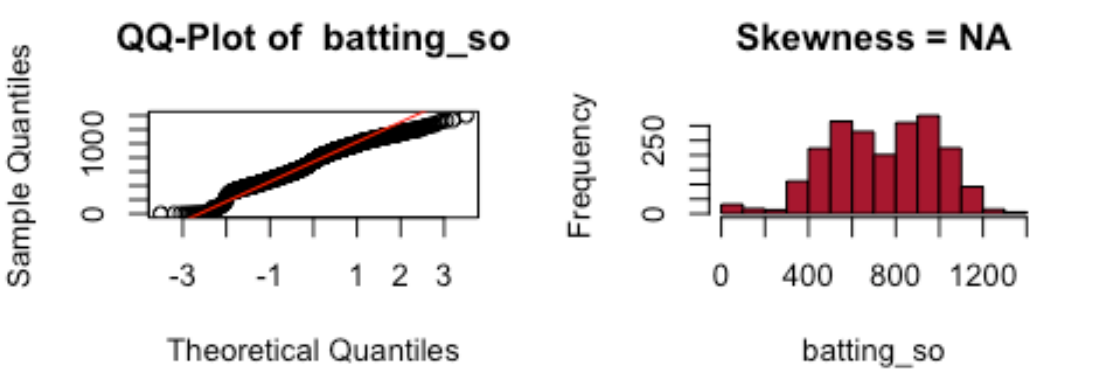
Skewness = 1.1094651877663
Frequency
0 600
0 50 150
batting_3b


QQ-Plot of batting_hr
Sample Quantiles
0
200
-3
-1
1
2
3
Theoretical Quantiles

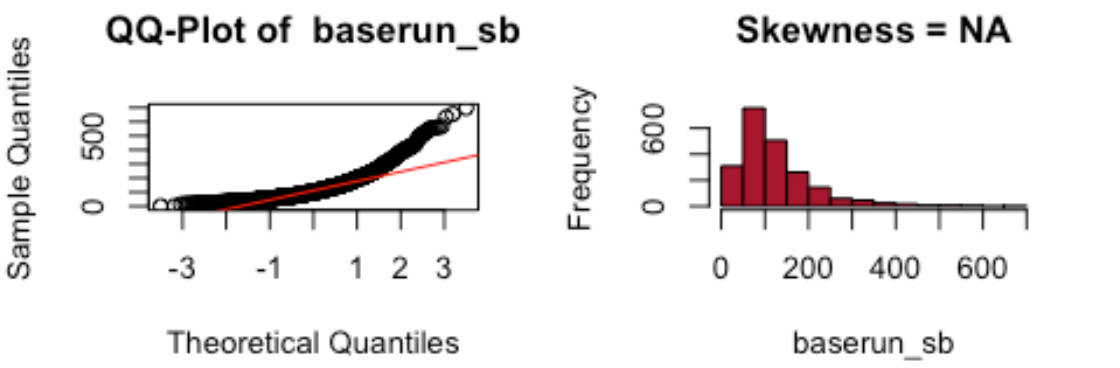
Skewness = 0.1860421437687
Frequency
0
200
0
50
150
250
batting_hr


QQ-Plot of batting_bb
Sample Quantiles
0
600
-3
-1
1
2
3
Theoretical Quantiles

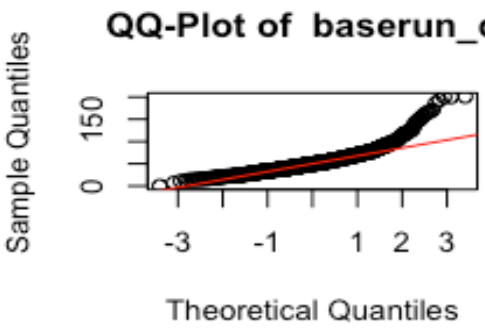
Skewness = -1.025759889690
Frequency
0
300
0
200
600
batting_bb

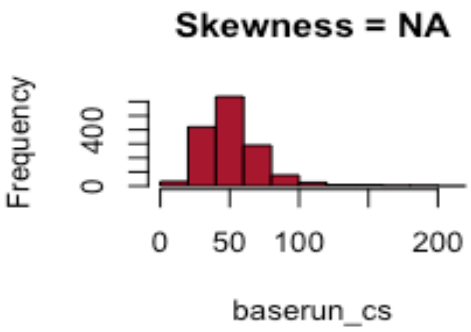

QQ-Plot of batting_so
Sample Quantiles
0
1000
-3
-1
1
2
3
Theoretical Quantiles

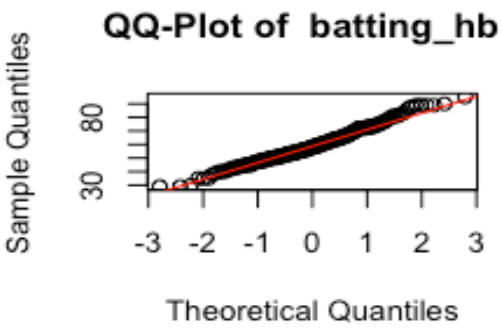
Skewness = NA
Frequency
0
250
0
400
800
1200
batting_so

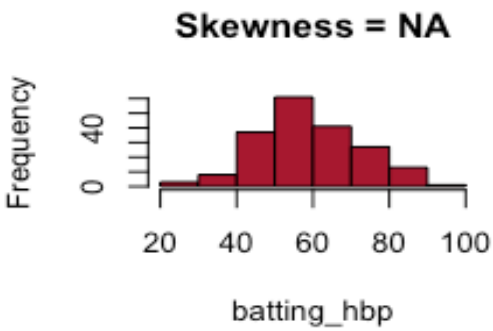

QQ-Plot of baserun_sb
Sample Quantiles
0
500
-3
-1
1
2
3
Theoretical Quantiles

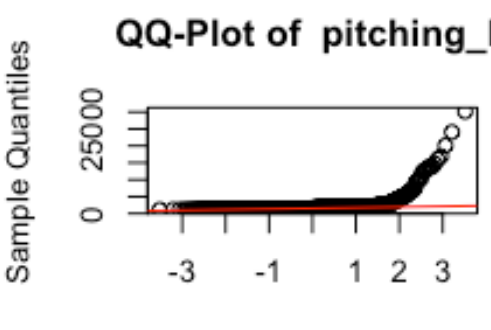
Skewness = NA
Frequency
0
600
0
200
400
600
baserun_sb

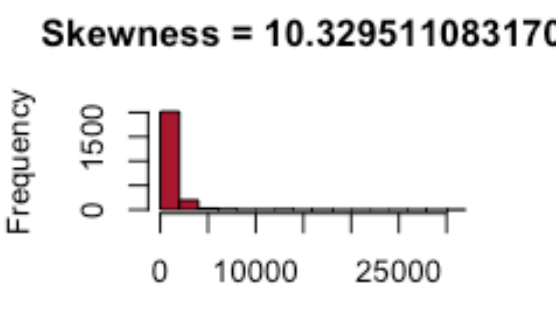

QQ-Plot of baserun_cs
Sample Quantiles
0
150
-3
-1
1
2
3
Theoretical Quantiles

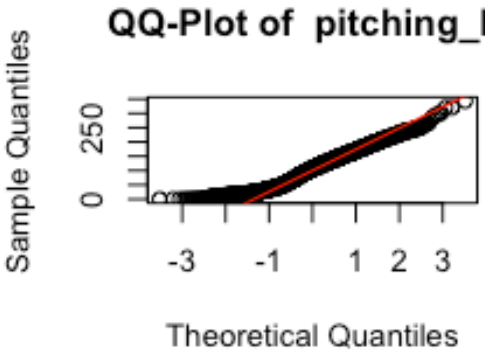

Skewness = NA
Frequency
0
400
0
50
100
200
baserun_cs

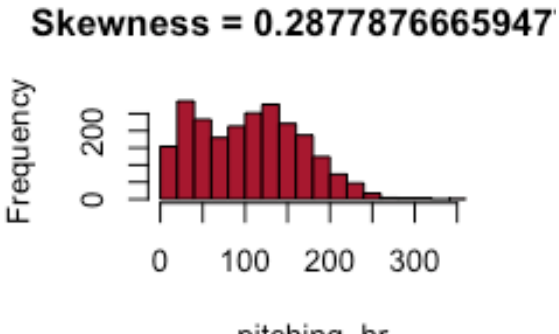

QQ-Plot of batting_hbp
Sample Quantiles
30
80
-3
-2
-1
0
1
2
3
Theoretical Quantiles

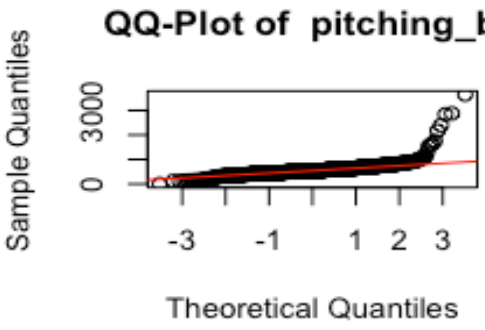

Skewness = NA
Frequency
0
40
20
40
60
80
100
batting_hbp

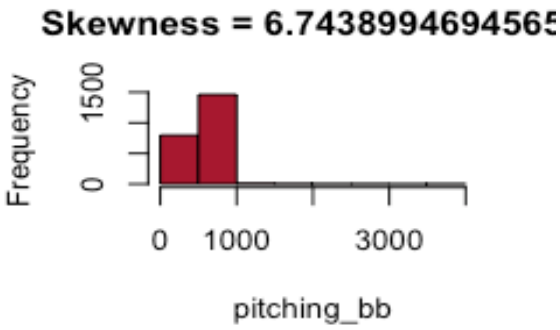

QQ-Plot of pitching_h
Sample Quantiles
0
25000
-3
-1
1
2
3
Theoretical Quantiles

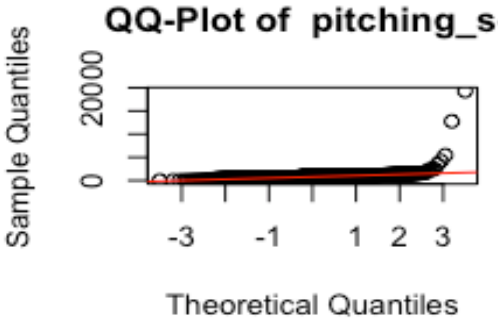

Skewness = 10.329511083170
Frequency
0
1500
0
10000
25000
pitching_h

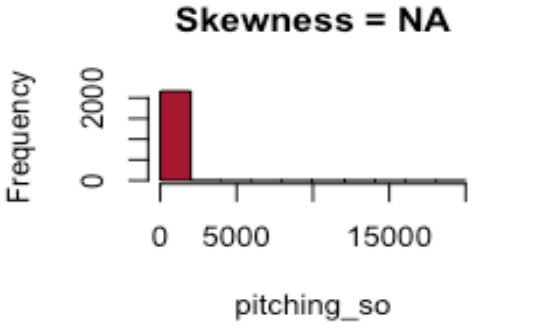

QQ-Plot of pitching_hr
Sample Quantiles
0
250
-3
-1
1
2
3
Theoretical Quantiles

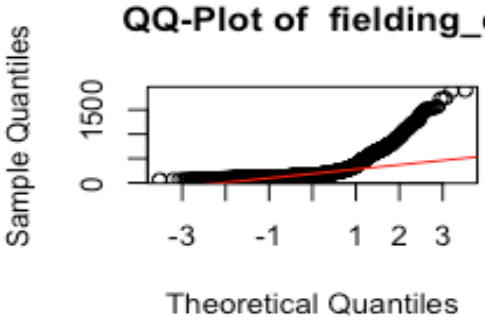

Skewness = 0.2877876665947
Frequency
0
200
0
100
200
300
pitching_hr

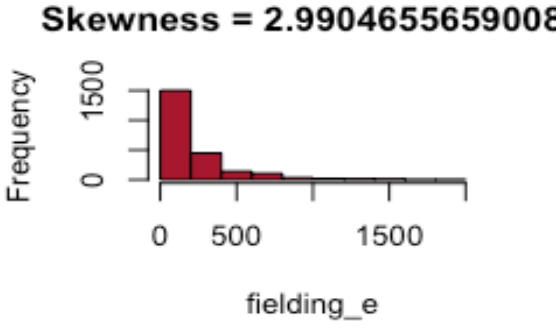

QQ-Plot of pitching_bb
Sample Quantiles
0
3000
-3
-1
1
2
3
Theoretical Quantiles

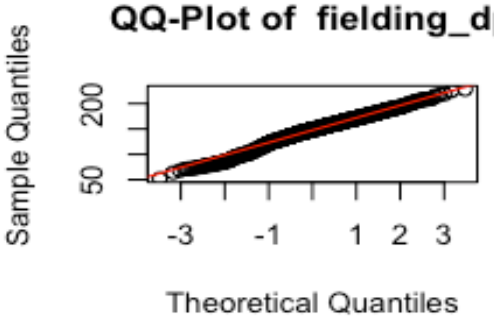

Skewness = 6.7438994694565
Frequency
0
1500
0
1000
3000
pitching_bb

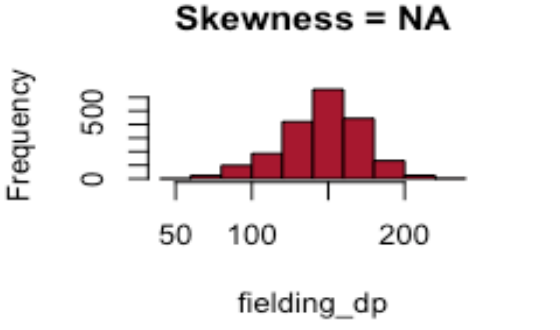

QQ-Plot of pitching_so
Sample Quantiles
0
20000
-3
-1
1
2
3
Theoretical Quantiles


Skewness = NA
Frequency
0
2000
0
5000
15000
pitching_so


QQ-Plot of fielding_e
Sample Quantiles
0
1500
-3
-1
1
2
3
Theoretical Quantiles


Skewness = 2.9904655659008
Frequency
0
1500
0
500
1500
fielding_e


QQ-Plot of fielding_dp
Sample Quantiles
50
200
-3
-1
1
2
3
Theoretical Quantiles


Skewness = NA
Frequency
0
500
50
100
200
fielding_dp

We would need to try certain transformation to correct for Skewness, with Box-Cox being the number one choice.

Step 3: Are there lots of NAs in the data?

R gives us a lot of ways to understand the distribution of `Nulls` within the data. Let's first try to calculate the percentage of Null values to the total number of observation.

```
NAPerc <-
  sapply(moneyball, function(x)
    (sum(is.na(x)) / length(x)) * 100) %>%
  data.frame()
NAPerc$Column <- rownames(NAPerc)
colnames(NAPerc) <- c("NA_Perc", "Col_Name")

# Trying to understand the percentage of NAs per Column
NA_col <- subset(NAPerc, NA_Perc > 0) %>% arrange(desc(NA_Perc))
NA_col
```

```
##     NA_Perc    Col_Name
## 1 91.608084 batting_hbp
## 2 33.919156  baserun_cs
## 3 12.565905 fielding_dp
## 4  5.755712  baserun_sb
## 5  4.481547  batting_so
## 6  4.481547 pitching_so
```

Let's look at the pattern of missing data to try to get more insights. It's clear that batting_hbp is going to be a problematic column with 92% of the data missing. Before we start the imputation or deleting variables, let's try to understand why we have missing data.

Let's use the `mice` package to help us understant how all the NAs behave in the data. `mice` provides a handy function called `md.pattern` that allows one to understand the pattern of missing data. Hopefully by looking at the pattern, we can have an idea on why the data could be missing.

```
md.pattern(moneyball) %>% data.frame()
```

The **first column** of the output shows the number of unique missing data patterns. There are 191 observations with nonmissing values, and there are 1295 observations with nonmissing values except for the variable batting_hbp. The **rightmost column** shows the number of *missing variables* in a particular missing pattern. For example, the first row has no missing value and it is “0” in the row. The **last row** counts the number of missing values for each variable. For example, the variable pitching_bb contains no missing values and the variable batting_so contains 102 missing values. This table can be helpful when you decide to drop some observations with missing variables exceeding a preset threshold.

After careful analysis, the decision is to keep `batting_hbp`. Because I want to transform it into a binary variable, I will keep it out until all th eimputation is done.

```
batting_hbp_bi <- if_else(is.na(moneyball$batting_hbp),0,1)
batting_hbp <- moneyball$batting_hbp
moneyball_trans <- subset(moneyball, select = -c(batting_hbp))
```

Let's impute and treat the data for missing values before testing it for multicollinearity.

The `mice` package will be the package used to help us with this task. Since we only have numeric values, mice will automatically chose PMM (Predictive Mean Matching) as the method. A great resource to understand this techinique is found here.

Let's add `batting_hbp` back into the data.

```
moneyball_imp$batting_hbp <- batting_hbp
moneyball_imp$batting_hbp_bi <- batting_hbp_bi
```

Step 4: Is there collinearity among the covariates?

Let's create a series of correlation matix to understand how each independent variable interacts with the dependent variable. This correlation matix will help us spot any infrigement of the assupmtions needed to develop a robust OLS model, namely multicollinearity. The `caret` package can help the user find those pairs and even suggest which one to remove.

The Caret package offers the `findcorrelation()`, which takes the correlation matrix as an input and finds the fields causing multicollinearity based on a threshold, the `cutoff` parameter. It in turns returns a vector with values that would need to be removed from our dataset due to correlation.

```
colnames(moneyball_imp)[findCorrelation(cor(moneyball_imp))]
```

```
## [1] "batting_hr"
```

Data Transformation

Let's introduce new variables through transformation:

1. `batting_1B = batting_h-(batting_2b + batting_3b + batting_hr)`
2. `free_bases_num = batting_hbp + batting_bb`
3. `total_bases = batting_1B + 2 * batting_2b + 3 * batting_3b + 4 * batting_hr + batting_bb + batting_hbp + baserun_sb`
4. `total_bases_allowed = pitching_bb + 4 * pitching_hr + pitching_h`
5. `HR_over_OP = batting_hr - pitching_hr`
6. `walks_over_OP = batting_bb - pitching_bb`
7. `SO_over_OP = pitching_so - batting_so`

```
moneyball_imp$batting_1B <- moneyball_imp$batting_h-
(moneyball_imp$batting_2b + moneyball_imp$batting_3b +
moneyball_imp$batting_hr)
moneyball_imp$free_bases_num <-
if_else(is.na(moneyball_imp$batting_hbp),0,as.numeric(moneyball_imp$bat
ting_hbp)) + moneyball_imp$batting_bb
moneyball_imp$total_bases <- moneyball_imp$batting_1B + 2 *
moneyball_imp$batting_2b + 3 * moneyball_imp$batting_3b + 4 *
moneyball_imp$batting_hr + moneyball_imp$batting_bb +
if_else(is.na(moneyball_imp$batting_hbp),0,as.numeric(moneyball_imp$bat
ting_hbp)) + moneyball_imp$baserun_sb
moneyball_imp$total_bases_allowed = moneyball_imp$pitching_bb + 4 *
moneyball_imp$pitching_hr + moneyball_imp$pitching_h
moneyball_imp$HR_over_OP = moneyball_imp$batting_hr -
moneyball_imp$pitching_hr
moneyball_imp$walks_over_OP = moneyball_imp$batting_bb -
moneyball_imp$pitching_bb
moneyball_imp$SO_over_OP = moneyball_imp$pitching_so -
moneyball_imp$batting_so
```

Now that we have imputed and created new variables, let's look at the correlation matrix to understand the correlation between the variables and the traget_wins

```
moneyball_imp <- subset(moneyball_imp, select = -c(batting_hbp))
cor(moneyball_imp)
```

Build a Model

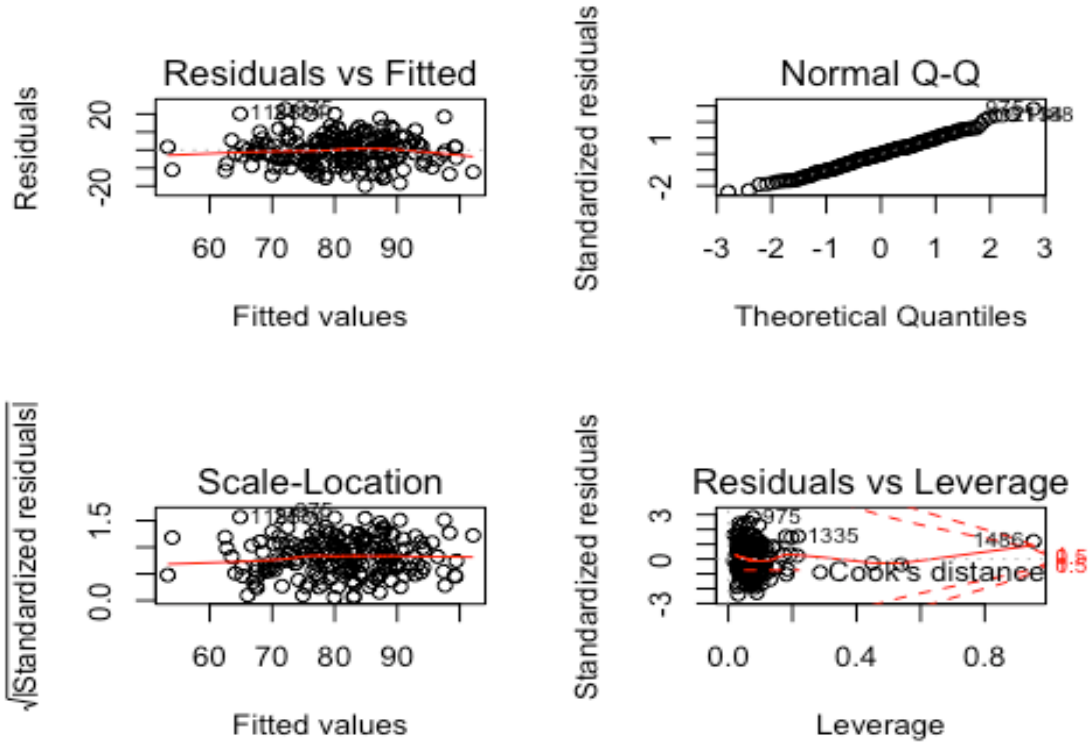
Let's test a model to establish a baseline

```
str(moneyball_imp)
```

```
## 'data.frame':    2276 obs. of  24 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 ...
##  $ batting_h          : int  1445 1339 1377 1387 1297 1279 1244 1273
1391 1271 ...
##  $ batting_2b         : int  194 219 232 209 186 200 179 171 197 213
...
##  $ batting_3b         : int  39 22 35 38 27 36 54 37 40 18 ...
##  $ batting_hr         : int  13 190 137 96 102 92 122 115 114 96 ...
##  $ batting_bb         : int  143 685 602 451 472 443 525 456 447 441
...
##  $ batting_so         : int  842 1075 917 922 920 973 1062 1027 922
827 ...
##  $ baserun_sb         : int  341 37 46 43 49 107 80 40 69 72 ...
##  $ baserun_cs         : int  193 28 27 30 39 59 54 36 27 34 ...
##  $ pitching_h         : int  9364 1347 1377 1396 1297 1279 1244 1281
1391 1271 ...
##  $ pitching_hr        : int  84 191 137 97 102 92 122 116 114 96 ...
##  $ pitching_bb        : int  927 689 602 454 472 443 525 459 447 441
```

```
...
##  $ pitching_so        : int  5456 1082 917 928 920 973 1062 1033 922
827 ...
##  $ fielding_e         : int  1011 193 175 164 138 123 136 112 127
131 ...
##  $ fielding_dp        : int  162 155 153 156 168 149 186 136 169 159
...
##  $ batting_hbp_bi     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ batting_1B         : int  1199 908 973 1044 982 951 889 950 1040
944 ...
##  $ free_bases_num     : num  143 685 602 451 472 443 525 456 447 441
...
##  $ total_bases        : num  2240 2894 2738 2454 2364 ...
##  $ total_bases_allowed: num  10627 2800 2527 2238 2177 ...
##  $ HR_over_OP         : int  -71 -1 0 -1 0 0 0 -1 0 0 ...
##  $ walks_over_OP      : int  -784 -4 0 -3 0 0 0 -3 0 0 ...
##  $ SO_over_OP         : int  4614 7 0 6 0 0 0 6 0 0 ...
```

```
base_model_all <- lm(target_wins ~ batting_h + batting_2b + batting_3b
+ batting_hr + batting_bb + batting_so + baserun_sb + baserun_cs +
pitching_h + pitching_hr + pitching_bb + pitching_so + fielding_e +
fielding_dp + batting_hbp + batting_hbp_bi + batting_1B +
free_bases_num + total_bases + total_bases_allowed + HR_over_OP +
walks_over_OP + SO_over_OP, data = moneyball_imp)
par(mfrow=c(2,2))
plot(base_model_all)
```



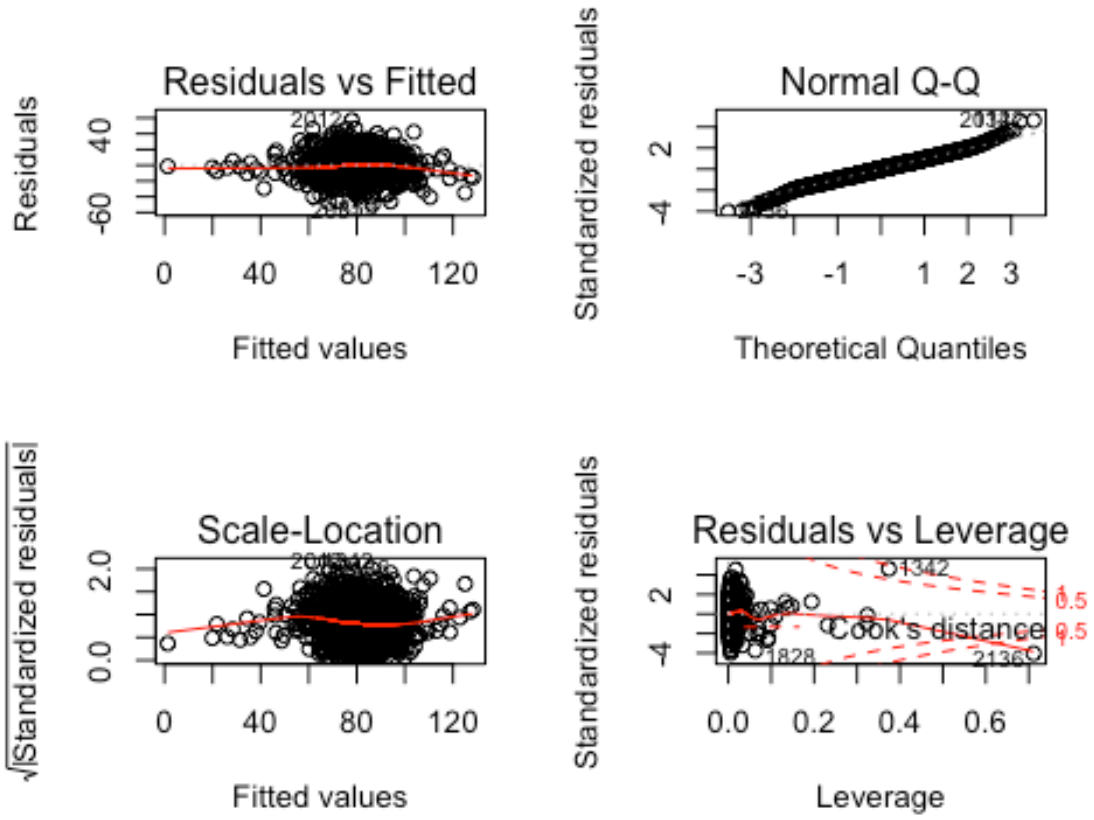
```r
summary(base_model_all)
```

```
##
## Call:
## lm(formula = target_wins ~ batting_h + batting_2b + batting_3b +
##     batting_hr + batting_bb + batting_so + baserun_sb + baserun_cs +
##     pitching_h + pitching_hr + pitching_bb + pitching_so +
fielding_e +
##     fielding_dp + batting_hbp + batting_hbp_bi + batting_1B +
##     free_bases_num + total_bases + total_bases_allowed + HR_over_OP
+
##     walks_over_OP + SO_over_OP, data = moneyball_imp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8708  -5.6564  -0.0599   5.2545  22.9274
##
## Coefficients: (8 not defined because of singularities)
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         60.28826   19.67842   3.064  0.00253 **
## batting_h            1.91348    2.76139   0.693  0.48927
## batting_2b           0.02639    0.03029   0.871  0.38484
## batting_3b          -0.10118    0.07751  -1.305  0.19348
## batting_hr          -4.84371   10.50851  -0.461  0.64542
## batting_bb          -4.45969    3.63624  -1.226  0.22167
## batting_so           0.34196    2.59876   0.132  0.89546
## baserun_sb           0.03304    0.02867   1.152  0.25071
## baserun_cs          -0.01104    0.07143  -0.155  0.87730
## pitching_h          -1.89096    2.76095  -0.685  0.49432
## pitching_hr          4.93043   10.50664   0.469  0.63946
## pitching_bb          4.51089    3.63372   1.241  0.21612
## pitching_so         -0.37364    2.59705  -0.144  0.88577
## fielding_e          -0.17204    0.04140  -4.155 5.08e-05 ***
## fielding_dp         -0.10819    0.03654  -2.961  0.00349 **
## batting_hbp          0.08247    0.04960   1.663  0.09815 .
## batting_hbp_bi            NA         NA      NA       NA
## batting_1B                NA         NA      NA       NA
## free_bases_num            NA         NA      NA       NA
## total_bases               NA         NA      NA       NA
## total_bases_allowed       NA         NA      NA       NA
## HR_over_OP                NA         NA      NA       NA
## walks_over_OP             NA         NA      NA       NA
## SO_over_OP                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.467 on 175 degrees of freedom
##   (2085 observations deleted due to missingness)
## Multiple R-squared:  0.5501, Adjusted R-squared:  0.5116
## F-statistic: 14.27 on 15 and 175 DF,  p-value: < 2.2e-16
```

```
mse <- function(sm)
  mean(sm$residuals^2)

paste('MSE equal ', mse(base_model_all))

## [1] "MSE equal  65.6852879651226"
```

Though R-squared and adjusted R-square is high, we can clearly see that this model dropping observations. Let's try to forget about the new additions, and build a model without them.

```
moneyball_orig <- moneyball_imp[,1:17]
base_model_orig <-
  lm(target_wins ~ batting_h + batting_2b + batting_3b + batting_hr +
batting_bb + batting_so + baserun_sb + baserun_cs + pitching_h +
pitching_hr + pitching_bb + pitching_so + fielding_e + fielding_dp,
data = moneyball_orig)
  par(mfrow = c(2, 2))
  plot(base_model_orig)
```




```
  summary(base_model_orig)

##
## Call:
```

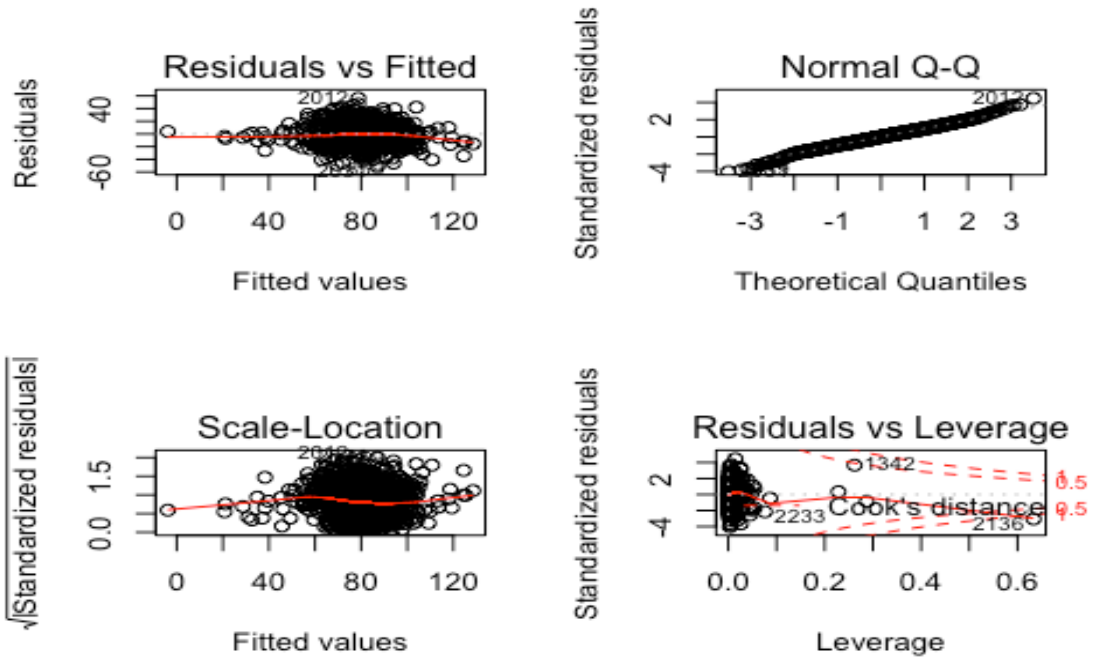
```
## lm(formula = target_wins ~ batting_h + batting_2b + batting_3b +
##     batting_hr + batting_bb + batting_so + baserun_sb + baserun_cs +
##     pitching_h + pitching_hr + pitching_bb + pitching_so +
fielding_e +
##     fielding_dp, data = moneyball_orig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.437  -8.273   0.109   8.115  57.063
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.5833869  5.2232323   6.621 4.44e-11 ***
## batting_h    0.0434011  0.0035801  12.123  < 2e-16 ***
## batting_2b  -0.0203630  0.0089278  -2.281  0.02265 *
## batting_3b   0.0295276  0.0166056   1.778  0.07551 .
## batting_hr   0.0604145  0.0265592   2.275  0.02302 *
## batting_bb   0.0140708  0.0056443   2.493  0.01274 *
## batting_so  -0.0168623  0.0025071  -6.726 2.20e-11 ***
## baserun_sb   0.0529984  0.0052813  10.035  < 2e-16 ***
## baserun_cs  -0.0047414  0.0104140  -0.455  0.64894
## pitching_h   0.0011718  0.0003812   3.074  0.00214 **
## pitching_hr  0.0198220  0.0235832   0.841  0.40071
## pitching_bb -0.0055801  0.0040211  -1.388  0.16536
## pitching_so  0.0026248  0.0008980   2.923  0.00350 **
## fielding_e  -0.0407587  0.0026676 -15.279  < 2e-16 ***
## fielding_dp -0.1067389  0.0130221  -8.197 4.09e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.67 on 2261 degrees of freedom
## Multiple R-squared:  0.3573, Adjusted R-squared:  0.3533
## F-statistic: 89.77 on 14 and 2261 DF,  p-value: < 2.2e-16
```

```
  paste('MSE equal ', mse(base_model_orig))
```

```
## [1] "MSE equal  159.414751654005"
```

This model looks good, from a performance point of view, but when I look at the variance of the residual I don't feel secure.
Let's build another model including only variables with low p-Values.

```
base_model_lp <-
  lm(target_wins ~ batting_h + batting_2b + batting_hr + batting_bb +
batting_so + baserun_sb + pitching_h + pitching_so + fielding_e +
fielding_dp, data = moneyball_orig)
  par(mfrow = c(2, 2))
  plot(base_model_lp)
```




```
  summary(base_model_lp)

##
## Call:
## lm(formula = target_wins ~ batting_h + batting_2b + batting_hr +
##     batting_bb + batting_so + baserun_sb + pitching_h + pitching_so
+
##     fielding_e + fielding_dp, data = moneyball_orig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.044  -8.404   0.170   8.266  56.224
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.7481921  5.0952471   6.623 4.37e-11 ***
## batting_h    0.0458675  0.0033260  13.790  < 2e-16 ***
## batting_2b  -0.0215281  0.0088402  -2.435  0.01496 *
## batting_hr   0.0771546  0.0089649   8.606  < 2e-16 ***
## batting_bb   0.0080930  0.0030417   2.661  0.00785 **
## batting_so  -0.0165511  0.0023941  -6.913 6.13e-12 ***
## baserun_sb   0.0527059  0.0041691  12.642  < 2e-16 ***
## pitching_h   0.0008875  0.0003305   2.686  0.00729 **
## pitching_so  0.0018158  0.0006642   2.734  0.00631 **
## fielding_e  -0.0405882  0.0026636 -15.238  < 2e-16 ***
## fielding_dp -0.1064669  0.0128053  -8.314  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 12.67 on 2265 degrees of freedom
## Multiple R-squared:  0.3557, Adjusted R-squared:  0.3529
## F-statistic: 125.1 on 10 and 2265 DF,  p-value: < 2.2e-16
```

```
  paste('MSE equal ', mse(base_model_lp))
```

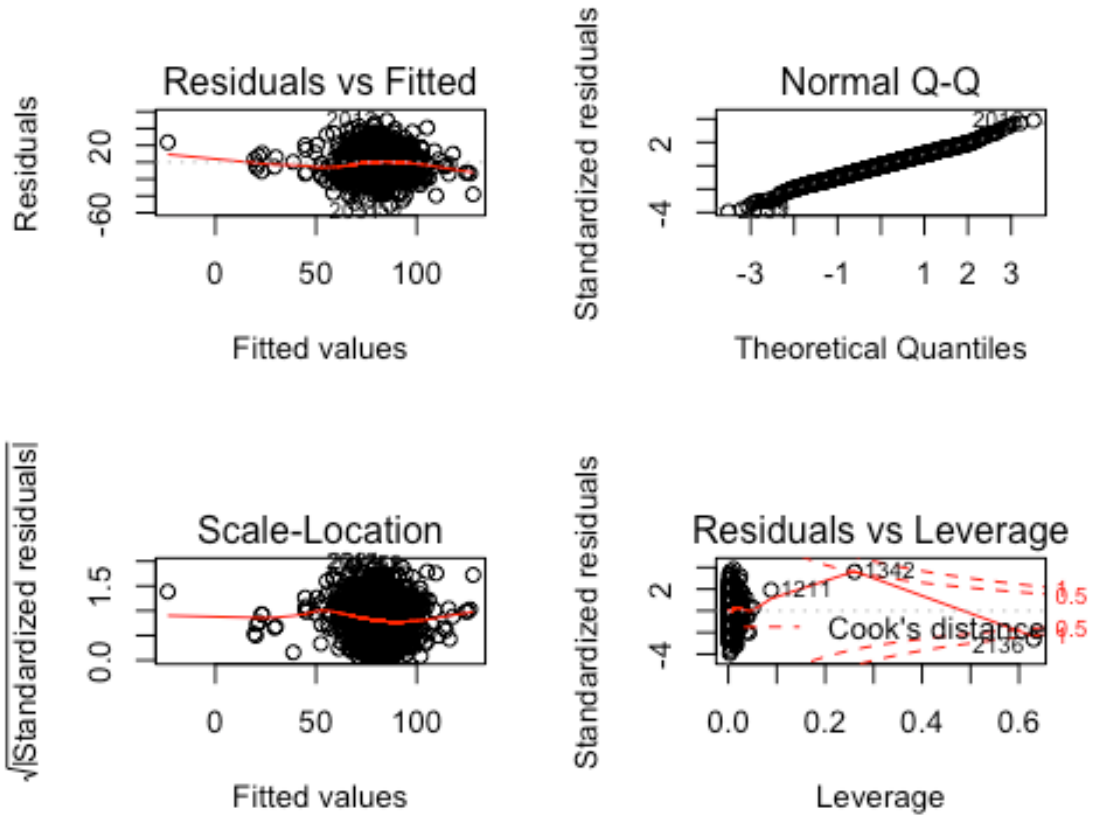
```
## [1] "MSE equal  159.788274102743"
```

Lets remove variables causing multicollinearity using `findCorrelation()`.

```
to_rm <-
colnames(cor(moneyball_imp)[,findCorrelation(cor(moneyball_imp))])
to_rm
```

```
## [1] "batting_hr"     "free_bases_num" "pitching_h"
```

```
base_model_noCol <-
  lm(target_wins ~ batting_h + batting_2b + batting_bb + batting_so +
baserun_sb + pitching_so + fielding_e + fielding_dp, data =
moneyball_orig)
  par(mfrow = c(2, 2))
  plot(base_model_noCol)
```




```
  summary(base_model_noCol)
```

```
##
## Call:
## lm(formula = target_wins ~ batting_h + batting_2b + batting_bb +
##     batting_so + baserun_sb + pitching_so + fielding_e +
fielding_dp,
##     data = moneyball_orig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.696  -8.530   0.266   8.443  49.730
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.9492583  4.3944541   2.036 0.041817 *
## batting_h    0.0574802  0.0031309  18.359  < 2e-16 ***
## batting_2b  -0.0192319  0.0089927  -2.139 0.032573 *
## batting_bb   0.0145786  0.0029924   4.872 1.18e-06 ***
## batting_so  -0.0027687  0.0017242  -1.606 0.108471
## baserun_sb   0.0384923  0.0038631   9.964  < 2e-16 ***
## pitching_so  0.0021644  0.0005971   3.624 0.000296 ***
## fielding_e  -0.0346238  0.0021304 -16.253  < 2e-16 ***
## fielding_dp -0.0829051  0.0126539  -6.552 7.01e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.9 on 2267 degrees of freedom
## Multiple R-squared:  0.3318, Adjusted R-squared:  0.3294
## F-statistic: 140.7 on 8 and 2267 DF,  p-value: < 2.2e-16
```

```
  paste('MSE equal ', mse(base_model_noCol))
```

```
## [1] "MSE equal  165.726715773464"
```

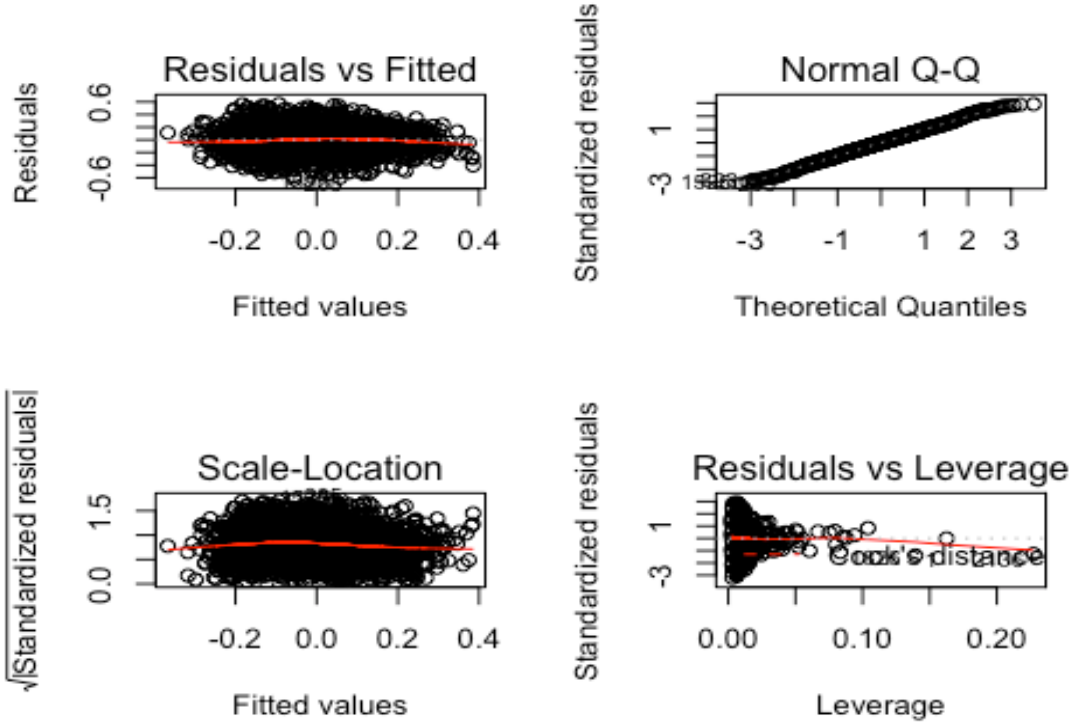
Though the rsquared value went down, there are some improvements on the Cook's distance chart. Now let's try to use use the `caret` package to apply the transformations we discussed earlier in our exploration phase.

1. Center and Scale the data
2. Fix the the problem with outliers by using spatial sign Transformation

3. Last but not least a boxcox transformation to take car of the skewness

```
trans <- preProcess(moneyball_imp, method =
c("center","scale","spatialSign","BoxCox"))
transformed <- predict(trans, moneyball_imp)
head(transformed)

trans_model_all <-
  lm(target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
batting_so + baserun_sb + baserun_cs + pitching_h + pitching_hr +
pitching_bb + pitching_so + fielding_e + fielding_dp + batting_hbp_bi +
```

```
batting_1B + free_bases_num + total_bases + total_bases_allowed +
HR_over_OP + walks_over_OP + SO_over_OP, data = transformed)
  par(mfrow = c(2, 2))
  plot(trans_model_all)
```




```
  summary(trans_model_all)

##
## Call:
## lm(formula = target_wins ~ batting_h + batting_2b + batting_3b +
##     batting_bb + batting_so + baserun_sb + baserun_cs + pitching_h +
##     pitching_hr + pitching_bb + pitching_so + fielding_e +
fielding_dp +
##     batting_hbp_bi + batting_1B + free_bases_num + total_bases +
##     total_bases_allowed + HR_over_OP + walks_over_OP + SO_over_OP,
##     data = transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62430 -0.12992  0.00302  0.12901  0.56473
##
## Coefficients: (2 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.002023   0.004503   0.449 0.653347
## batting_h            -0.157125   0.218126  -0.720 0.471391
## batting_2b           -0.154427   0.092172  -1.675 0.093991 .
```

```
## batting_3b           0.151461   0.082784   1.830 0.067444 .
## batting_bb          -0.465134   0.724456  -0.642 0.520909
## batting_so          -0.550649   0.094236  -5.843 5.86e-09 ***
## baserun_sb           0.062377   0.106454   0.586 0.557968
## baserun_cs          -0.002407   0.037904  -0.064 0.949370
## pitching_h          -0.122984   0.118058  -1.042 0.297650
## pitching_hr         -0.105658   0.208575  -0.507 0.612505
## pitching_bb         -0.367891   0.127904  -2.876 0.004061 **
## pitching_so          0.583123   0.175578   3.321 0.000911 ***
## fielding_e          -0.526787   0.035344 -14.905  < 2e-16 ***
## fielding_dp         -0.174484   0.021579  -8.086 9.96e-16 ***
## batting_hbp_bi      -0.219031   0.098283  -2.229 0.025940 *
## batting_1B           0.133992   0.158747   0.844 0.398725
## free_bases_num       0.660590   0.745334   0.886 0.375550
## total_bases          0.729920   0.309296   2.360 0.018363 *
## total_bases_allowed  0.066528   0.087480   0.760 0.447039
## HR_over_OP          -0.051305   0.072371  -0.709 0.478450
## walks_over_OP              NA         NA      NA       NA
## SO_over_OP                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1948 on 2256 degrees of freedom
## Multiple R-squared:  0.3021, Adjusted R-squared:  0.2963
## F-statistic: 51.41 on 19 and 2256 DF,  p-value: < 2.2e-16
```

```
  paste('MSE equal ', mse(trans_model_all))
```

```
## [1] "MSE equal  0.0376220132488467"
```

Looking at Cook's Distance, it's clear that we have influential data, but the other charts look right where they should be.

Let's try, stepwise approach. 1. Both direction

```
stepwise_base_model_bd <- stepAIC(trans_model_all, direction = "both")
```

```
## Start:  AIC=-7425.66
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + baserun_cs + pitching_h + pitching_hr
+
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
+
##     HR_over_OP + walks_over_OP + SO_over_OP
##
##
## Step:  AIC=-7425.66
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + baserun_cs + pitching_h + pitching_hr
```

```
+
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
+
##     HR_over_OP + walks_over_OP
##
##
## Step:  AIC=-7425.66
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + baserun_cs + pitching_h + pitching_hr
+
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
+
##     HR_over_OP
##
##                        Df Sum of Sq    RSS     AIC
## - baserun_cs            1    0.0002 85.628 -7427.7
## - pitching_hr           1    0.0097 85.637 -7427.4
## - baserun_sb            1    0.0130 85.641 -7427.3
## - batting_bb            1    0.0156 85.643 -7427.2
## - HR_over_OP            1    0.0191 85.647 -7427.2
## - batting_h             1    0.0197 85.647 -7427.1
## - total_bases_allowed   1    0.0220 85.650 -7427.1
## - batting_1B            1    0.0270 85.655 -7426.9
## - free_bases_num        1    0.0298 85.658 -7426.9
## - pitching_h            1    0.0412 85.669 -7426.6
## <none>                             85.628 -7425.7
## - batting_2b            1    0.1065 85.734 -7424.8
## - batting_3b            1    0.1271 85.755 -7424.3
## - batting_hbp_bi        1    0.1885 85.816 -7422.7
## - total_bases           1    0.2114 85.839 -7422.0
## - pitching_bb           1    0.3140 85.942 -7419.3
## - pitching_so           1    0.4187 86.046 -7416.6
## - batting_so            1    1.2960 86.924 -7393.5
## - fielding_dp           1    2.4816 88.109 -7362.6
## - fielding_e            1    8.4317 94.059 -7213.9
##
## Step:  AIC=-7427.65
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + pitching_h + pitching_hr + pitching_bb
+
##     pitching_so + fielding_e + fielding_dp + batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
+
##     HR_over_OP
##
##                        Df Sum of Sq    RSS     AIC
```

```
## - pitching_hr          1    0.0096 85.638 -7429.4
## - baserun_sb           1    0.0130 85.641 -7429.3
## - batting_bb           1    0.0156 85.643 -7429.2
## - HR_over_OP           1    0.0191 85.647 -7429.1
## - batting_h            1    0.0199 85.648 -7429.1
## - total_bases_allowed  1    0.0231 85.651 -7429.0
## - batting_1B           1    0.0277 85.656 -7428.9
## - free_bases_num       1    0.0298 85.658 -7428.9
## - pitching_h           1    0.0423 85.670 -7428.5
## <none>                             85.628 -7427.7
## - batting_2b           1    0.1071 85.735 -7426.8
## - batting_3b           1    0.1274 85.755 -7426.3
## + baserun_cs           1    0.0002 85.628 -7425.7
## - batting_hbp_bi       1    0.1885 85.816 -7424.6
## - total_bases          1    0.2113 85.839 -7424.0
## - pitching_bb          1    0.3149 85.943 -7421.3
## - pitching_so          1    0.4185 86.046 -7418.6
## - batting_so           1    1.2974 86.925 -7395.4
## - fielding_dp          1    2.4847 88.113 -7364.6
## - fielding_e           1    8.4734 94.101 -7214.9
##
## Step:  AIC=-7429.4
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + pitching_h + pitching_bb + pitching_so
+
##     fielding_e + fielding_dp + batting_hbp_bi + batting_1B +
##     free_bases_num + total_bases + total_bases_allowed + HR_over_OP
##
##                        Df Sum of Sq    RSS     AIC
## - HR_over_OP            1    0.0095 85.647 -7431.1
## - batting_h             1    0.0157 85.653 -7431.0
## - batting_bb            1    0.0173 85.655 -7430.9
## - total_bases_allowed   1    0.0284 85.666 -7430.6
## - free_bases_num        1    0.0368 85.674 -7430.4
## - batting_1B            1    0.0575 85.695 -7429.9
## - pitching_h            1    0.0575 85.695 -7429.9
## <none>                             85.638 -7429.4
## + pitching_hr           1    0.0096 85.628 -7427.7
## + baserun_cs            1    0.0001 85.637 -7427.4
## - baserun_sb            1    0.1819 85.819 -7426.6
## - batting_2b            1    0.1856 85.823 -7426.5
## - batting_hbp_bi        1    0.1859 85.823 -7426.5
## - pitching_bb           1    0.3081 85.946 -7423.2
## - pitching_so           1    0.4138 86.051 -7420.4
## - total_bases           1    0.7315 86.369 -7412.0
## - batting_3b            1    0.8476 86.485 -7409.0
## - batting_so            1    1.3026 86.940 -7397.0
## - fielding_dp           1    2.5208 88.158 -7365.4
## - fielding_e            1    8.6348 94.272 -7212.8
##
```

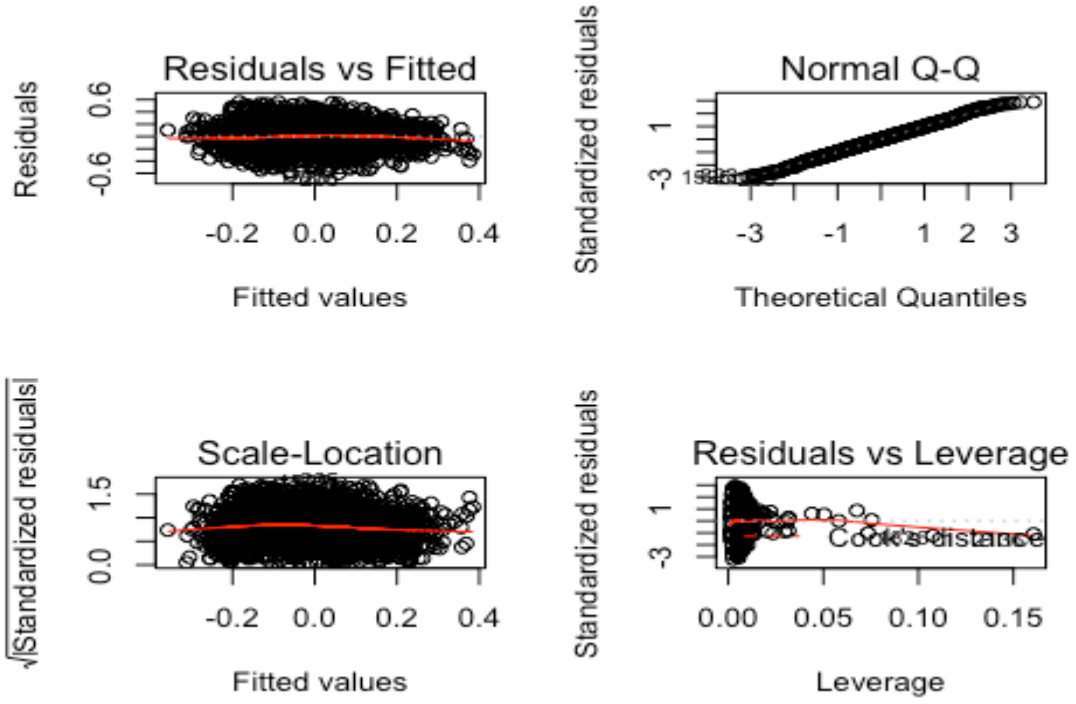
```
## Step:  AIC=-7431.14
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + pitching_h + pitching_bb + pitching_so
+
##     fielding_e + fielding_dp + batting_hbp_bi + batting_1B +
##     free_bases_num + total_bases + total_bases_allowed
##
##                        Df Sum of Sq    RSS     AIC
## - batting_h             1    0.0182 85.665 -7432.7
## - batting_bb            1    0.0183 85.665 -7432.7
## - total_bases_allowed   1    0.0234 85.670 -7432.5
## - free_bases_num        1    0.0361 85.683 -7432.2
## - pitching_h            1    0.0485 85.695 -7431.9
## - batting_1B            1    0.0561 85.703 -7431.7
## <none>                             85.647 -7431.1
## + HR_over_OP            1    0.0095 85.638 -7429.4
## + baserun_cs            1    0.0002 85.647 -7429.1
## + pitching_hr           1    0.0001 85.647 -7429.1
## - baserun_sb            1    0.1724 85.819 -7428.6
## - batting_hbp_bi        1    0.1893 85.836 -7428.1
## - batting_2b            1    0.1951 85.842 -7428.0
## - pitching_bb           1    0.3182 85.965 -7424.7
## - pitching_so           1    0.4212 86.068 -7422.0
## - total_bases           1    0.7882 86.435 -7412.3
## - batting_3b            1    0.8382 86.485 -7411.0
## - batting_so            1    1.3090 86.956 -7398.6
## - fielding_dp           1    2.5113 88.158 -7367.4
## - fielding_e            1    8.8032 94.450 -7210.5
##
## Step:  AIC=-7432.66
## target_wins ~ batting_2b + batting_3b + batting_bb + batting_so +
##     baserun_sb + pitching_h + pitching_bb + pitching_so + fielding_e
+
##     fielding_dp + batting_hbp_bi + batting_1B + free_bases_num +
##     total_bases + total_bases_allowed
##
##                        Df Sum of Sq    RSS     AIC
## - total_bases_allowed   1    0.0156 85.681 -7434.2
## - batting_bb            1    0.0177 85.683 -7434.2
## - free_bases_num        1    0.0382 85.703 -7433.6
## - pitching_h            1    0.0479 85.713 -7433.4
## <none>                             85.665 -7432.7
## - batting_1B            1    0.0930 85.758 -7432.2
## + batting_h             1    0.0182 85.647 -7431.1
## + HR_over_OP            1    0.0121 85.653 -7431.0
## + baserun_cs            1    0.0005 85.665 -7430.7
## + pitching_hr           1    0.0004 85.665 -7430.7
## - batting_hbp_bi        1    0.1914 85.857 -7429.6
## - pitching_bb           1    0.3054 85.971 -7426.6
## - pitching_so           1    0.4470 86.112 -7422.8
```

```
## - baserun_sb           1    0.5464 86.212 -7420.2
## - batting_2b           1    0.6689 86.334 -7417.0
## - batting_3b           1    0.8611 86.526 -7411.9
## - batting_so           1    1.3564 87.022 -7398.9
## - total_bases          1    1.9865 87.652 -7382.5
## - fielding_dp          1    2.5023 88.168 -7369.1
## - fielding_e           1    8.7852 94.450 -7212.5
##
## Step:  AIC=-7434.25
## target_wins ~ batting_2b + batting_3b + batting_bb + batting_so +
##     baserun_sb + pitching_h + pitching_bb + pitching_so + fielding_e
+
##     fielding_dp + batting_hbp_bi + batting_1B + free_bases_num +
##     total_bases
##
##                        Df Sum of Sq    RSS     AIC
## - batting_bb            1    0.0167 85.697 -7435.8
## - pitching_h            1    0.0347 85.716 -7435.3
## - free_bases_num        1    0.0359 85.717 -7435.3
## <none>                              85.681 -7434.2
## - batting_1B            1    0.0846 85.765 -7434.0
## + total_bases_allowed   1    0.0156 85.665 -7432.7
## + batting_h             1    0.0104 85.670 -7432.5
## + HR_over_OP            1    0.0066 85.674 -7432.4
## + baserun_cs            1    0.0015 85.679 -7432.3
## + pitching_hr           1    0.0003 85.681 -7432.3
## - batting_hbp_bi        1    0.1936 85.874 -7431.1
## - pitching_bb           1    0.2926 85.973 -7428.5
## - pitching_so           1    0.4405 86.121 -7424.6
## - baserun_sb            1    0.6221 86.303 -7419.8
## - batting_3b            1    1.1274 86.808 -7406.5
## - batting_2b            1    1.3097 86.991 -7401.7
## - batting_so            1    1.3451 87.026 -7400.8
## - fielding_dp           1    2.4869 88.168 -7371.1
## - total_bases           1    5.3475 91.028 -7298.5
## - fielding_e            1    8.8053 94.486 -7213.6
##
## Step:  AIC=-7435.8
## target_wins ~ batting_2b + batting_3b + batting_so + baserun_sb +
##     pitching_h + pitching_bb + pitching_so + fielding_e +
fielding_dp +
##     batting_hbp_bi + batting_1B + free_bases_num + total_bases
##
##                        Df Sum of Sq    RSS     AIC
## - pitching_h            1    0.0314 85.729 -7437.0
## <none>                              85.697 -7435.8
## - batting_1B            1    0.0813 85.779 -7435.6
## + batting_bb            1    0.0167 85.681 -7434.2
## + walks_over_OP         1    0.0167 85.681 -7434.2
## + total_bases_allowed   1    0.0145 85.683 -7434.2
```

```
## + batting_h             1    0.0101 85.687 -7434.1
## + HR_over_OP            1    0.0075 85.690 -7434.0
## + baserun_cs            1    0.0014 85.696 -7433.8
## + pitching_hr           1    0.0003 85.697 -7433.8
## - free_bases_num        1    0.2108 85.908 -7432.2
## - pitching_bb           1    0.3139 86.011 -7429.5
## - pitching_so           1    0.4504 86.148 -7425.9
## - baserun_sb            1    0.6237 86.321 -7421.3
## - batting_3b            1    1.1231 86.821 -7408.2
## - batting_2b            1    1.3154 87.013 -7403.1
## - batting_so            1    1.3572 87.055 -7402.0
## - batting_hbp_bi        1    1.4961 87.193 -7398.4
## - fielding_dp           1    2.4944 88.192 -7372.5
## - total_bases           1    5.3330 91.030 -7300.4
## - fielding_e            1    8.7892 94.487 -7215.6
##
## Step:  AIC=-7436.97
## target_wins ~ batting_2b + batting_3b + batting_so + baserun_sb +
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases
##
##                        Df Sum of Sq    RSS     AIC
## - batting_1B            1    0.0499 85.779 -7437.6
## <none>                               85.729 -7437.0
## + pitching_h            1    0.0314 85.697 -7435.8
## + batting_h             1    0.0192 85.710 -7435.5
## + batting_bb            1    0.0133 85.716 -7435.3
## + walks_over_OP         1    0.0133 85.716 -7435.3
## + pitching_hr           1    0.0048 85.724 -7435.1
## + total_bases_allowed   1    0.0021 85.727 -7435.0
## + HR_over_OP            1    0.0016 85.727 -7435.0
## + baserun_cs            1    0.0012 85.728 -7435.0
## - pitching_so           1    0.4213 86.150 -7427.8
## - free_bases_num        1    0.4948 86.224 -7425.9
## - pitching_bb           1    0.5811 86.310 -7423.6
## - baserun_sb            1    0.7308 86.460 -7419.7
## - batting_3b            1    1.1052 86.834 -7409.8
## - batting_so            1    1.3953 87.124 -7402.2
## - batting_2b            1    1.5423 87.271 -7398.4
## - batting_hbp_bi        1    1.8417 87.571 -7390.6
## - fielding_dp           1    2.4866 88.216 -7373.9
## - total_bases           1    7.4968 93.226 -7248.2
## - fielding_e            1    9.1865 94.915 -7207.3
##
## Step:  AIC=-7437.65
## target_wins ~ batting_2b + batting_3b + batting_so + baserun_sb +
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     free_bases_num + total_bases
```

```
##
##                         Df Sum of Sq    RSS     AIC
## <none>                                  85.779 -7437.6
## + pitching_hr            1    0.0531 85.726 -7437.1
## + batting_1B             1    0.0499 85.729 -7437.0
## + batting_h              1    0.0335 85.745 -7436.5
## + batting_bb             1    0.0134 85.765 -7436.0
## + walks_over_OP          1    0.0134 85.765 -7436.0
## + total_bases_allowed    1    0.0049 85.774 -7435.8
## + baserun_cs             1    0.0008 85.778 -7435.7
## + HR_over_OP             1    0.0004 85.778 -7435.7
## + pitching_h             1    0.0001 85.779 -7435.6
## - pitching_so            1    0.4145 86.193 -7428.7
## - free_bases_num         1    0.4704 86.249 -7427.2
## - pitching_bb            1    0.5937 86.372 -7423.9
## - baserun_sb             1    0.7480 86.527 -7419.9
## - batting_3b             1    1.0992 86.878 -7410.7
## - batting_so             1    1.5816 87.360 -7398.1
## - batting_2b             1    1.6217 87.400 -7397.0
## - batting_hbp_bi         1    1.9217 87.700 -7389.2
## - fielding_dp            1    2.4468 88.226 -7375.6
## - fielding_e             1    9.1383 94.917 -7209.2
## - total_bases            1    9.5063 95.285 -7200.4
```

```r
par(mfrow = c(2, 2))
  plot(stepwise_base_model_bd)
```



```
  summary(stepwise_base_model_bd)
```

```
##
## Call:
## lm(formula = target_wins ~ batting_2b + batting_3b + batting_so +
##     baserun_sb + pitching_bb + pitching_so + fielding_e +
fielding_dp +
##     batting_hbp_bi + free_bases_num + total_bases, data =
transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62069 -0.12940  0.00108  0.13000  0.56085
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.001382   0.004415   0.313 0.754362
## batting_2b     -0.169401   0.025893  -6.542 7.46e-11 ***
## batting_3b      0.160093   0.029723   5.386 7.94e-08 ***
## batting_so     -0.548829   0.084946  -6.461 1.27e-10 ***
## baserun_sb      0.121656   0.027380   4.443 9.29e-06 ***
## pitching_bb    -0.372319   0.094056  -3.958 7.78e-05 ***
## pitching_so     0.541114   0.163607   3.307 0.000956 ***
## fielding_e     -0.527065   0.033938 -15.530  < 2e-16 ***
## fielding_dp    -0.171256   0.021311  -8.036 1.48e-15 ***
## batting_hbp_bi -0.169102   0.023744  -7.122 1.42e-12 ***
## free_bases_num  0.276863   0.078575   3.524 0.000434 ***
## total_bases     0.550855   0.034776  15.840  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1946 on 2264 degrees of freedom
## Multiple R-squared:  0.3009, Adjusted R-squared:  0.2975
## F-statistic: 88.59 on 11 and 2264 DF,  p-value: < 2.2e-16
```

```
paste('MSE equal ', mse(stepwise_base_model_bd))
```

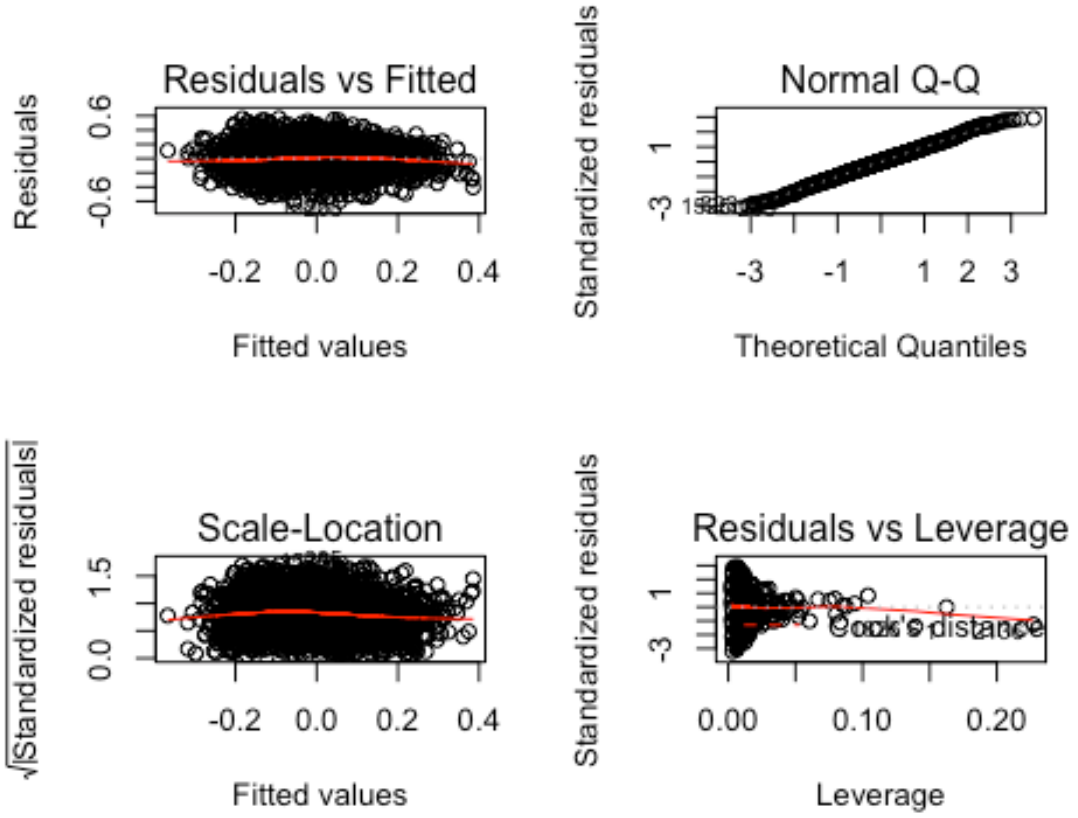
```
## [1] "MSE equal  0.0376883812165135"
```

2. Forward direction

```
stepwise_base_model_fw <- stepAIC(trans_model_all, direction =
"forward")
```

```
## Start:  AIC=-7425.66
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + baserun_cs + pitching_h + pitching_hr
+
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
```

```
+
##     HR_over_OP + walks_over_OP + SO_over_OP

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




```
  summary(stepwise_base_model_fw)

##
## Call:
## lm(formula = target_wins ~ batting_h + batting_2b + batting_3b +
##     batting_bb + batting_so + baserun_sb + baserun_cs + pitching_h +
##     pitching_hr + pitching_bb + pitching_so + fielding_e +
fielding_dp +
##     batting_hbp_bi + batting_1B + free_bases_num + total_bases +
##     total_bases_allowed + HR_over_OP + walks_over_OP + SO_over_OP,
##     data = transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62430 -0.12992  0.00302  0.12901  0.56473
##
## Coefficients: (2 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          0.002023   0.004503   0.449 0.653347
## batting_h           -0.157125   0.218126  -0.720 0.471391
## batting_2b          -0.154427   0.092172  -1.675 0.093991 .
## batting_3b           0.151461   0.082784   1.830 0.067444 .
## batting_bb          -0.465134   0.724456  -0.642 0.520909
## batting_so          -0.550649   0.094236  -5.843 5.86e-09 ***
## baserun_sb           0.062377   0.106454   0.586 0.557968
## baserun_cs          -0.002407   0.037904  -0.064 0.949370
## pitching_h          -0.122984   0.118058  -1.042 0.297650
## pitching_hr         -0.105658   0.208575  -0.507 0.612505
## pitching_bb         -0.367891   0.127904  -2.876 0.004061 **
## pitching_so          0.583123   0.175578   3.321 0.000911 ***
## fielding_e          -0.526787   0.035344 -14.905  < 2e-16 ***
## fielding_dp         -0.174484   0.021579  -8.086 9.96e-16 ***
## batting_hbp_bi      -0.219031   0.098283  -2.229 0.025940 *
## batting_1B           0.133992   0.158747   0.844 0.398725
## free_bases_num       0.660590   0.745334   0.886 0.375550
## total_bases          0.729920   0.309296   2.360 0.018363 *
## total_bases_allowed  0.066528   0.087480   0.760 0.447039
## HR_over_OP          -0.051305   0.072371  -0.709 0.478450
## walks_over_OP              NA         NA      NA       NA
## SO_over_OP                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1948 on 2256 degrees of freedom
## Multiple R-squared:  0.3021, Adjusted R-squared:  0.2963
## F-statistic: 51.41 on 19 and 2256 DF,  p-value: < 2.2e-16
```

```
paste('MSE equal ', mse(stepwise_base_model_fw))
```

```
## [1] "MSE equal  0.0376220132488467"
```

3. Backwards direction

```
stepwise_base_model_bw <- stepAIC(trans_model_all, direction =
"backward")
```

```
## Start:  AIC=-7425.66
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + baserun_cs + pitching_h + pitching_hr
+
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
+
##     HR_over_OP + walks_over_OP + SO_over_OP
##
##
## Step:  AIC=-7425.66
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
```

```
##     batting_so + baserun_sb + baserun_cs + pitching_h + pitching_hr
+
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
+
##     HR_over_OP + walks_over_OP
##
##
## Step:  AIC=-7425.66
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + baserun_cs + pitching_h + pitching_hr
+
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
+
##     HR_over_OP
##
##                        Df Sum of Sq    RSS     AIC
## - baserun_cs            1    0.0002 85.628 -7427.7
## - pitching_hr           1    0.0097 85.637 -7427.4
## - baserun_sb            1    0.0130 85.641 -7427.3
## - batting_bb            1    0.0156 85.643 -7427.2
## - HR_over_OP            1    0.0191 85.647 -7427.2
## - batting_h             1    0.0197 85.647 -7427.1
## - total_bases_allowed   1    0.0220 85.650 -7427.1
## - batting_1B            1    0.0270 85.655 -7426.9
## - free_bases_num        1    0.0298 85.658 -7426.9
## - pitching_h            1    0.0412 85.669 -7426.6
## <none>                             85.628 -7425.7
## - batting_2b            1    0.1065 85.734 -7424.8
## - batting_3b            1    0.1271 85.755 -7424.3
## - batting_hbp_bi        1    0.1885 85.816 -7422.7
## - total_bases           1    0.2114 85.839 -7422.0
## - pitching_bb           1    0.3140 85.942 -7419.3
## - pitching_so           1    0.4187 86.046 -7416.6
## - batting_so            1    1.2960 86.924 -7393.5
## - fielding_dp           1    2.4816 88.109 -7362.6
## - fielding_e            1    8.4317 94.059 -7213.9
##
## Step:  AIC=-7427.65
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + pitching_h + pitching_hr + pitching_bb
+
##     pitching_so + fielding_e + fielding_dp + batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases + total_bases_allowed
+
##     HR_over_OP
##
```

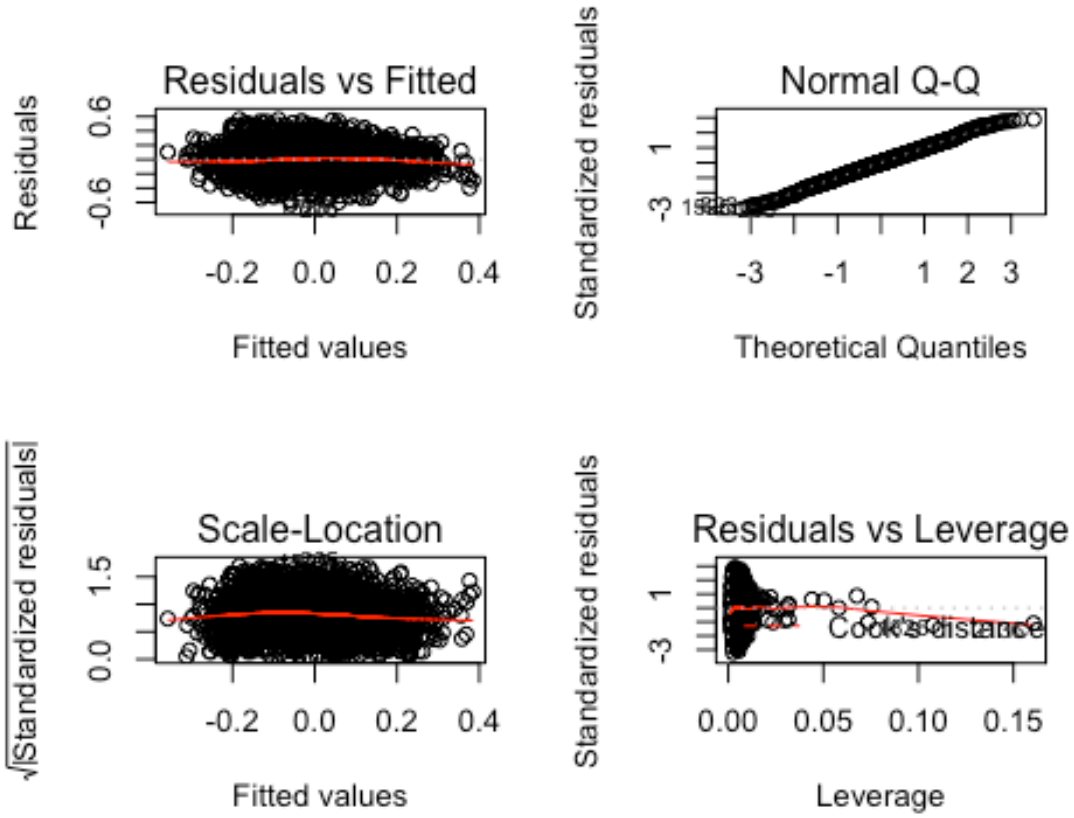
```
##                         Df Sum of Sq    RSS     AIC
## - pitching_hr            1    0.0096 85.638 -7429.4
## - baserun_sb             1    0.0130 85.641 -7429.3
## - batting_bb             1    0.0156 85.643 -7429.2
## - HR_over_OP             1    0.0191 85.647 -7429.1
## - batting_h              1    0.0199 85.648 -7429.1
## - total_bases_allowed    1    0.0231 85.651 -7429.0
## - batting_1B             1    0.0277 85.656 -7428.9
## - free_bases_num         1    0.0298 85.658 -7428.9
## - pitching_h             1    0.0423 85.670 -7428.5
## <none>                             85.628 -7427.7
## - batting_2b             1    0.1071 85.735 -7426.8
## - batting_3b             1    0.1274 85.755 -7426.3
## - batting_hbp_bi         1    0.1885 85.816 -7424.6
## - total_bases            1    0.2113 85.839 -7424.0
## - pitching_bb            1    0.3149 85.943 -7421.3
## - pitching_so            1    0.4185 86.046 -7418.6
## - batting_so             1    1.2974 86.925 -7395.4
## - fielding_dp            1    2.4847 88.113 -7364.6
## - fielding_e             1    8.4734 94.101 -7214.9
##
## Step:  AIC=-7429.4
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
##     batting_so + baserun_sb + pitching_h + pitching_bb + pitching_so
+
##     fielding_e + fielding_dp + batting_hbp_bi + batting_1B +
##     free_bases_num + total_bases + total_bases_allowed + HR_over_OP
##
##                         Df Sum of Sq    RSS     AIC
## - HR_over_OP             1    0.0095 85.647 -7431.1
## - batting_h              1    0.0157 85.653 -7431.0
## - batting_bb             1    0.0173 85.655 -7430.9
## - total_bases_allowed    1    0.0284 85.666 -7430.6
## - free_bases_num         1    0.0368 85.674 -7430.4
## - batting_1B             1    0.0575 85.695 -7429.9
## - pitching_h             1    0.0575 85.695 -7429.9
## <none>                             85.638 -7429.4
## - baserun_sb             1    0.1819 85.819 -7426.6
## - batting_2b             1    0.1856 85.823 -7426.5
## - batting_hbp_bi         1    0.1859 85.823 -7426.5
## - pitching_bb            1    0.3081 85.946 -7423.2
## - pitching_so            1    0.4138 86.051 -7420.4
## - total_bases            1    0.7315 86.369 -7412.0
## - batting_3b             1    0.8476 86.485 -7409.0
## - batting_so             1    1.3026 86.940 -7397.0
## - fielding_dp            1    2.5208 88.158 -7365.4
## - fielding_e             1    8.6348 94.272 -7212.8
##
## Step:  AIC=-7431.14
## target_wins ~ batting_h + batting_2b + batting_3b + batting_bb +
```

```
##     batting_so + baserun_sb + pitching_h + pitching_bb + pitching_so
+
##     fielding_e + fielding_dp + batting_hbp_bi + batting_1B +
##     free_bases_num + total_bases + total_bases_allowed
##
##                        Df Sum of Sq    RSS     AIC
## - batting_h             1    0.0182 85.665 -7432.7
## - batting_bb            1    0.0183 85.665 -7432.7
## - total_bases_allowed   1    0.0234 85.670 -7432.5
## - free_bases_num        1    0.0361 85.683 -7432.2
## - pitching_h            1    0.0485 85.695 -7431.9
## - batting_1B            1    0.0561 85.703 -7431.7
## <none>                             85.647 -7431.1
## - baserun_sb            1    0.1724 85.819 -7428.6
## - batting_hbp_bi        1    0.1893 85.836 -7428.1
## - batting_2b            1    0.1951 85.842 -7428.0
## - pitching_bb           1    0.3182 85.965 -7424.7
## - pitching_so           1    0.4212 86.068 -7422.0
## - total_bases           1    0.7882 86.435 -7412.3
## - batting_3b            1    0.8382 86.485 -7411.0
## - batting_so            1    1.3090 86.956 -7398.6
## - fielding_dp           1    2.5113 88.158 -7367.4
## - fielding_e            1    8.8032 94.450 -7210.5
##
## Step:  AIC=-7432.66
## target_wins ~ batting_2b + batting_3b + batting_bb + batting_so +
##     baserun_sb + pitching_h + pitching_bb + pitching_so + fielding_e
+
##     fielding_dp + batting_hbp_bi + batting_1B + free_bases_num +
##     total_bases + total_bases_allowed
##
##                        Df Sum of Sq    RSS     AIC
## - total_bases_allowed   1    0.0156 85.681 -7434.2
## - batting_bb            1    0.0177 85.683 -7434.2
## - free_bases_num        1    0.0382 85.703 -7433.6
## - pitching_h            1    0.0479 85.713 -7433.4
## <none>                             85.665 -7432.7
## - batting_1B            1    0.0930 85.758 -7432.2
## - batting_hbp_bi        1    0.1914 85.857 -7429.6
## - pitching_bb           1    0.3054 85.971 -7426.6
## - pitching_so           1    0.4470 86.112 -7422.8
## - baserun_sb            1    0.5464 86.212 -7420.2
## - batting_2b            1    0.6689 86.334 -7417.0
## - batting_3b            1    0.8611 86.526 -7411.9
## - batting_so            1    1.3564 87.022 -7398.9
## - total_bases           1    1.9865 87.652 -7382.5
## - fielding_dp           1    2.5023 88.168 -7369.1
## - fielding_e            1    8.7852 94.450 -7212.5
##
## Step:  AIC=-7434.25
```

```
## target_wins ~ batting_2b + batting_3b + batting_bb + batting_so +
##     baserun_sb + pitching_h + pitching_bb + pitching_so + fielding_e
+
##     fielding_dp + batting_hbp_bi + batting_1B + free_bases_num +
##     total_bases
##
##                   Df Sum of Sq    RSS     AIC
## - batting_bb       1    0.0167 85.697 -7435.8
## - pitching_h       1    0.0347 85.716 -7435.3
## - free_bases_num   1    0.0359 85.717 -7435.3
## <none>                         85.681 -7434.2
## - batting_1B       1    0.0846 85.765 -7434.0
## - batting_hbp_bi   1    0.1936 85.874 -7431.1
## - pitching_bb      1    0.2926 85.973 -7428.5
## - pitching_so      1    0.4405 86.121 -7424.6
## - baserun_sb       1    0.6221 86.303 -7419.8
## - batting_3b       1    1.1274 86.808 -7406.5
## - batting_2b       1    1.3097 86.991 -7401.7
## - batting_so       1    1.3451 87.026 -7400.8
## - fielding_dp      1    2.4869 88.168 -7371.1
## - total_bases      1    5.3475 91.028 -7298.5
## - fielding_e       1    8.8053 94.486 -7213.6
##
## Step:  AIC=-7435.8
## target_wins ~ batting_2b + batting_3b + batting_so + baserun_sb +
##     pitching_h + pitching_bb + pitching_so + fielding_e +
fielding_dp +
##     batting_hbp_bi + batting_1B + free_bases_num + total_bases
##
##                   Df Sum of Sq    RSS     AIC
## - pitching_h       1    0.0314 85.729 -7437.0
## <none>                         85.697 -7435.8
## - batting_1B       1    0.0813 85.779 -7435.6
## - free_bases_num   1    0.2108 85.908 -7432.2
## - pitching_bb      1    0.3139 86.011 -7429.5
## - pitching_so      1    0.4504 86.148 -7425.9
## - baserun_sb       1    0.6237 86.321 -7421.3
## - batting_3b       1    1.1231 86.821 -7408.2
## - batting_2b       1    1.3154 87.013 -7403.1
## - batting_so       1    1.3572 87.055 -7402.0
## - batting_hbp_bi   1    1.4961 87.193 -7398.4
## - fielding_dp      1    2.4944 88.192 -7372.5
## - total_bases      1    5.3330 91.030 -7300.4
## - fielding_e       1    8.7892 94.487 -7215.6
##
## Step:  AIC=-7436.97
## target_wins ~ batting_2b + batting_3b + batting_so + baserun_sb +
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     batting_1B + free_bases_num + total_bases
```

```
##
##                   Df Sum of Sq    RSS     AIC
## - batting_1B       1    0.0499 85.779 -7437.6
## <none>                         85.729 -7437.0
## - pitching_so      1    0.4213 86.150 -7427.8
## - free_bases_num   1    0.4948 86.224 -7425.9
## - pitching_bb      1    0.5811 86.310 -7423.6
## - baserun_sb       1    0.7308 86.460 -7419.7
## - batting_3b       1    1.1052 86.834 -7409.8
## - batting_so       1    1.3953 87.124 -7402.2
## - batting_2b       1    1.5423 87.271 -7398.4
## - batting_hbp_bi   1    1.8417 87.571 -7390.6
## - fielding_dp      1    2.4866 88.216 -7373.9
## - total_bases      1    7.4968 93.226 -7248.2
## - fielding_e       1    9.1865 94.915 -7207.3
##
## Step:  AIC=-7437.65
## target_wins ~ batting_2b + batting_3b + batting_so + baserun_sb +
##     pitching_bb + pitching_so + fielding_e + fielding_dp +
batting_hbp_bi +
##     free_bases_num + total_bases
##
##                   Df Sum of Sq    RSS     AIC
## <none>                         85.779 -7437.6
## - pitching_so      1    0.4145 86.193 -7428.7
## - free_bases_num   1    0.4704 86.249 -7427.2
## - pitching_bb      1    0.5937 86.372 -7423.9
## - baserun_sb       1    0.7480 86.527 -7419.9
## - batting_3b       1    1.0992 86.878 -7410.7
## - batting_so       1    1.5816 87.360 -7398.1
## - batting_2b       1    1.6217 87.400 -7397.0
## - batting_hbp_bi   1    1.9217 87.700 -7389.2
## - fielding_dp      1    2.4468 88.226 -7375.6
## - fielding_e       1    9.1383 94.917 -7209.2
## - total_bases      1    9.5063 95.285 -7200.4
```

```r
par(mfrow = c(2, 2))
  plot(stepwise_base_model_bw)
```




```
summary(stepwise_base_model_bw)
```

```
##
## Call:
## lm(formula = target_wins ~ batting_2b + batting_3b +
batting_so +
##     baserun_sb + pitching_bb + pitching_so + fielding_e +
fielding_dp +
##     batting_hbp_bi + free_bases_num + total_bases, data =
transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62069 -0.12940  0.00108  0.13000  0.56085
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.001382   0.004415   0.313 0.754362
## batting_2b     -0.169401   0.025893  -6.542 7.46e-11 ***
## batting_3b      0.160093   0.029723   5.386 7.94e-08 ***
## batting_so     -0.548829   0.084946  -6.461 1.27e-10 ***
## baserun_sb      0.121656   0.027380   4.443 9.29e-06 ***
## pitching_bb    -0.372319   0.094056  -3.958 7.78e-05 ***
```

```
## pitching_so     0.541114   0.163607   3.307 0.000956 ***
## fielding_e     -0.527065   0.033938 -15.530  < 2e-16 ***
## fielding_dp    -0.171256   0.021311  -8.036 1.48e-15 ***
## batting_hbp_bi -0.169102   0.023744  -7.122 1.42e-12 ***
## free_bases_num  0.276863   0.078575   3.524 0.000434 ***
## total_bases     0.550855   0.034776  15.840  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1946 on 2264 degrees of freedom
## Multiple R-squared:  0.3009, Adjusted R-squared:  0.2975
## F-statistic: 88.59 on 11 and 2264 DF,  p-value: < 2.2e-16
```

```
paste('MSE equal ', mse(stepwise_base_model_bw))
```

```
## [1] "MSE equal  0.0376883812165135"
```

Conclusion

It definitely made a difference when the transformation were applied. One can see the difference in the residual plots. The residual is now normal(per QQ plot), and there are no patterns when we look at he Rsiduals Vs Fitted plot. When looking at the Rsquared and Adjusted Rsquared together with the residual plots, it's easy to conclude that the model with the stepwise approach together with the transformations is the one that leads to a better model.

Though RMSE and Rsquared from the other models seem to suggest otherwise, the stepwise model appears to be more stable. I also noticed by looking at the Cook's Distance plot that there are influncial observations, but for some reason I could not get robust regression to work. From my understanding, robust regression would put less enphasis on those data points, leading to a more accurate model.