Money Ball OLS Regression Project Report

By Mimi Trinh

INTRODUCTION

The money ball project starts with the training dataset containing 2276 observations with 17 variables. Each record represents a professional baseball team from the year 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162-game season. The 17 variables included in the dataset are:

- INDEX: identification variable with no use for the predictive model
- TARGET_WINS: the wins for a team over a single season
- TEAM_BATTING_H: base hits by batters (1B, 2B, 3B, HR)
- TEAM_BATTING_2B: doubles by batters (2B)
- TEAM_BATTING_3B: triples by batters (3B)
- TEAM_BATTING_HR: homeruns by batters (4B)
- TEAM_BATTING_BB: walks by batters
- TEAM_BATTING_HBP: batter hit by pitch (get a free base)
- TEAM_BATTING_SO: strikeouts by batters
- TEAM_BASERUN_SB: stolen bases
- TEAM_BASERUN_CS: caught stealing
- TEAM_FIELDING_E: errors
- TEAM_FIELDING_DP: double plays
- TEAM_PITCHING_BB: walks allowed
- TEAM_PITCHING_H: hits allowed
- TEAM_PITCHING_HR: homeruns allowed
- TEAM_PITCHING_SO: strikeouts by pitchers

The fundamental goal of this project is to use the train dataset to identify variables to build an ordinary least square (OLS) regression predictive model that can forecast the wins for a team over a single season. Specifically, the analyst will build this predictive model using the train dataset using TARGET_WINS variable where we know how many wins occurred. Then the analyst will apply this model on the test dataset to provide the results to stakeholders. This report includes four stages of the project 1) data exploration 2) data preparation 3) model development 4) model selection.

RESULTS

Section 1: Data Exploration

```
'data.frame':   2276 obs. of  17 variables:
 $ INDEX           : int  1 2 3 4 5 6 7 8 11 12 ...
 $ TARGET_WINS     : int  39 70 86 70 82 75 80 85 86 76 ...
 $ TEAM_BATTING_H  : int  1445 1339 1377 1387 1297 1279 1244 1273 1391 1271 ...
 $ TEAM_BATTING_2B : int  194 219 232 209 186 200 179 171 197 213 ...
 $ TEAM_BATTING_3B : int  39 22 35 38 27 36 54 37 40 18 ...
 $ TEAM_BATTING_HR : int  13 190 137 96 102 92 122 115 114 96 ...
 $ TEAM_BATTING_BB : int  143 685 602 451 472 443 525 456 447 441 ...
 $ TEAM_BATTING_SO : int  842 1075 917 922 920 973 1062 1027 922 827 ...
 $ TEAM_BASERUN_SB : int  NA 37 46 43 49 107 80 40 69 72 ...
 $ TEAM_BASERUN_CS : int  NA 28 27 30 39 59 54 36 27 34 ...
 $ TEAM_BATTING_HBP: int  NA NA NA NA NA NA NA NA NA NA ...
 $ TEAM_PITCHING_H : int  9364 1347 1377 1396 1297 1279 1244 1281 1391 1271 ...
 $ TEAM_PITCHING_HR: int  84 191 137 97 102 92 122 116 114 96 ...
 $ TEAM_PITCHING_BB: int  927 689 602 454 472 443 525 459 447 441 ...
 $ TEAM_PITCHING_SO: int  5456 1082 917 928 920 973 1062 1033 922 827 ...
 $ TEAM_FIELDING_E : int  1011 193 175 164 138 123 136 112 127 131 ...
 $ TEAM_FIELDING_DP: int  NA 155 153 156 168 149 186 136 169 159 ...
```

Above is an overview of the train dataset used in this project to develop the predictive model. There are 2776 observations and 17 variables in the dataset. All 17 variables are in integer format, which is appropriate for the analysis. No additional work is needed to change the format of the variables. The numbers on the right of the variables show the first couple observations in the dataset, which indicate missing values as "NA" in some variables. This issue will need to be addressed prior to developing the predictive model.

```
          INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
    0.004217721     -0.398986126      1.572369635     0.215243643
 TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
    1.110196784      0.186164822     -1.026436289               NA
 TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H
              NA               NA               NA    10.336322498
TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
    0.287977438      6.748346485               NA     2.992437518
TEAM_FIELDING_DP
              NA
```

One of the assumptions of OLS regression analysis is that all variables are normally distributed. If a variable has skewness number of 0, it means the variable has a perfect normal distribution. From the skewness result above, TEAM_BATTING_H, TEAM_BATTING_3B, TEAM_BATTING_BB, and TEAM_FIELDING_E variables moderately violate the normality assumption whereas TEAM_PITCHING_H and TEAM_PITCHING_BB severely violate the normality assumption. The remaining variables have skewness values between -1 and 1, so they are relatively normally distributed. Variables with NA value in the table above have missing values. Excluding their missing values, below is the skewness number for each of these variables.

- TEAM_BATTING_SO: -0.2980057

- TEAM_BASERUN_SB: 1.973794
- TEAM_BASERUN_CS: 1.978191
- TEAM_BATTING_HBP: 0.3210938
- TEAM_PITCHING_SO: 22.18986
- TEAM_FIELDING_DP: -0.3892324

According to the skewness results above, TEAM_BASERUN_SB and TEAM_BASERUN_CS have moderate normality issue whereas TEAM_PITCHING_SO has severe normality issue. The remaining three variables are relatively normally distributed.

```
     INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
 Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
 1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
 Median :1270.5   Median : 82.00   Median :1454   Median :238.0
 Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
 3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
 Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0

 TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
 Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
 1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0
 Median : 47.00   Median :102.00   Median :512.0   Median : 750.0
 Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 735.6
 3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0
 Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
                                                   NA's   :102
 TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H
 Min.   :  0.0   Min.   :  0.0   Min.   :29.00    Min.   : 1137
 1st Qu.: 66.0   1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419
 Median :101.0   Median : 49.0   Median :58.00    Median : 1518
 Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779
 3rd Qu.:156.0   3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682
 Max.   :697.0   Max.   :201.0   Max.   :95.00    Max.   :30132
 NA's   :131     NA's   :772     NA's   :2085
 TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
 Min.   :  0.0    Min.   :   0.0   Min.   :    0.0   Min.   :  65.0
 1st Qu.: 50.0    1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0
 Median :107.0    Median : 536.5   Median :  813.5   Median : 159.0
 Mean   :105.7    Mean   : 553.0   Mean   :  817.7   Mean   : 246.5
 3rd Qu.:150.0    3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2
 Max.   :343.0    Max.   :3645.0   Max.   :19278.0   Max.   :1898.0
                                   NA's   :102
 TEAM_FIELDING_DP
 Min.   : 52.0
 1st Qu.:131.0
 Median :149.0
 Mean   :146.4
 3rd Qu.:164.0
 Max.   :228.0
 NA's   :286
```

The table above summarizes the key attributes of each variable such as minimum, maximum, median, mean, Q1, Q3, and missing value. From the table above, we identify the following variables as predictors with missing value.

- TEAM_BATTING_SO: 102 missing values
- TEAM_BASERUN_SB: 131
- TEAM_BASERUN_CS: 772
- TEAM_BATTING_HBP: 2085
- TEAM_PITCHING_SO: 102
- TEAM_FIELDING_DP: 286

Among these variables, TEAM_BATTING_HBP has the most missing values, which account for 92% of total 2276 records. Therefore, we may decide to remove this variable from the study due to its lack of usefulness.

In addition, according to the summary table above, TARGET_WINS variable has no negative or more than 162 wins, which means that there's no obvious error in this variable. From speaking to industry expert, we know that it's reasonable to assume a team will get at least 30 wins but no more than 120 wins. However, this is not the case since the result shows a minimum of 0 and maximum of 146 wins.

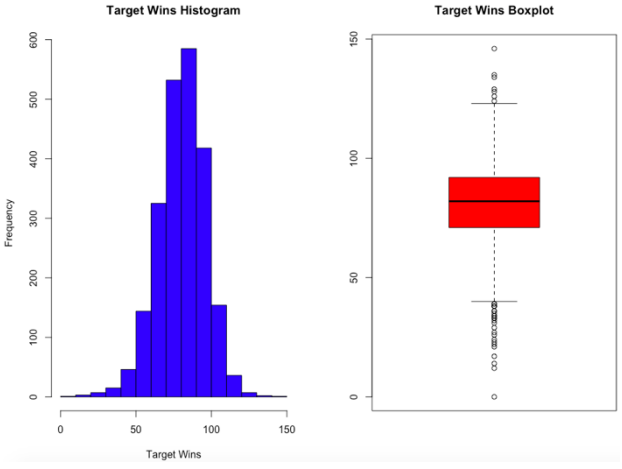
Below is the list of TARGET_WINS below 30

```
23 21 22 17  0 24 27 14 26 12 29
```

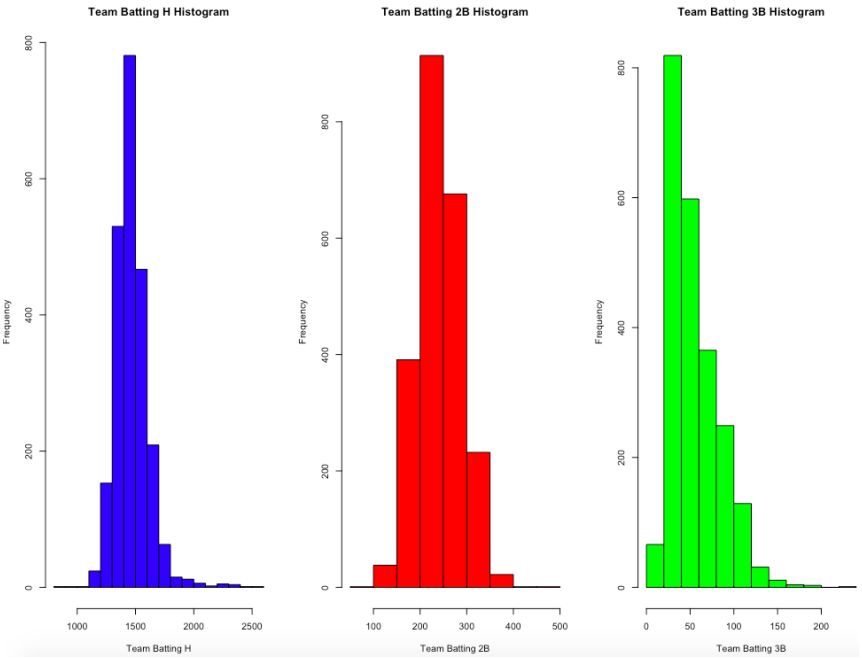
Below is the list of TARGET_WINS above 120

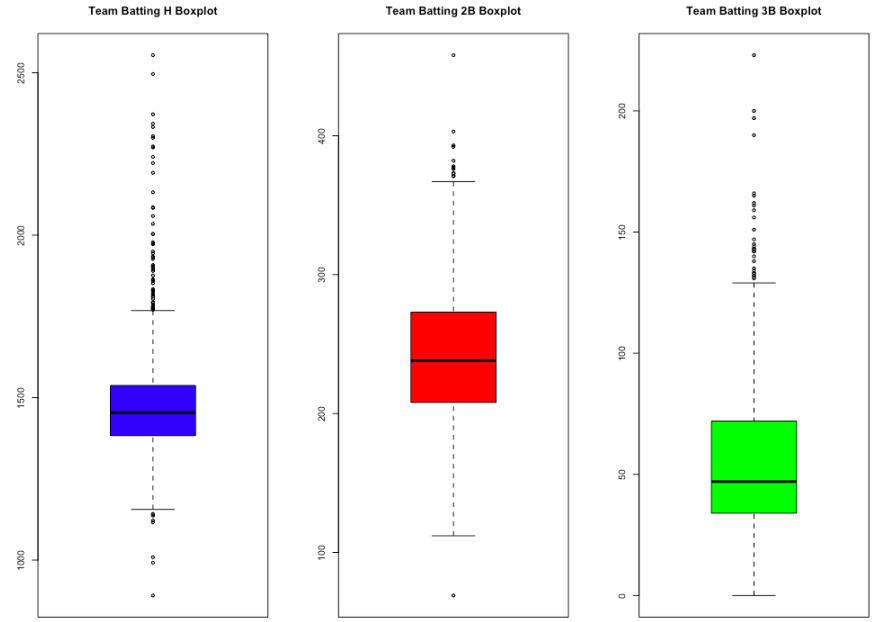
```
134 146 128 129 126 124 122 122 123 135
```

Though these numbers don't follow the expert's reasonable assumption, it's still possible these records are correct. In other words, we can't prove these records are errors, so we will keep them in the dataset.

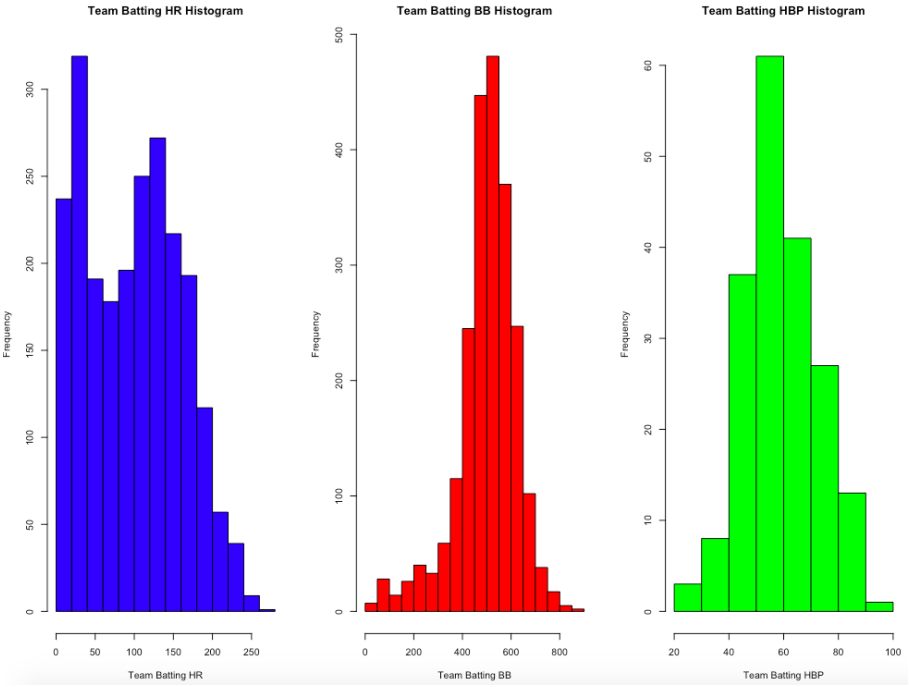


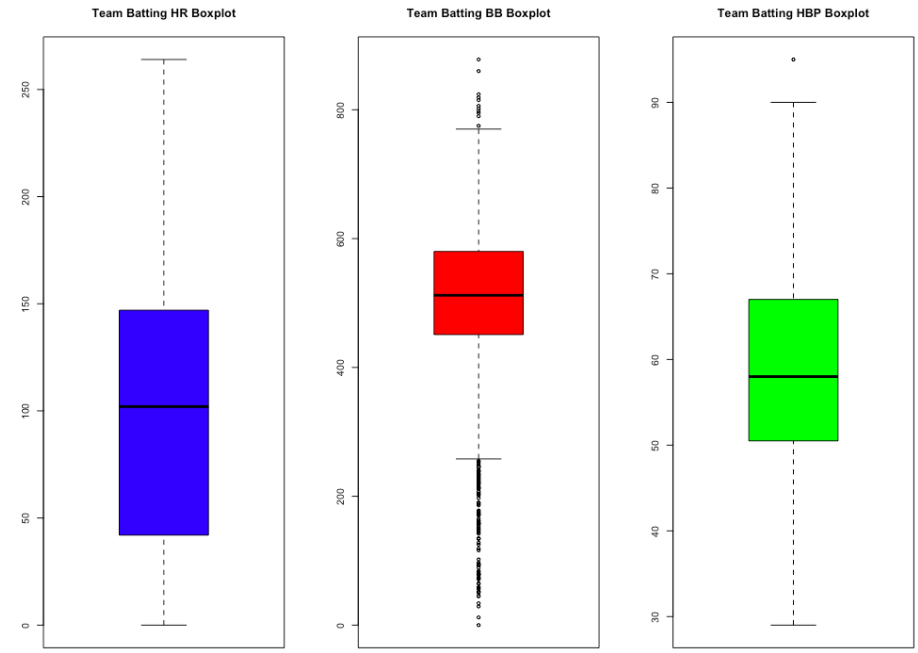
The histogram and boxplot above both indicate that TARGET_WINS is relatively normally distributed. There's no extreme outlier in the variable that significantly skews the data.



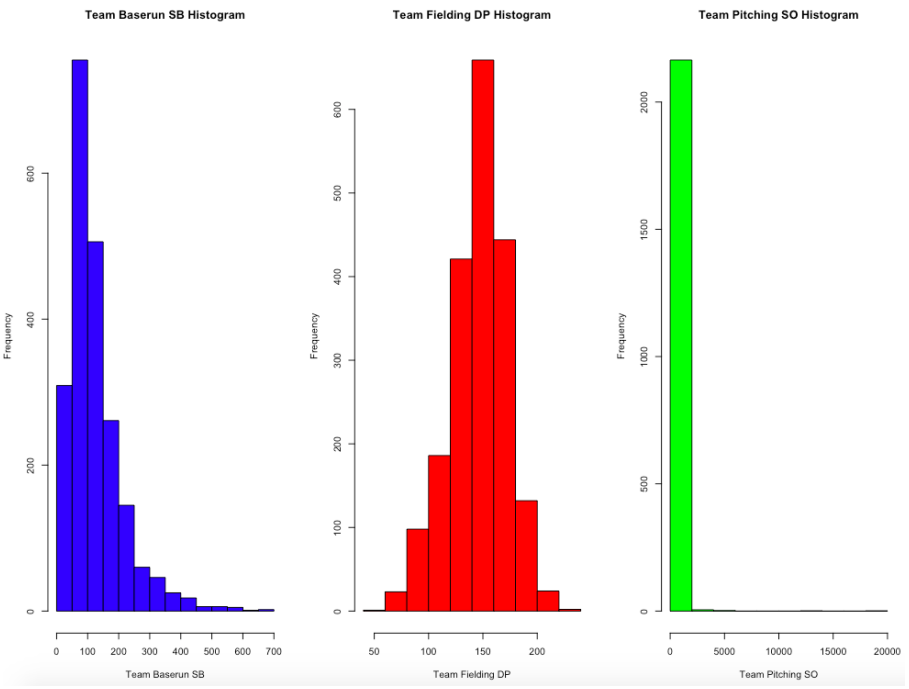


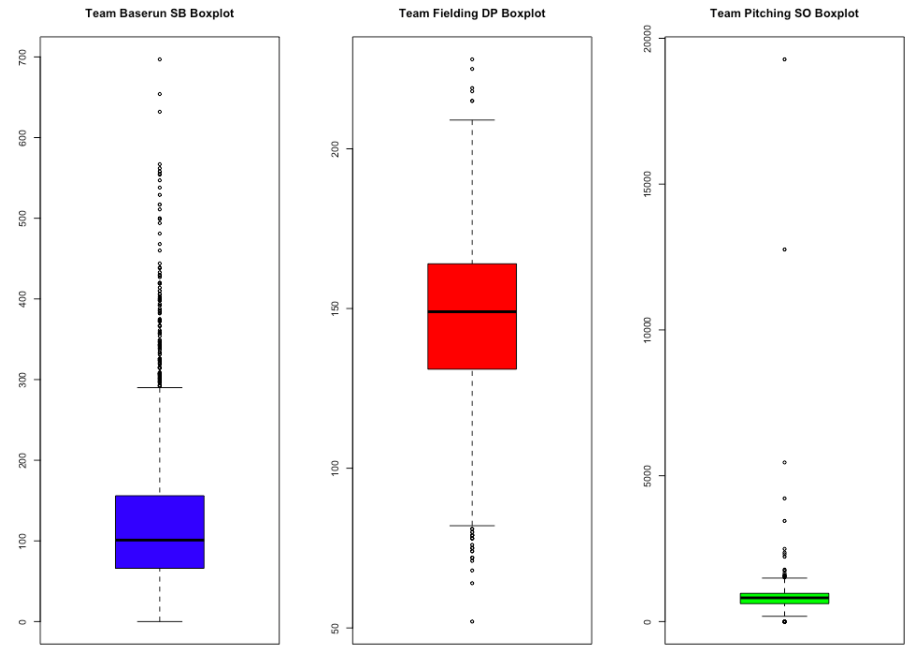
The histograms and boxplots above confirm our conclusion earlier from observing the skewness numbers: there's a moderate normality issue with TEAM_BATTING_H and TEAM_BATTING_3B whereas TEAM_BATTING_2B doesn't have this issue. Specifically, from the visuals above, it shows that TEAM_BATTING_H and TEAM_BATTING_3B may skew toward the right with outliers on the right tail of the variable. In other words, the majority of the outliers are much larger than other data points in the variables.



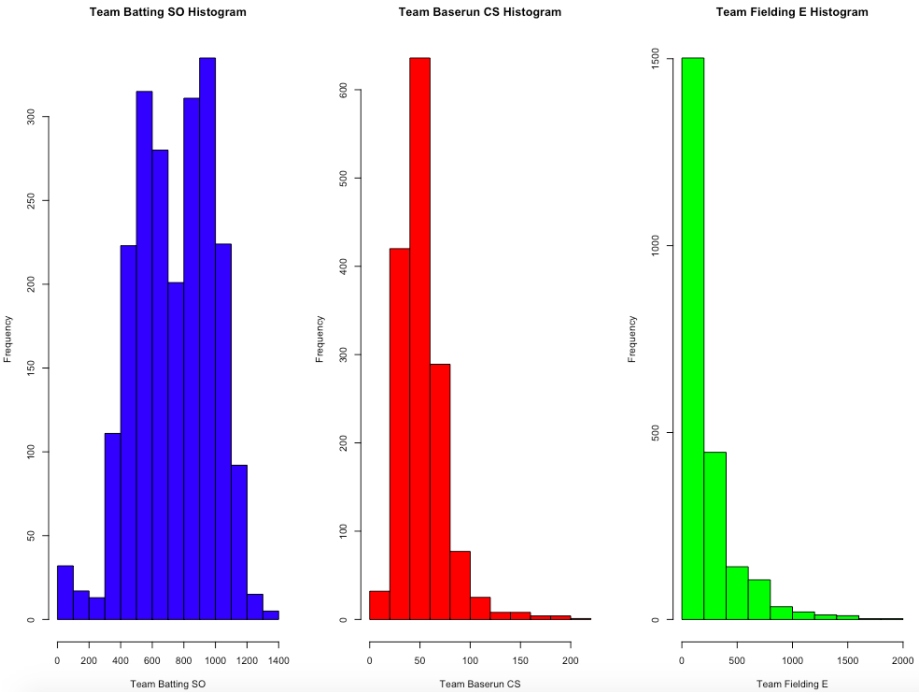


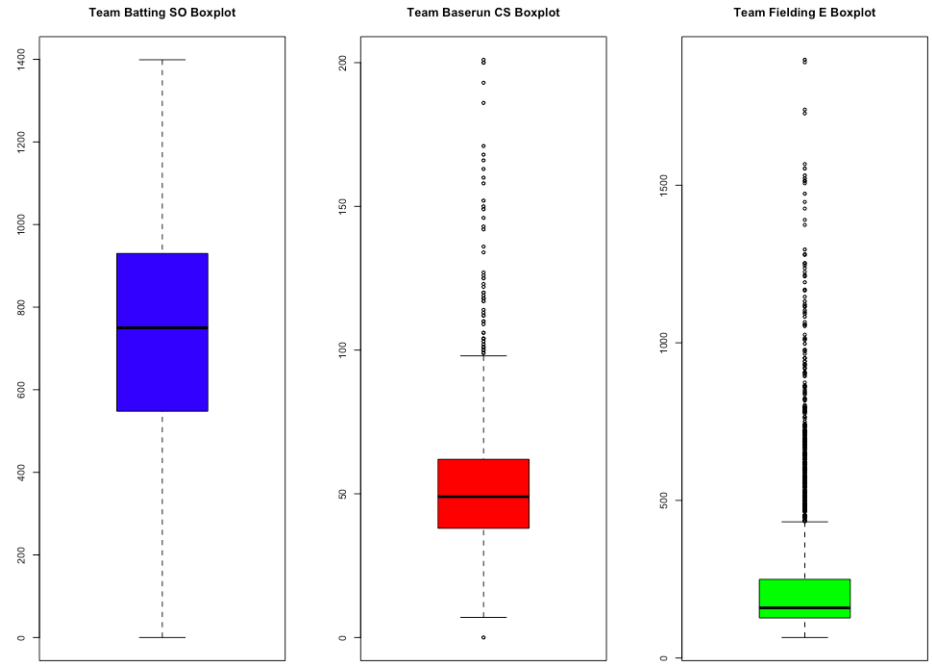
The histograms and boxplots above confirm the earlier conclusion from observing the skewness numbers that TEAM_BATTING_HR and TEAM_BATTING_HBP follow a normal distribution whereas TEAM_BATTING_BB moderately violates this normality assumption. Specifically, this variable has outliers on the left side of the tail, which means that majority of its outliers are much smaller than other data points, which skews the data to the left.



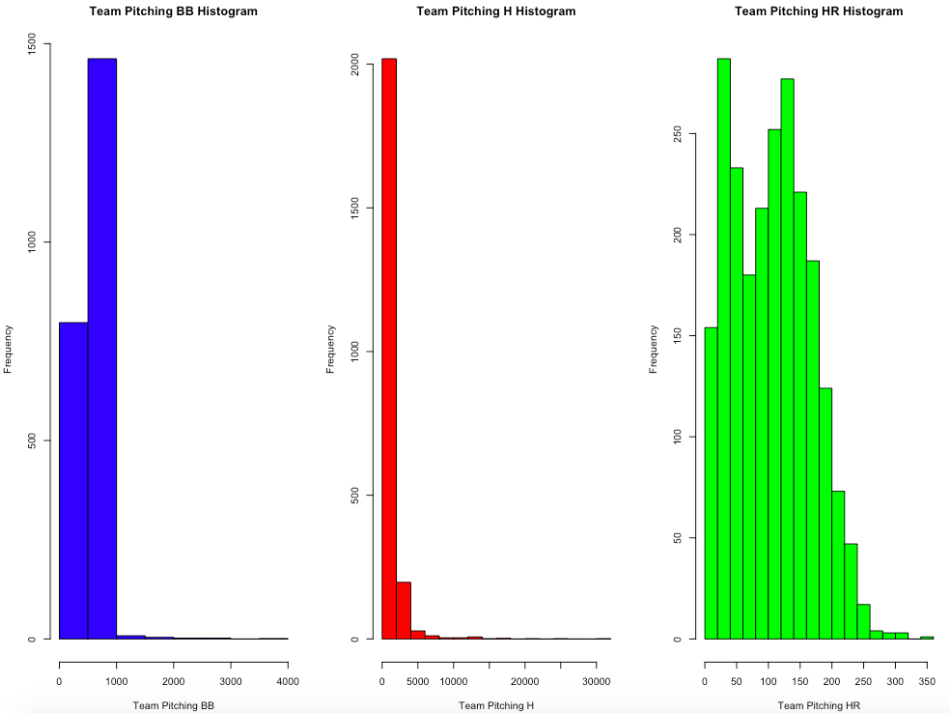


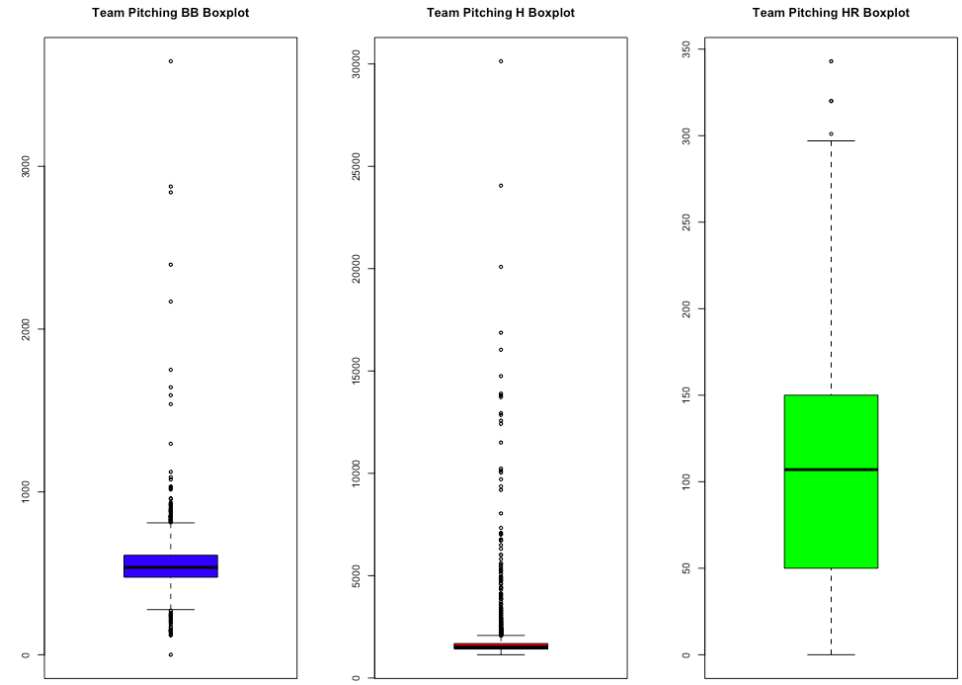
The histograms and boxplots above confirm the conclusion earlier from observing the skewness results that TEAM_BASERUN_SB moderately violates the normality assumption. Specifically, its outliers skew toward the right. TEAM_FIELDING_DP has outliers, but since it has outliers on both ends, it may still create a bell curve of normal distribution, which explains why its skewness number is close to 0. In contrast, TEAM_PITCHING_SO has severe normality issue, which is reflected by its high skewness number. Specifically, the outliers skew toward the right with two very big outliers on top of the boxplot and three relatively big outliers following in the boxplot.





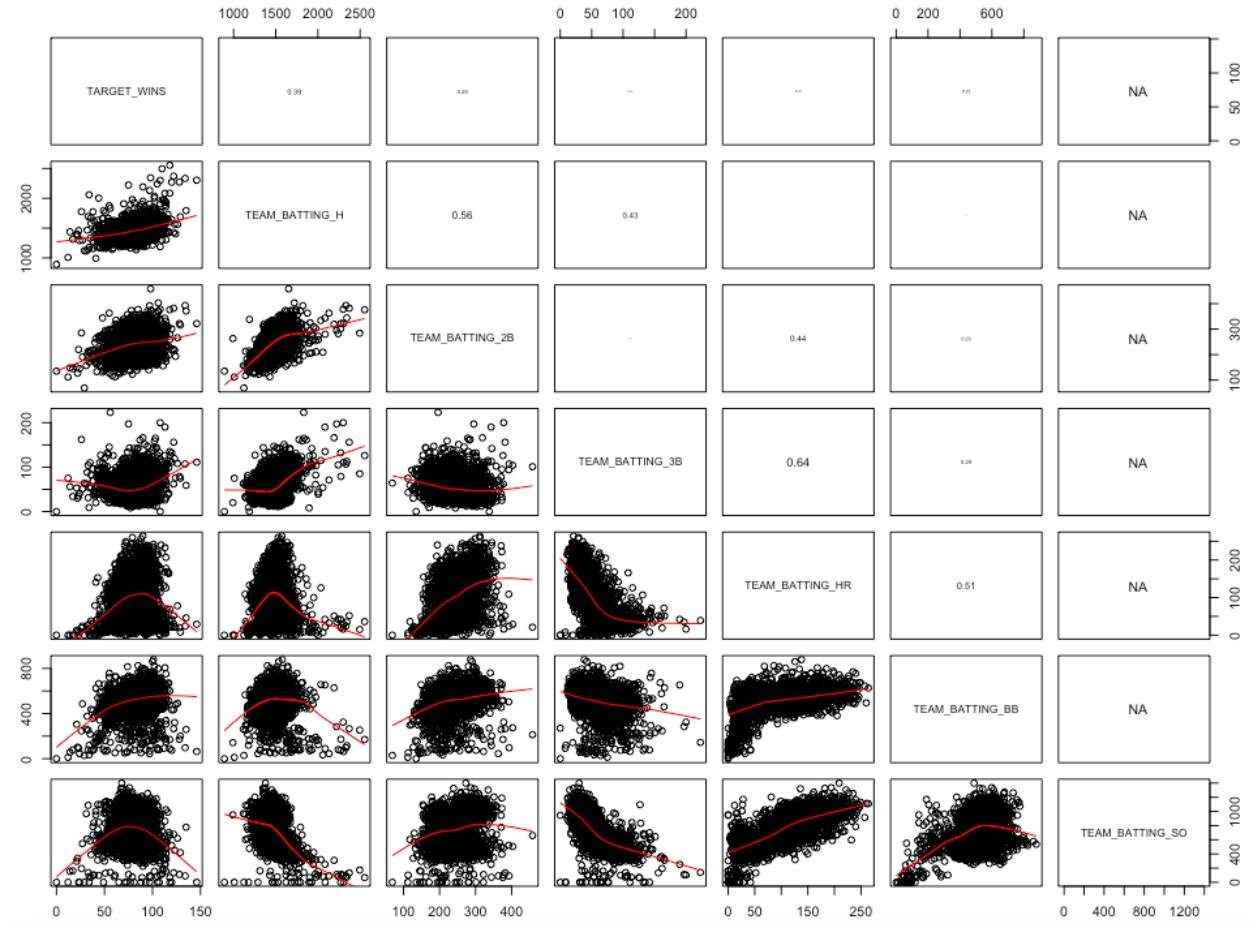
Similar to the results of the skewness numbers, the histograms and boxplots show that TEAM_BATTING_SO follows a normal distribution whereas TEAM_BASERUN_CS and TEAM_FIELDING_E violate this assumption. Specifically, both variables have outliers skewing toward the right.



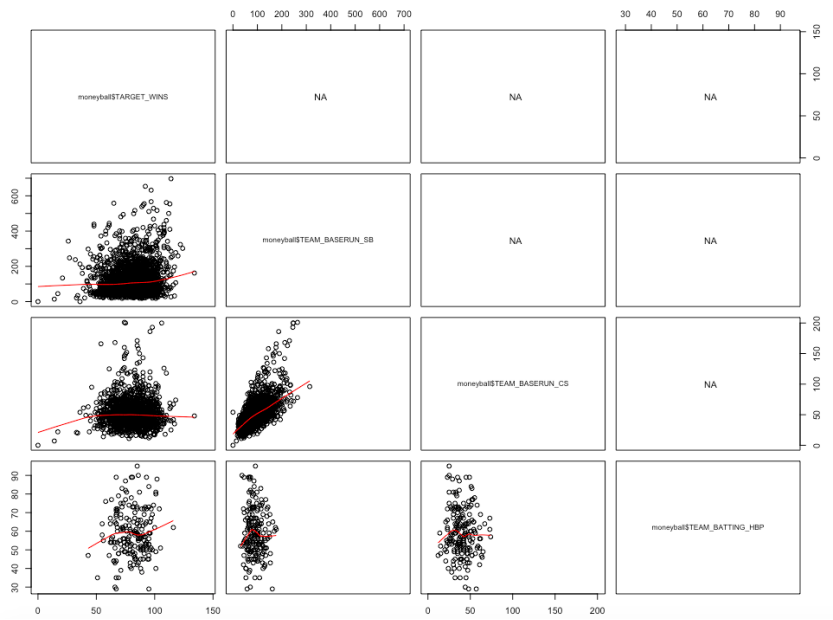


Similar to the skewness results earlier, the histograms and boxplots above indicate that TEAM_PITCHING_HR follows a normal distribution whereas TEAM_PITCHING_BB and TEAM_PITCHING_H face severe normality assumption violation. Specifically, both variables have outliers skewing toward the right.

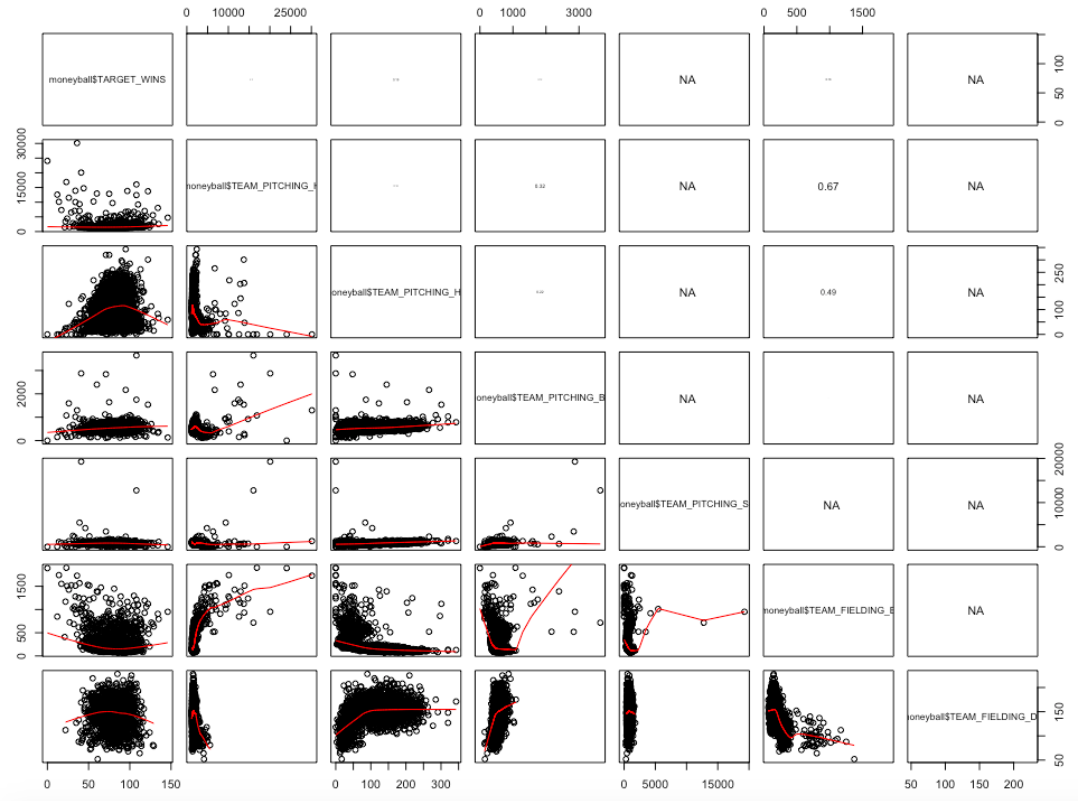
This section of the report identifies variables with missing value and outlier issues, which both need to be addressed prior to developing the predictive model. The following section of the report will fix these missing value and outlier issues.



From the correlation plot above, we may have multicollinearity issue in the dataset where predictor variables are correlated with each other. Specifically, TEAM_FIELDING_SO seem to have a negative correlation with TEAM_BATTING_H and positive correlation with TEAM_BATTING_3B, TEAM_BATTING_HR, and TEAM_BATTING_BB. This is another important issue we need to address in the project. However, unlike missing value and outlier issues that need to be fixed prior to building the model, we can address multicollinearity issue during the model development process using VIF technique, which is explained in later section of this report.



From the correlation plot above, TEAM_BASERUN_SB and TEAM_BASERUN_CS have a strong positive correlation, which may suggest a multicollinearity issue in the dataset. Again we can address this issue later during the model development stage using VIF technique. In addition, comparing to other variables, TEAM_BATTING_HBP only has a few data points as shown in the plot above, which confirm our intent earlier to remove this variable from the study because it doesn't add value to the predictive model due to too many missing values in the variable.



The correlation plot above doesn't show any significant correlation among variables, so there's no concern for multicollinearity here. There may be some non-linear relationship among the predictors such as a log relationship between TEAM_PITCHING_H and TEAM_FIELDING_E. However, these relationships don't seem to be very strong, so this should not be a concern for the predictive model.

Section 2: Data Preparation

The first part of this section fixes missing value issue whereas the second part of this section handles outlier issue. Among 3478 total missing values we have in the train dataset, below is the breakdown of number of missing values for each variable.

```
"TEAM_BATTING_SO"   "102"
"TEAM_BASERUN_SB"   "131"
"TEAM_BASERUN_CS"   "772"
"TEAM_BATTING_HDP" "2085"
"TEAM_PITCHING_SO" "102"
"TEAM_FIELD_DP"     "286"
```

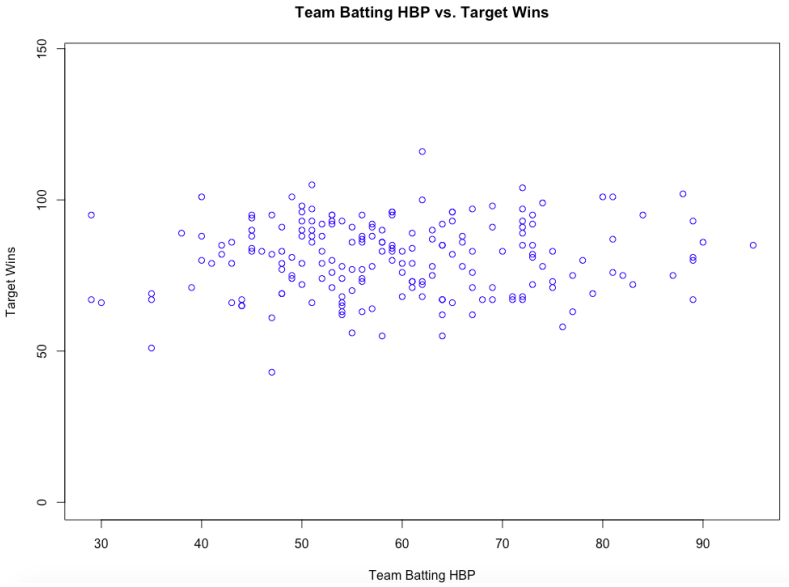
There are typically two steps to address missing values. First, a flag variable is created to identify if a variable has known or missing value. Because the fact that a variable has missing value may be predictive, this flag variable is included in the model. For example, a flag variable for TEAM_BASERUN_CS is created and called M_TEAM_BASERUN_CS where a 0 is assigned to records with known value and a 1 is assigned to records with missing value in the variable. A flag variable is created for each of the six variables above with missing values. Second, another new variable is also created to fix or impute variables with missing values. For example, a new variable IMP_TEAM_BASERUN_CS is created to impute the variable TEAM_BASERUN_CS by transforming it. Below are five common techniques to impute variables with missing values.

1. Delete records with missing values
2. Avoid using a variable that has a missing value
3. Use a business rule to fix a missing value
4. Fill in the missing value with an average value (i.e. Mean, Median, Mode)
5. Use a decision tree (or similar tool) to build a model to impute the missing value.

This project uses approaches #2 and #4 mentioned above. The details of how to impute these variables are explained in the following subsections.

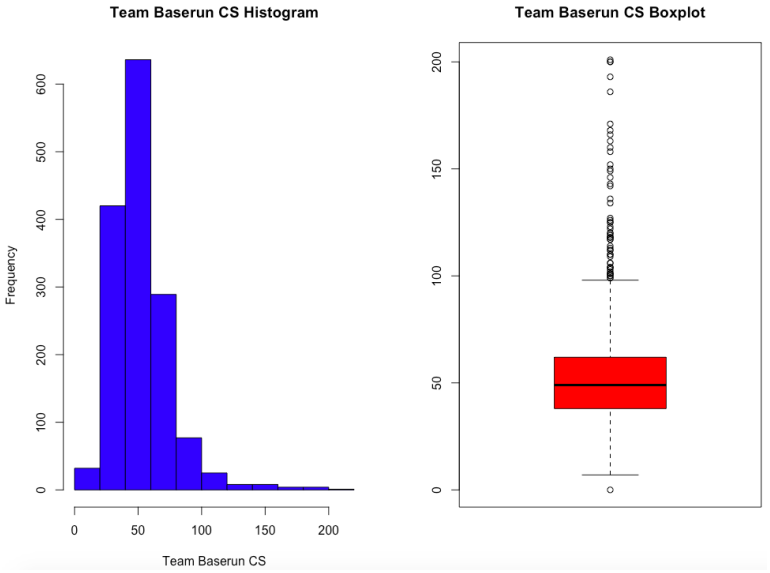
Subsection 2.1: Missing Value in TEAM_BATTING_HDP

As mentioned earlier, TEAM_BATTING_HDP has 92% of its data missing, so this variable doesn't have a good predictive capability for the model.



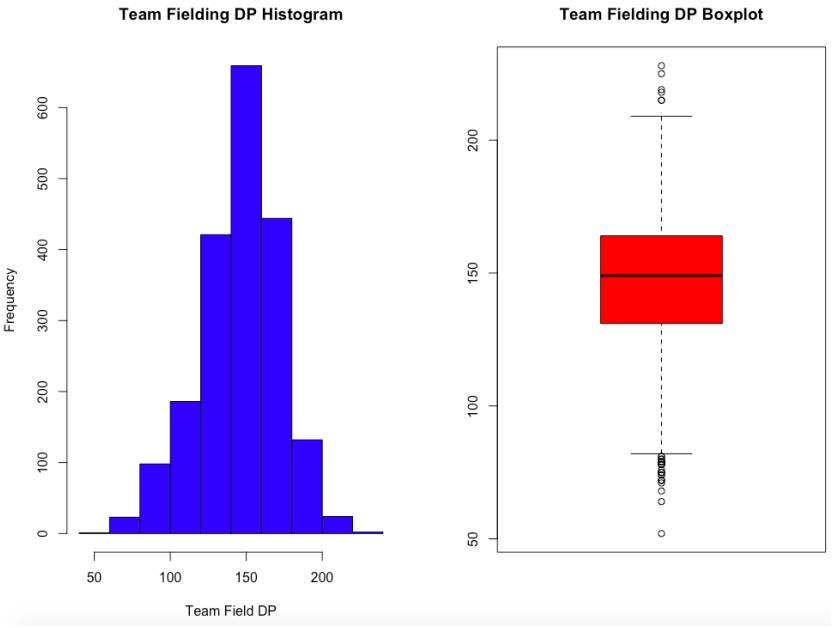
In addition, the plot above shows that there's no relationship between TEAM_BATTING_HBP and TARGET_WINS, so it's reasonable to assume that this variable won't be useful to include in the model. Therefore, we will use the second approach mentioned earlier to avoid using this variable in the model development process.

Subsection 2.2: Missing Value in TEAM_BASERUN_CS



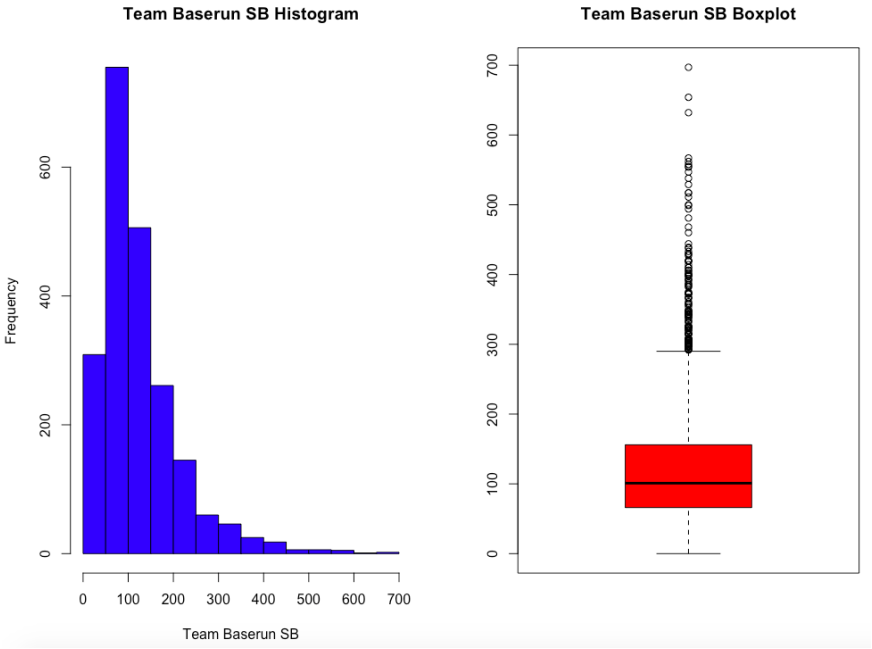
As shown earlier in previous section and the two charts above, TEAM_BASERUN_CS has the skewness number of 1.978191, which means that it's not normally distributed. Therefore, in order to avoid the skewness of outliers, we will use approach #4 to replace the missing records with the median value of 49.

Subsection 2.3: Missing Value in TEAM_FIELDING_DP



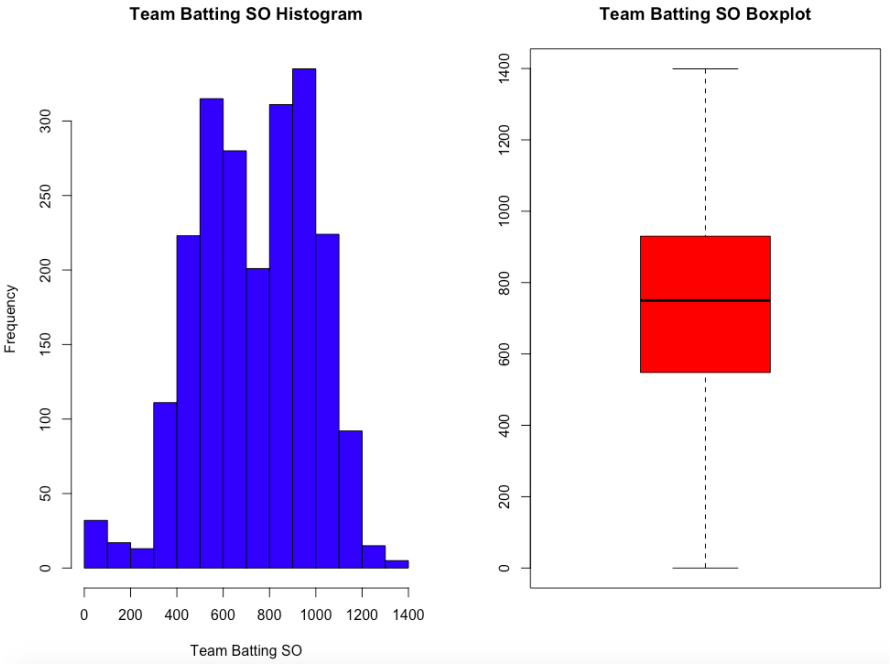
From the charts above, though TEAM_FIELDING_DP has outliers, it still follows a normal distribution since the outliers take place on both ends of the tail. The variable has a skewness of -0.3892324, which also confirms its normality. Therefore, we will replace the missing value with the mean of the variable, which is 146.3879.

Subsection 2.4: Missing Value in TEAM_BASERUN_SB



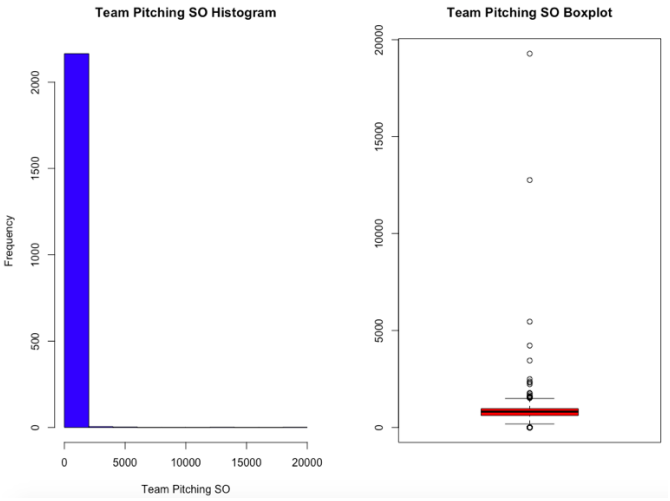
Unlike the previous variable, TEAM_BASERUN_SB has many outliers as shown by the histogram and boxplot above, along with a skewness number of 1.973794. Therefore, we will use the median value of 101 to replace missing records in this variable to avoid the skewness of outliers.

Subsection 2.5: Missing Value in TEAM_BATTING_SO



The two charts above along with the skewness of -0.2980057, it's obvious that TEAM_BATTING_SO doesn't have outliers. Therefore, we will replace missing records in this variable with its mean of 735.6053.

Subsection 2.6: Missing Value in TEAM_PITCHING_SO



Along with the big skewness number of 22.18986, the histogram and boxplot above indicate that TEAM_PITCHING_SO has severe normality issues with many outliers. Therefore, to avoid the big skewness of these influential outliers, we will use the median of 813.5 to replace the missing values in this variable.

Since we address the missing value issue in the first half of this section, the second half of this section will handle outlier issue. As mentioned earlier, a skewness number of 0 indicates a perfect normal distribution. The higher the absolute value of the skewness number, the more outliers in the data.

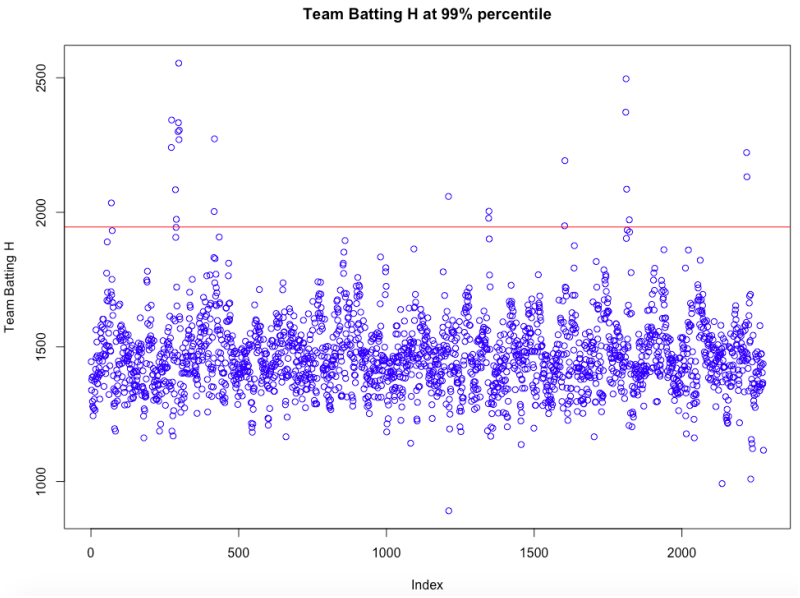
```
    TARGET_WINS  TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
     -0.3989861       1.5723696       0.2152436       1.1101968
TEAM_BATTING_HR TEAM_BATTING_BB
      0.1861648      -1.0264363
TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
     10.3363225        0.2879774        6.7483465
TEAM_FIELDING_E
       2.992438
IMP_TEAM_BASERUN_CS
          2.603888
IMP_TEAM_FIELDING_DP
         -0.4162637
IMP_TEAM_BASERUN_SB
           2.06719
IMP_TEAM_BATTING_SO
         -0.3049165
IMP_TEAM_PITCHING_SO
           22.70541
```

From the skewness results above, we have small outlier issues with variables TEAM_BATTING_H, TEAM_BATTING_3B, and TEAM_BATTING_BB. We have moderate outlier issues with variables TEAM_FIELDING_E, IMP_TEAM_BASERUN_CS, and IMP_TEAM_BASERUN_SB. Finally, we have severe outlier issues with variables TEAM_PITCHING_H, TEAM_PITCHING_BB, and TEAM_PITCHING_SO. Below are potential solutions to address the problem of outliers.

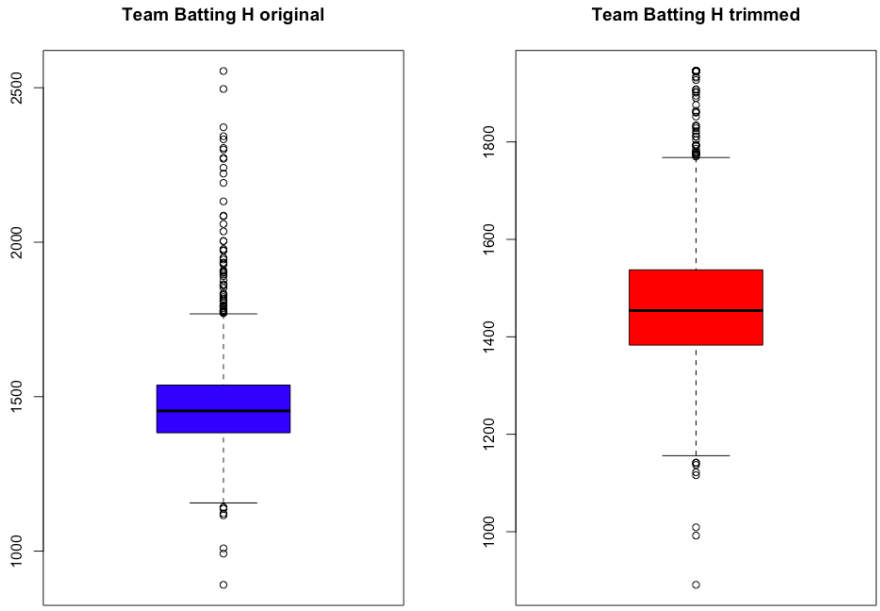
1. Increase size of dataset
2. Increase number of variables used
3. Remove outliers prior to building model
4. Build multiple models
5. Use a different modeling technique (decision tree, random forests, gradient boosting)
6. Transform variables

Due to limited time and resources, we can't take approaches #1 and #2. Due to the small sample size of the train dataset, we can't take approaches #3 and #4. We will use approach #6 for this project. If the outlier issue is still severe after the transformation, we will consider approach #5.

Subsection 2.7: Outlier in TEAM_BATTING_H

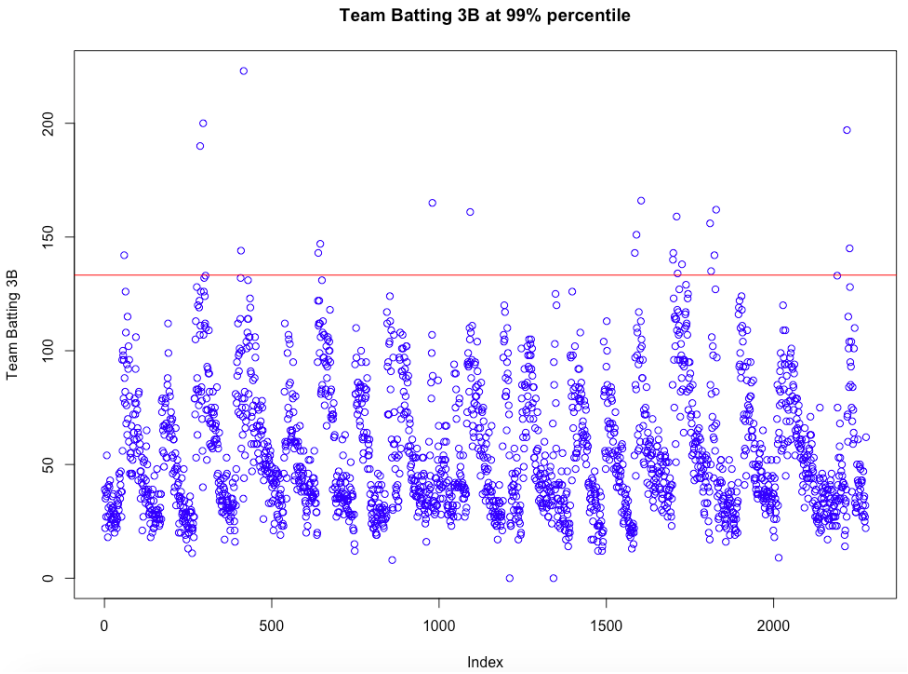


For TEAM_BATTING_H, we will transform the variable by trimming it at 99% level of 1945.5 value, which is shown as the red line in the plot above.

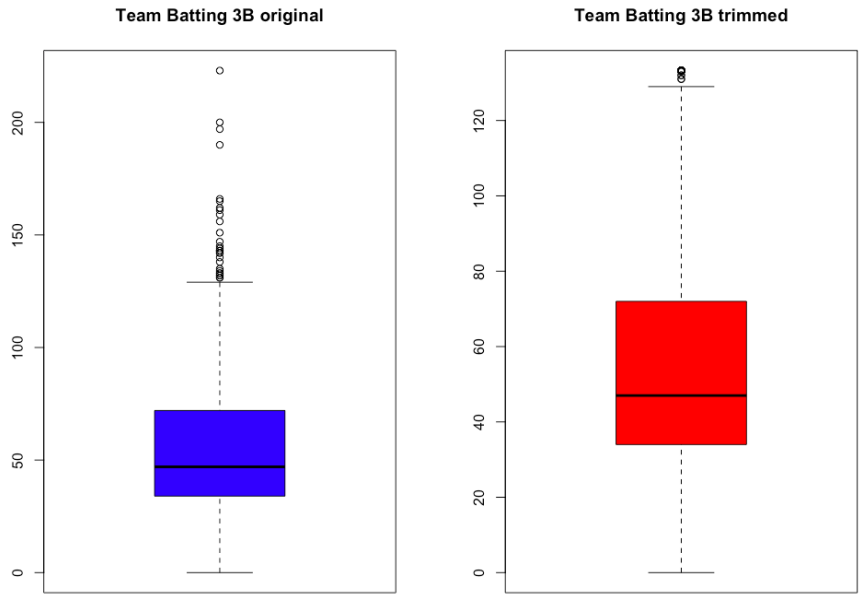


The boxplots above show the variable before and after the trimming where there are much fewer outliers after the transformation. In addition, the skewness number of the original variable is 1.57237 whereas the number of the trimmed variable is 0.7014834. Therefore, we have successfully transformed the variable by trimming it to 99% level to fix the outlier issue. We only trimmed the upper level of the variable since the majority of its outliers skew toward the right tail.

Subsection 2.8: Outlier in TEAM_BATTING_3B

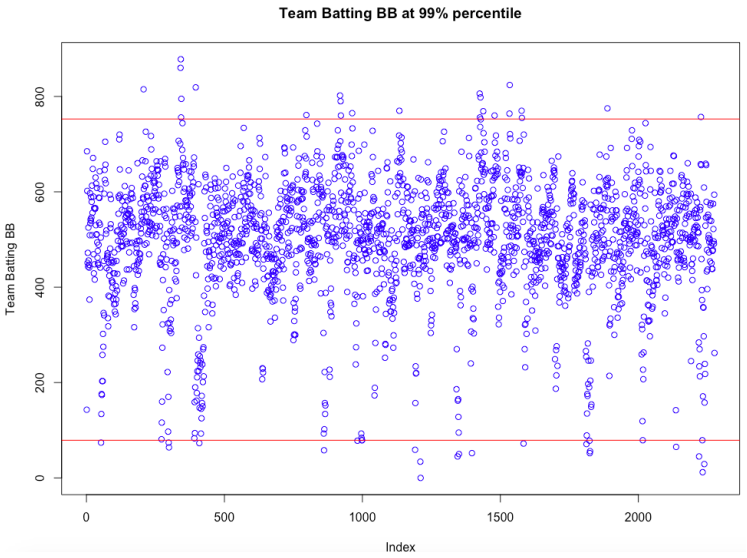


Similarly, we will address the outlier issue in TEAM_BATTING_3B through transformation by trimming it to 99% level of 133.25 value, as shown in the red line in the plot above.

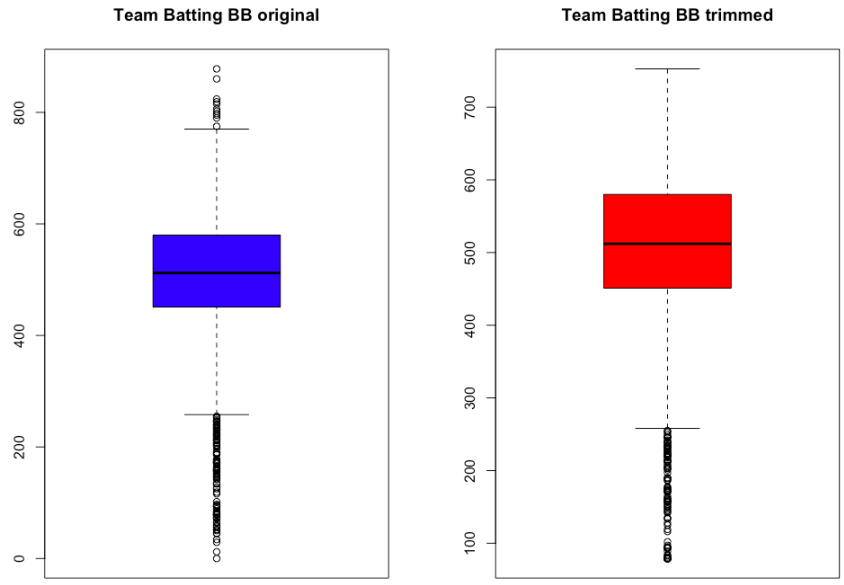


The boxplots above show that there are much fewer outliers after the transformation. Moreover, the skewness number of the original variable is 1.110197, and the number of the trimmed variable is 0.854194. Therefore, we have successfully fixed the outlier issue in TEAM_BATTING_3B.

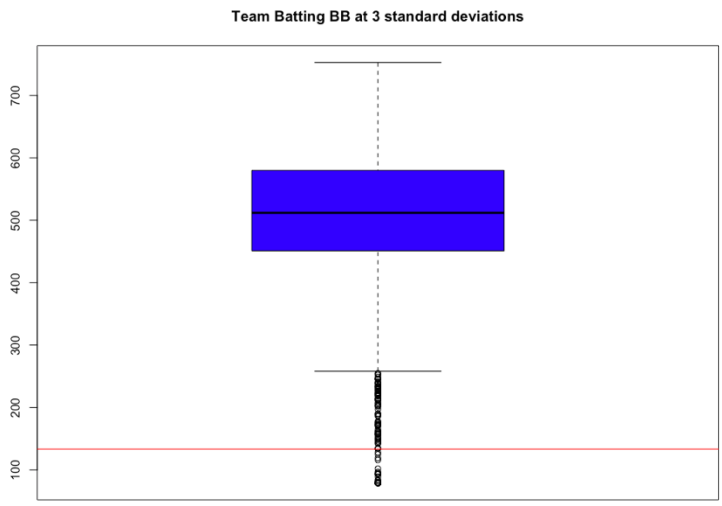
Subsection 2.9: Outlier in TEAM_BATTING_BB



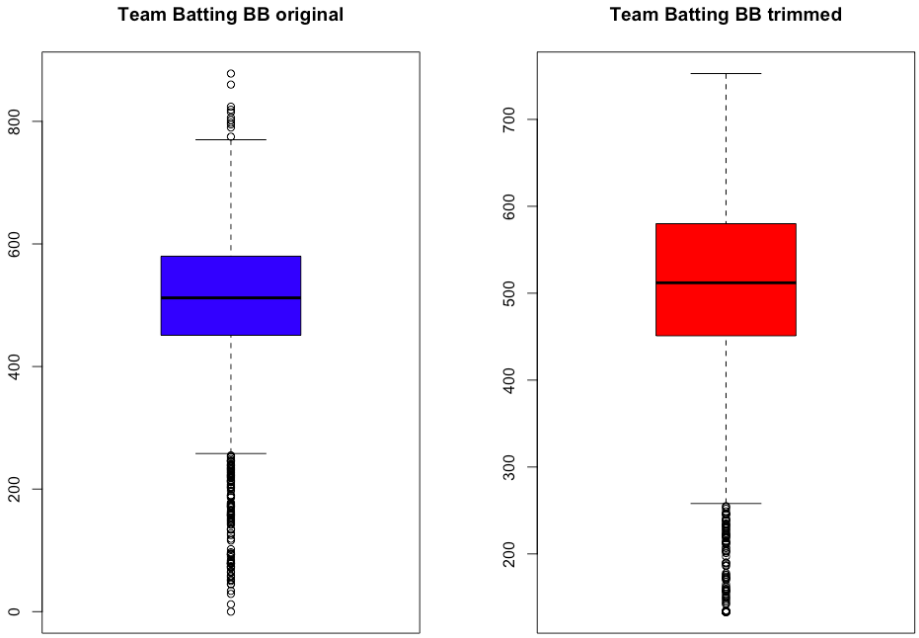
The variable TEAM_BATTING_BB has many outliers on both tails. Therefore, we will trim it at 99% level using both lower level of 79 and upper level of 752.75, as shown by the two red lines in the plot above.



The boxplots above show that we have fixed outlier issues on the right tail or upper level using the 99% trim. The skewness of the original variable is -1.026436, and the skewness of the trimmed variable is -1.038008. So, we still didn't completely address the outlier issue in this variable. Therefore, we will continue to transform this variable by using another approach, which is the z-standardization. Using z-standardization, any records above or below three standard deviation from the mean is considered an extreme outlier. We will trim these extreme outliers.



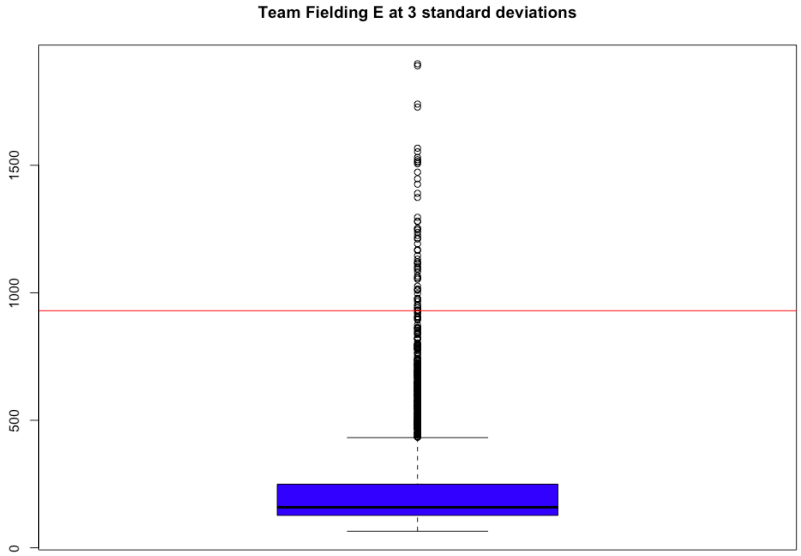
From the boxplot above, there's no extreme outlier on the upper level, which is correct since we already addressed outliers on the upper level using the 99% trim. However, there are still a lot of outliers in the lower level (data points below the red line on the plot above); some of these are extreme outliers. We will transform the variable by trimming these extreme outliers.



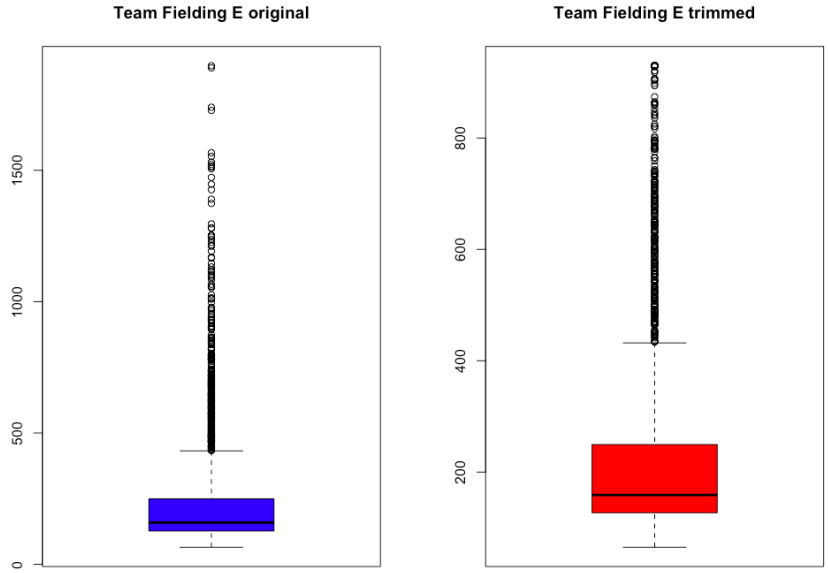
According to the boxplots above, there are much fewer outliers in the variable after the 99% trim and z-standardization transformation. The skewness number of the original variable is -1.026436 whereas the skewness number of the variable is -0.9006631. Therefore, although there are still outliers in the variable, specifically in the lower level of the data, we have managed to keep the outlier issue under control for this variable. Since the skewness number of the trimmed variable is within the -1 to 1 range, there's no need to explore another option to address the outlier issue for this variable.

Subsection 2.10: Outlier in TEAM_FIELDING_E

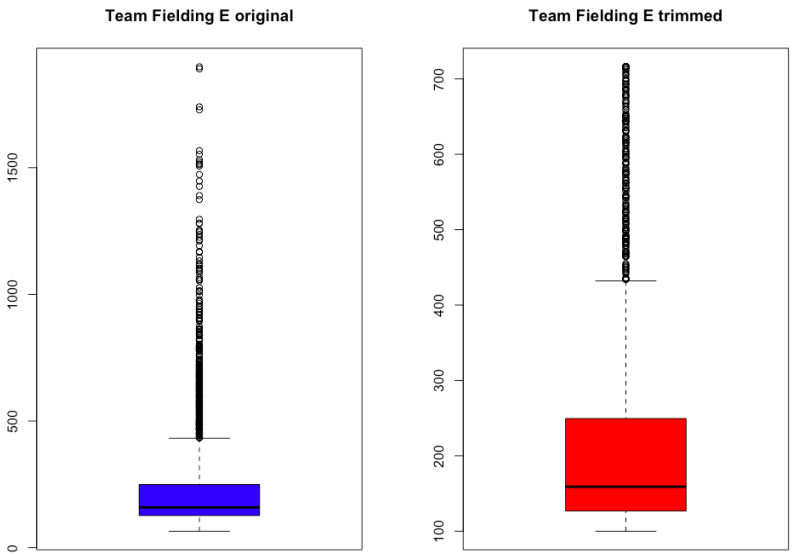
Since TEAM_FIELDING_E and other subsequent variables in the remaining of this section have more severe outlier issues, we will start the transformation process with the z-standardization.



The boxplot above shows that there's no outlier in the lower level. Instead all outliers lie in the upper level, and the red line indicates three standard deviation away from the mean, where we will trim the data as part of the z-standardization procedure.

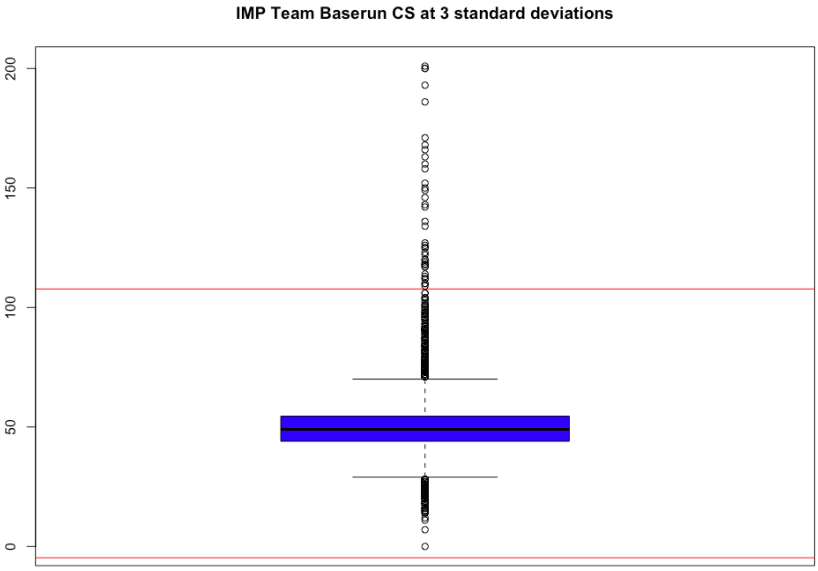


The boxplots above show that we have reduced the outliers in the variable, but we still have a lot of outliers left. In addition, the skewness before the trim is 2.992438 and after the trim is 2.131177. The skewness number is still high, so we will take an additional step to trim at 95% level to get better results for the transformation.

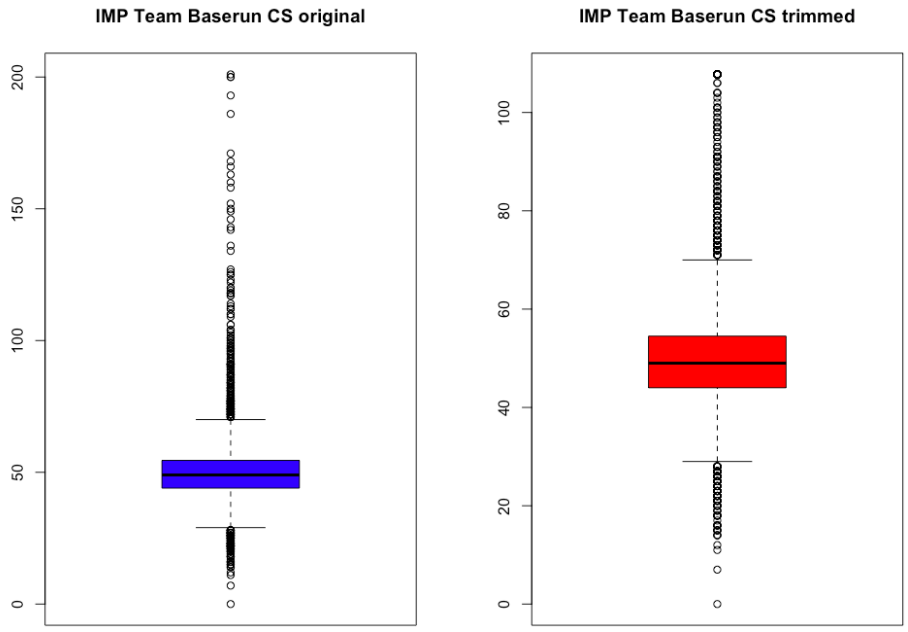


The boxplots show that we have removed more outliers after the 95% trim. The skewness before the transformation is and after two transformations is 1.794715. The skewness number is still high, but at least it's still under 2.0. Even though we didn't completely address the outlier issue in this variable, we have minimized it and kept it under control.

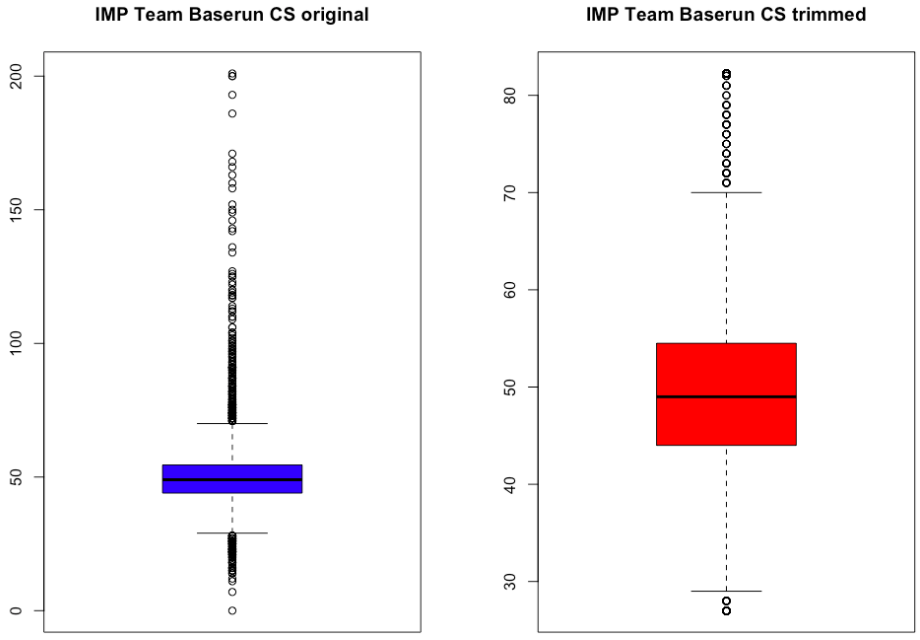
Subsection 2.11: Outlier in IMP_TEAM_BASERUN_CS



Similar to the previous variable, IMP_TEAM_BASERUN_CS also has many outliers. There's no extreme outlier at the lower level. Rather they all lie in the upper level of the data, so we only need to trim the extreme outliers above the second red line on the chart above using z-standardization trimming.

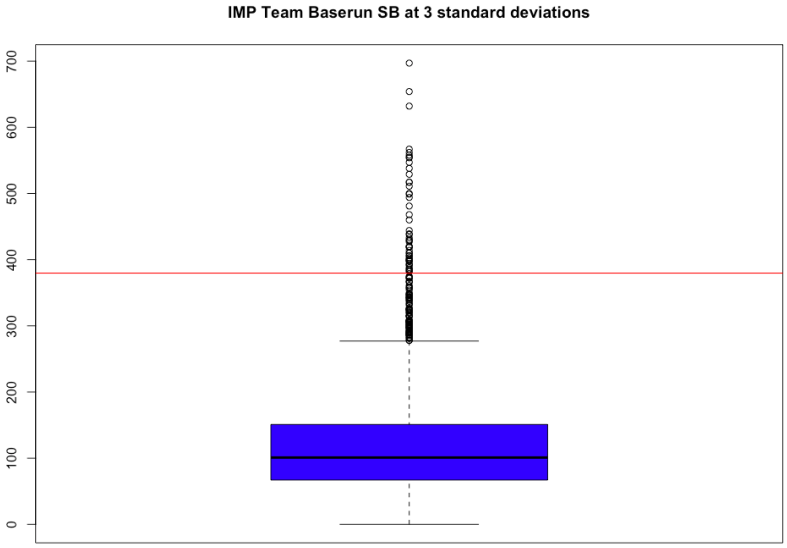


From the boxplots above, we have trimmed many outliers. The original skewness is 2.603888, and the trimmed skewness is 1.141844. However, there's still room for improvement. So we will take an additional step to trim at 95% level.

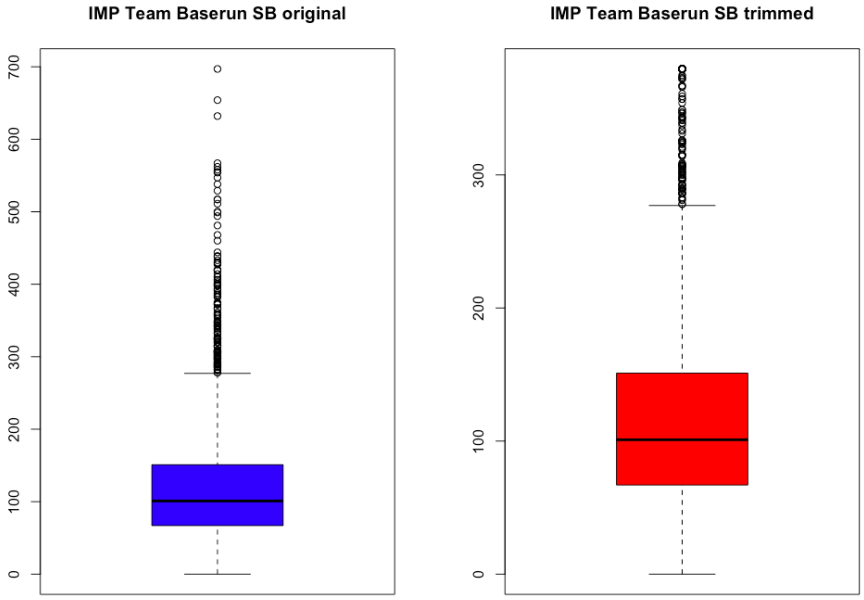


The boxplots above show that we have trimmed majority of outliers in the data. The original skewness is 2.603888, and the trimmed skewness is 0.6206941, which is within the -1 to 1 range. Therefore, we have successfully addressed the outlier issue for this variable using the combination of z-standardization and 95% level trim.

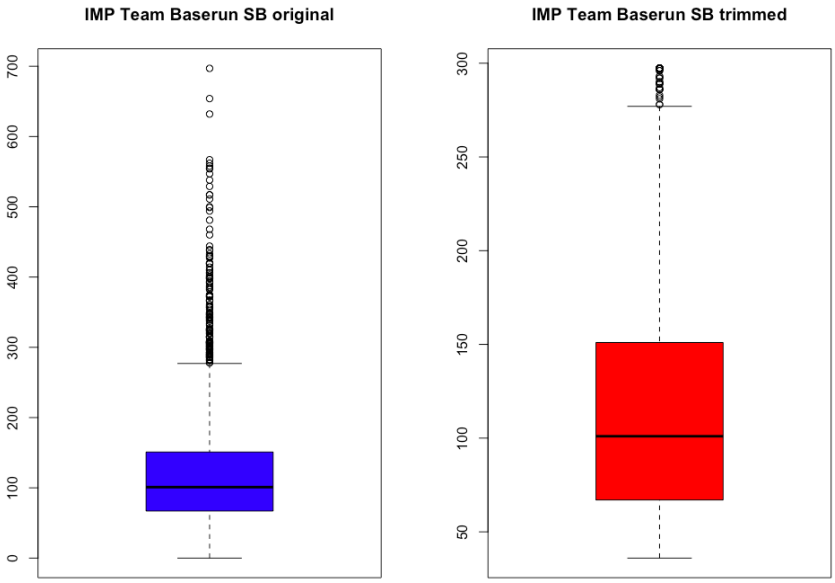
Subsection 2.12: Outlier in IMP_TEAM_BASERUN_SB



Similar to the previous variable, IMP_TEAM_BASERUN_SB also only has outliers at the upper level, which is three standard deviation above the mean, indicated as the data points above the red line in the boxplot above.

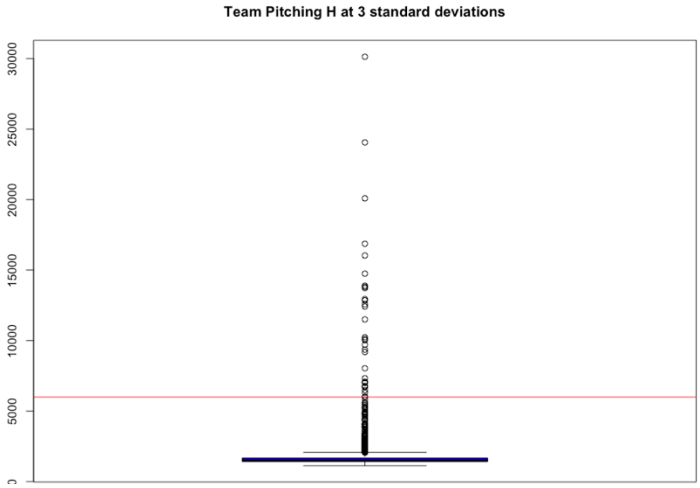


After trimming extreme outliers three standard deviation above the mean, as shown by the boxplots above, we have addressed many outliers. The skewness before the transformation is 2.06719 and after is 1.463464. We can still improve this skewness number by conducting a 95% trim on the variable.

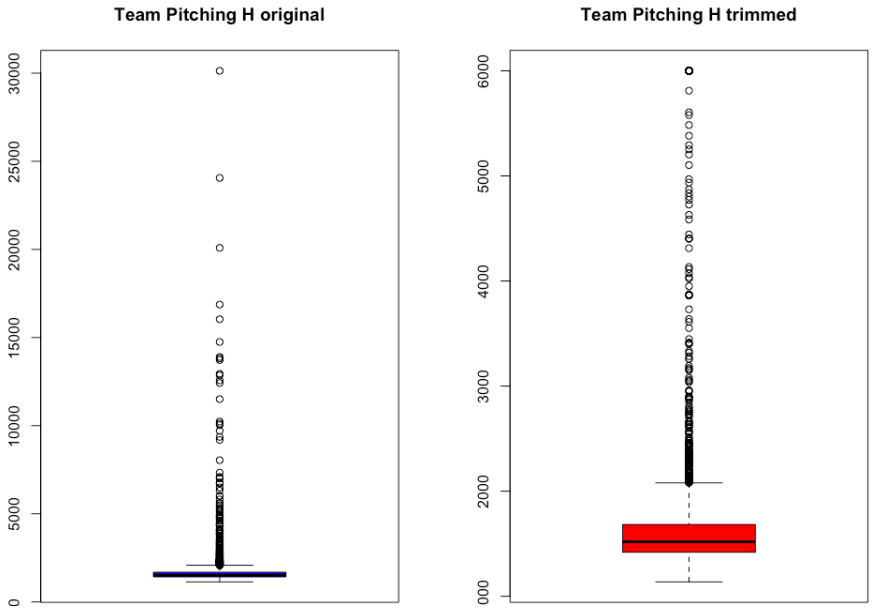


After two transformations, z-standardization and 95% trim, we have trimmed majority of the outliers in this variable. The skewness originally is 2.06719 and after two transformations is 1.089263, which is close enough to the -1 to 1 range. Therefore, we have successfully transformed the variable to fix its outlier issue.

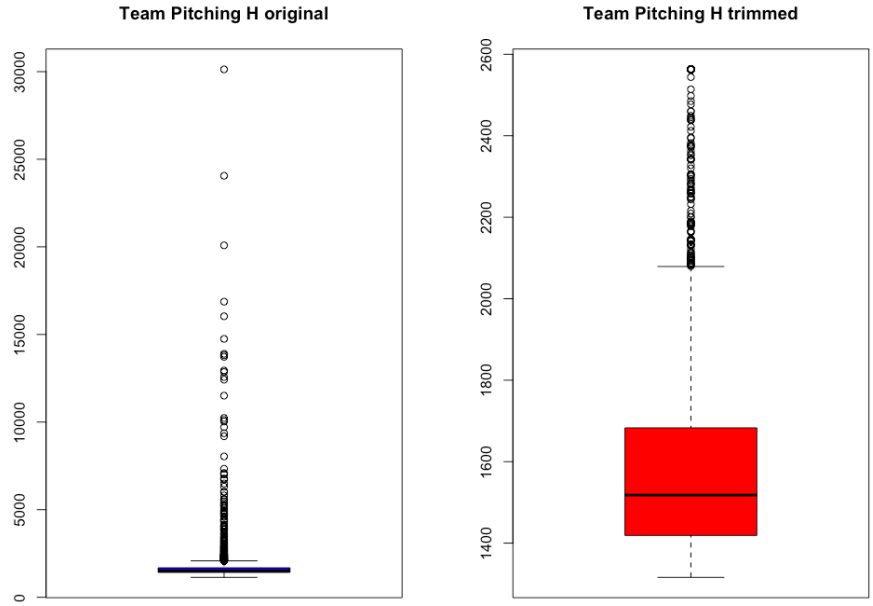
Subsection 2.13: Outlier in TEAM_PITCHING_H



Unlike previous variables, TEAM_PITCHING_H has very severe outlier issue, especially with a few influential points with absolute values much higher than the rest of the data points. There's no outlier in the lower level. All outliers lie in the upper level of the distribution. We will start the transformation process with a z-standardization and then 95% trim.

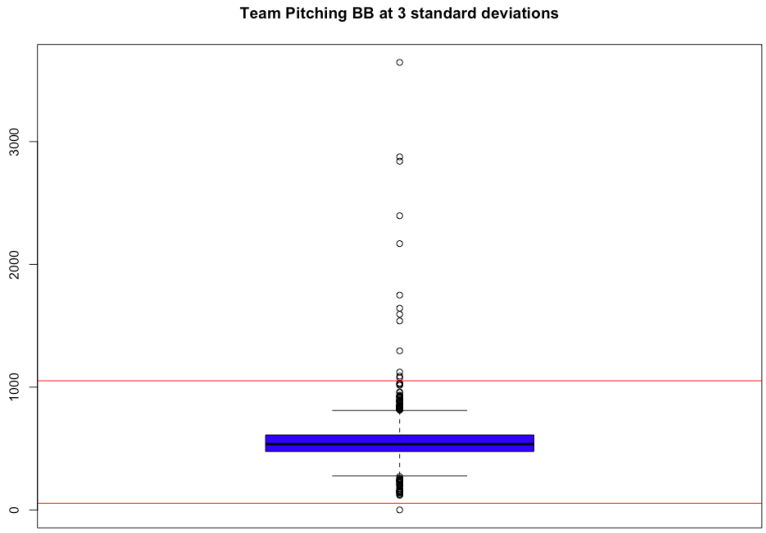


The z-standardization transformation has trimmed many outliers from the original data. The skewness before is 10.33632 and after the transformation is 4.531804, which is a significant drop. However, this is still a very high skewness number. So, we will continue the transformation process with a 95% trim.

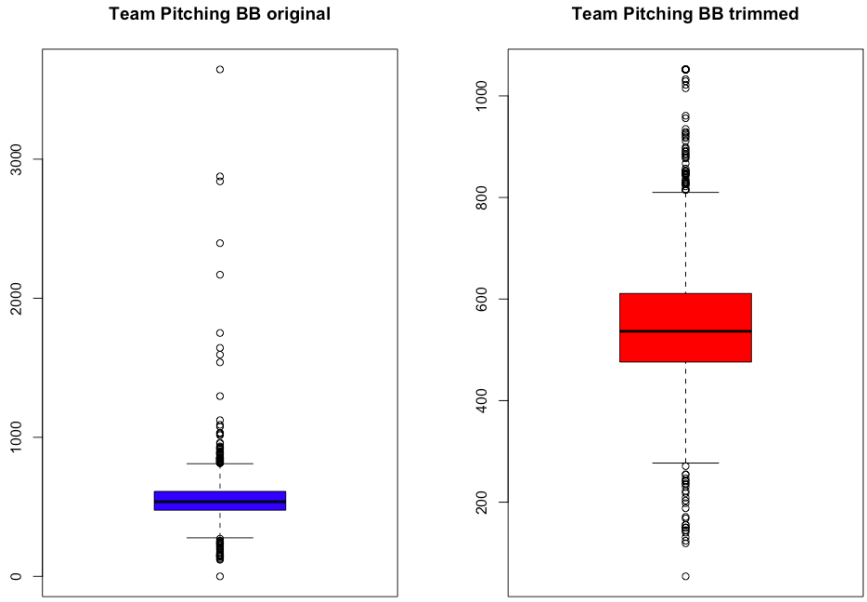


After two transformation, we have trimmed many outliers from this variable. The skewness at the beginning is 10.33632 and after the transformation is 1.805284. Even though it's not in the -1 to 1 range, the skewness is still under 2.0, which is good. Thus, although we didn't address completely the outlier issue in this variable, we have minimized it and kept it under control.

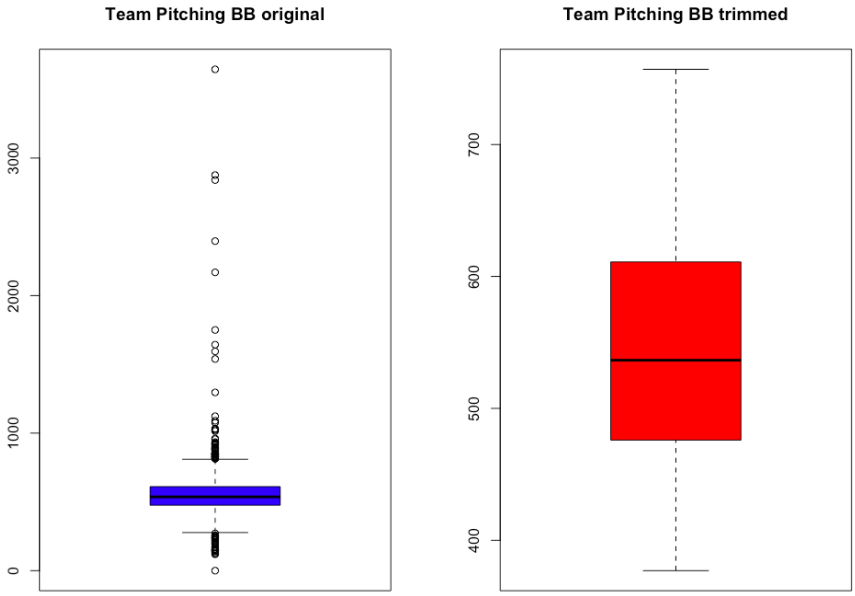
Subsection 2.14 Outlier in TEAM_PITCHING_BB



Similar to the previous variable, TEAM_PITCHING_BB also has a severe outlier issue with many influential points at the upper level with absolute values much higher than the majority of the data points. However, this variable also has a few outliers at the lower level. Thus, a combination of z-standardization and 95% trim is needed to address the issue.

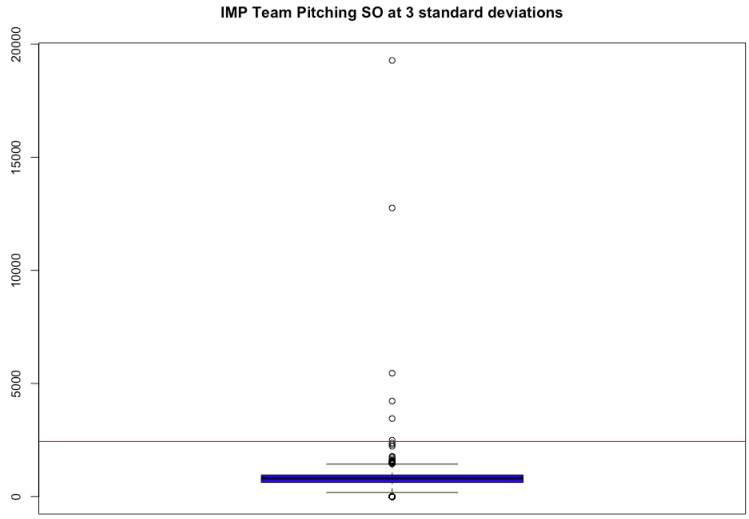


After the first transformation using z-standardization, we have trimmed majority of outliers at the upper level, as shown by the boxplots above. The skewness before is 6.748346 and after is 0.5595788. Although the skewness number is within the -1 to 1 range, since we still have many outliers on the two ends of the distribution after the z-standardization transformation, to be conservative, we will proceed with the 95% trim as part of the transformation process.

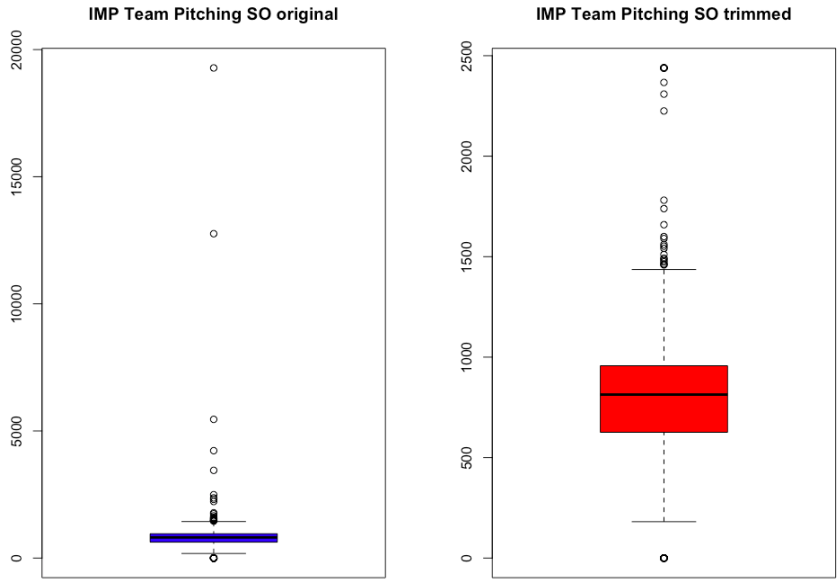


The boxplots show that after two transformation, the variable no longer has outlier. The skewness number originally is 6.748346 and now after two transformations is 0.3422309. Therefore, we have successfully and completely addressed the severe outlier issue for this variable.

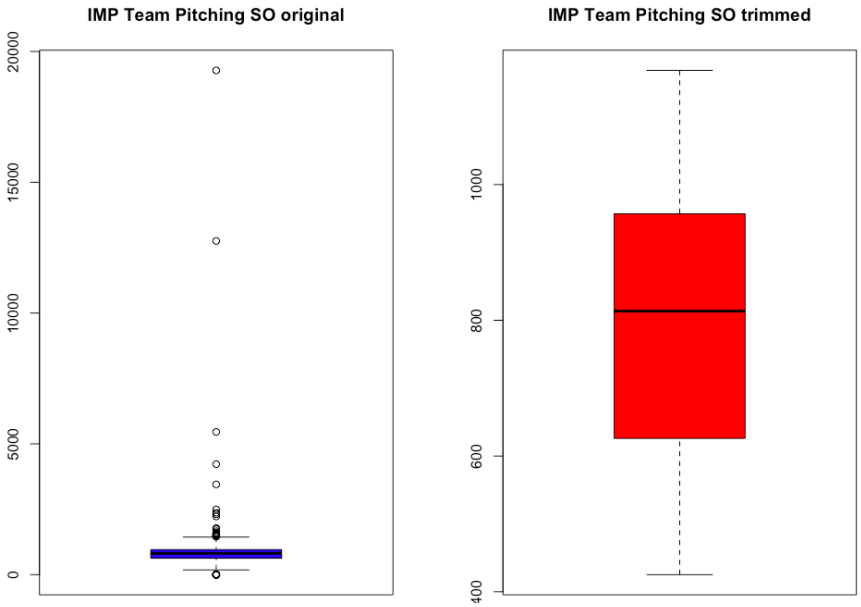
Subsection 2.15: Outlier in IMP_TEAM_PITCHING_SO



Similar to the previous variable, IMP_TEAM_PITCHING_SO also has severe outlier issue. There's only one outlier at the lower level whereas majority of outliers lie in the upper level. Specifically, there are many influential points with absolute value much higher than the rest of the data points, as revealed by the boxplot above. Therefore, a combination of z-standardization and 95% trim is necessary for the transformation process to address the severe outlier issue.



The z-standardization transformation has trimmed a lot of outliers at the upper level. The original skewness is 22.70541 and after the first transformation is 0.8284722. Although this is in the -1 to 1 range, there's still room for improvement, and we still see a few outliers in the boxplot. So we will proceed with the transformation process using the 95% trim.

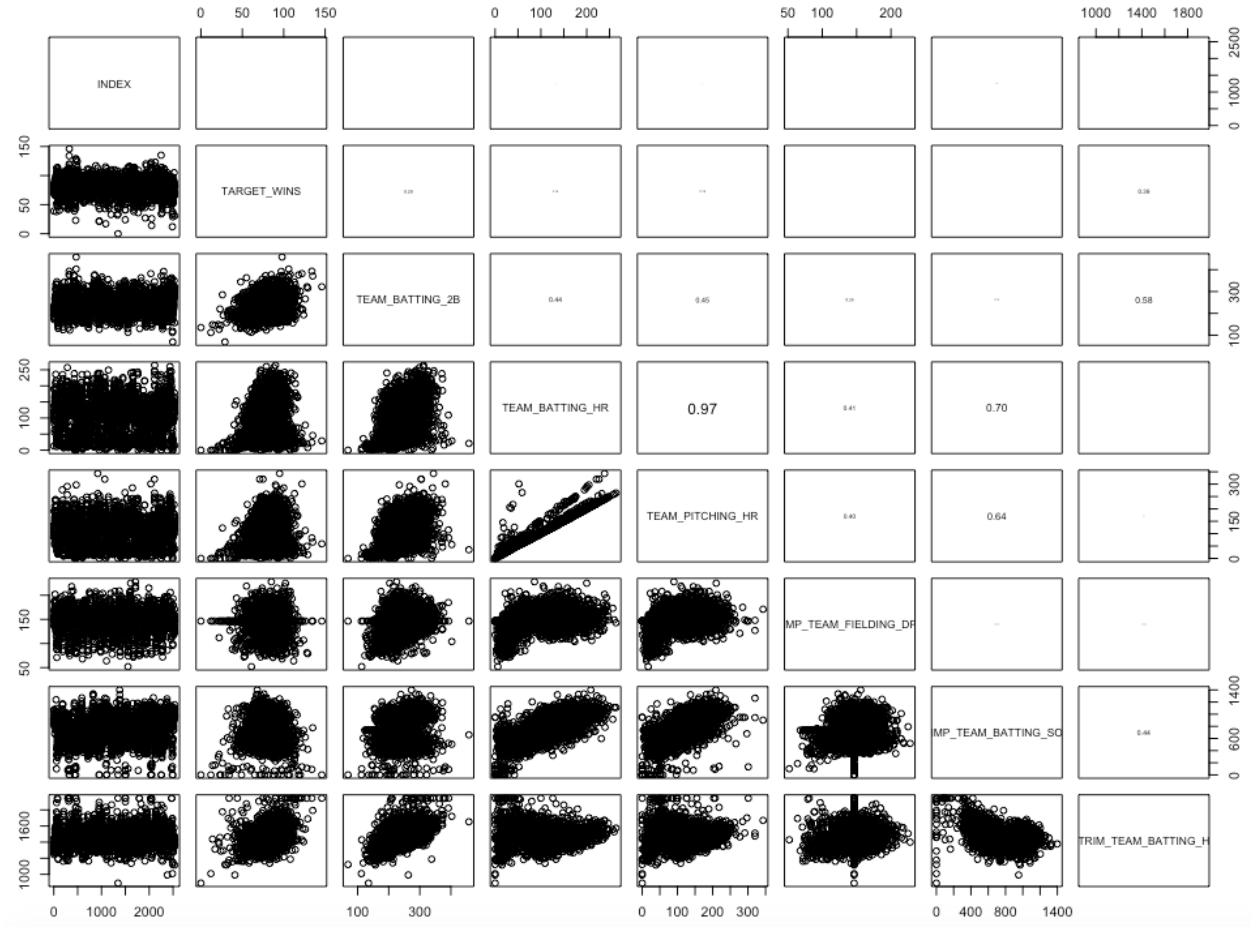


The original skewness is 22.70541, and now it's -0.04381974 after two transformations. From the boxplots above, we don't see any more outlier after the second transformation. Therefore, we have successfully and completely addressed the severe outlier issue in this variable.

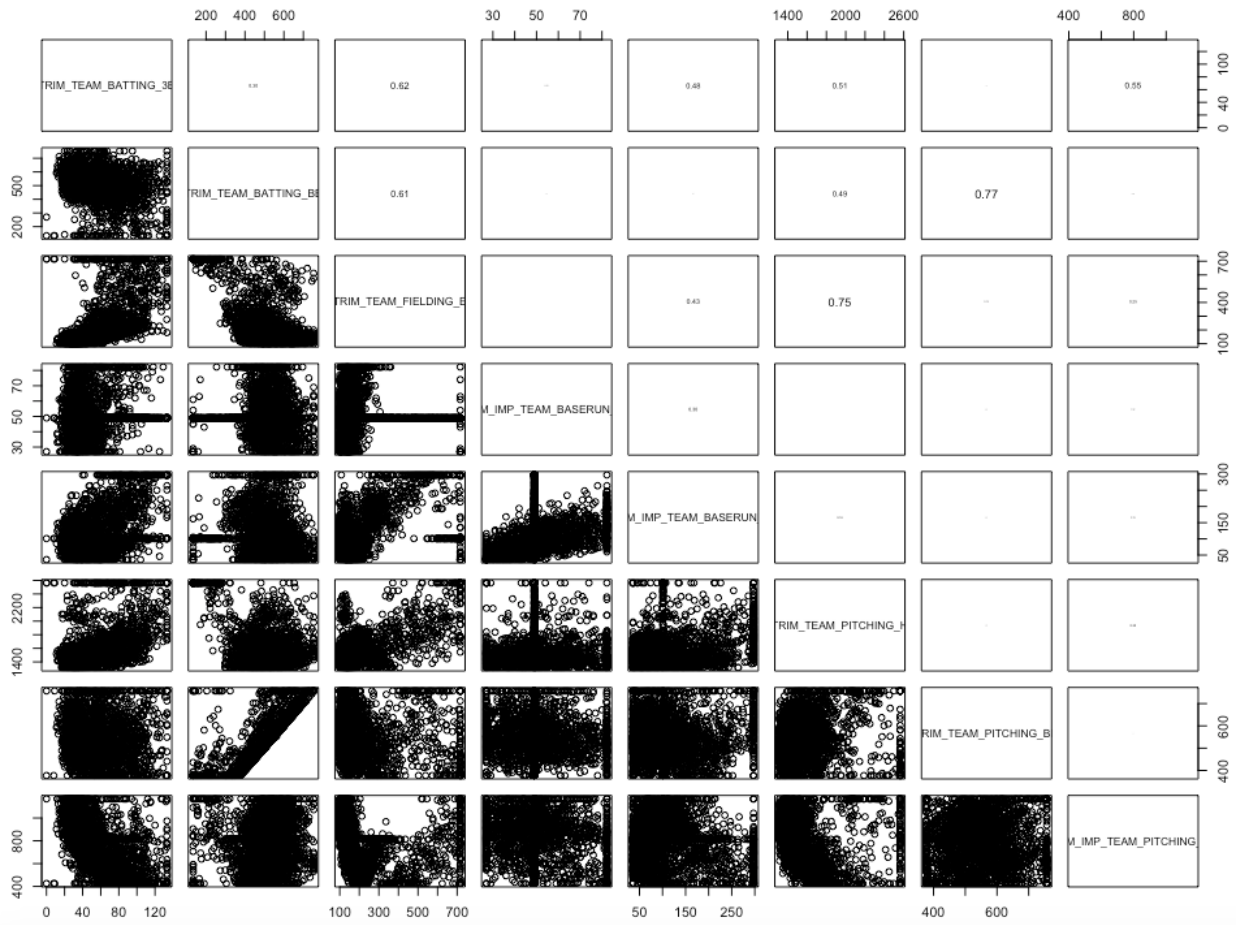
By the end of this section, we have addressed both missing value and outlier issues in the dataset, which is necessary prior to the model development process.

Section 3: Model Development

If there's no missing value or outlier issue with a variable, that variable will be entered into the model development as it is. If there's a missing value issue with a variable, the flag and imputed variables will be used to build the model instead of the original variable. If a variable has outlier issue, the trimmed variable will be included in the model building process instead of the initial variable. For example, TEAM_BASERUN_CS has both missing value and outlier issues. Therefore, the original variable will not be used in this section. Instead the new flag and imputed variables M_TEAM_BASERUN_CS and TRIM_IMP_TEAM_BASERUN_CS will be used.



The correlation plot above shows that we may have multicollinearity issue in the dataset, especially the strong correlation between TEAM_BATTING_HR and TEAM_PITCHING_HR with a 0.97 almost perfect correlation number. We will visit this topic when we start building the model by observing the VIF value.



The correlation plot above also alarms us the 0.75 correlation number between TRIM_TEAM_FIELDING_E and TRIM_TEAM_PITCHING_H, which may cause multicollinearity issue later on during the model development process.

The model development process will start with three models using three variable selection method: stepwise, backward, and forward. The forward selection method starts with an equation containing no variable, only a constant term. It looks at each variable in the dataset and start adding one at a time to the model if the variable has a significant correlation with the response variable TARGET_WINS. In contrast, the backward selection method starts with a full equation including all predictor variables in the model. Then it drops one variable at a time if the variable doesn't have any significant correlation with the response variable. The stepwise method is a combination of the two where it adds and drops variables at the same time to get the right variables in the model with significant correlation with the response variable.

Subsection 3.1: Stepwise Model #1

Using the stepwise selection method, we have the following results.

```
Call:
lm(formula = TARGET_WINS ~ INDEX + TEAM_BATTING_HR + TEAM_PITCHING_HR +
    M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB + M_TEAM_BATTING_SO +
    IMP_TEAM_FIELDING_DP + IMP_TEAM_BATTING_SO + TRIM_TEAM_BATTING_H +
    TRIM_TEAM_BATTING_3B + TRIM_TEAM_BATTING_BB + TRIM_TEAM_FIELDING_E +
    TRIM_IMP_TEAM_BASERUN_CS + TRIM_IMP_TEAM_BASERUN_SB + TRIM_TEAM_PITCHING_H +
    TRIM_IMP_TEAM_PITCHING_SO, data = model)

Residuals:
    Min      1Q  Median      3Q     Max
-63.242  -7.761   0.174   7.865  67.315

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             35.6841746  5.3913601   6.619 4.51e-11 ***
INDEX                   -0.0005326  0.0003564  -1.494   0.1352
TEAM_BATTING_HR         -0.0457097  0.0282151  -1.620   0.1054
TEAM_PITCHING_HR         0.1065187  0.0252130   4.225 2.49e-05 ***
M_TEAM_FIELDING_DP       6.8918688  1.6089036   4.284 1.92e-05 ***
M_TEAM_BASERUN_SB       43.0812845  2.3041673  18.697  < 2e-16 ***
M_TEAM_BATTING_SO        6.3351424  1.5186361   4.172 3.14e-05 ***
IMP_TEAM_FIELDING_DP    -0.0992949  0.0140484  -7.068 2.08e-12 ***
IMP_TEAM_BATTING_SO     -0.0238717  0.0059108  -4.039 5.55e-05 ***
TRIM_TEAM_BATTING_H      0.0514255  0.0037971  13.543  < 2e-16 ***
TRIM_TEAM_BATTING_3B     0.1223420  0.0166114   7.365 2.47e-13 ***
TRIM_TEAM_BATTING_BB     0.0267150  0.0032868   8.128 7.11e-16 ***
TRIM_TEAM_FIELDING_E    -0.0746662  0.0051683 -14.447  < 2e-16 ***
TRIM_IMP_TEAM_BASERUN_CS -0.1036704  0.0229347  -4.520 6.50e-06 ***
TRIM_IMP_TEAM_BASERUN_SB  0.0808986  0.0064491  12.544  < 2e-16 ***
TRIM_TEAM_PITCHING_H    -0.0146699  0.0028613  -5.127 3.19e-07 ***
TRIM_IMP_TEAM_PITCHING_SO  0.0107682  0.0056959   1.891   0.0588 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.35 on 2259 degrees of freedom
Multiple R-squared:  0.3894,    Adjusted R-squared:  0.3851
F-statistic: 90.04 on 16 and 2259 DF,  p-value: < 2.2e-16
```

The overall model is statistically significant at 95% confidence level since its p-value is 2.2e-16, which is less than alpha of 0.05. The adjusted R-squared is 0.3851, which means 38.51% of the variation in TARGET_WINS can be explained by the model. Since R uses the alpha of 0.15 as a threshold to determine the significant correlation of predictor variables, it includes variables with individual p-value higher than 0.05 in the model. For example, TEAM_BATTING_HR has a p-value of 0.1054, which is significant at 89% confidence level but not at 95% confidence level. Also, INDEX is an identification variable that shouldn’t be included in the model. Therefore, we will remove these variables and rerun the regression model based on the results above.

```
Call:
lm(formula = model$TARGET_WINS ~ model$TEAM_PITCHING_HR + model$M_TEAM_FIELDING_
DP +
    model$M_TEAM_BASERUN_SB + model$M_TEAM_BATTING_SO + model$IMP_TEAM_FIELDING_
DP +
    model$IMP_TEAM_BATTING_SO + model$TRIM_TEAM_BATTING_H + model$TRIM_TEAM_BATT
ING_3B +
    model$TRIM_TEAM_BATTING_BB + model$TRIM_TEAM_FIELDING_E +
    model$TRIM_IMP_TEAM_BASERUN_CS + model$TRIM_IMP_TEAM_BASERUN_SB +
    model$TRIM_TEAM_PITCHING_H + model$TRIM_IMP_TEAM_PITCHING_SO)

Residuals:
    Min      1Q  Median      3Q     Max
-63.078  -7.810   0.160   7.707  66.935

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  34.898297   5.379472   6.487 1.07e-10 ***
model$TEAM_PITCHING_HR        0.067848   0.008700   7.798 9.51e-15 ***
model$M_TEAM_FIELDING_DP      6.925695   1.609775   4.302 1.76e-05 ***
model$M_TEAM_BASERUN_SB      41.843653   2.172579  19.260  < 2e-16 ***
model$M_TEAM_BATTING_SO       6.690207   1.510636   4.429 9.93e-06 ***
model$IMP_TEAM_FIELDING_DP   -0.100069   0.014052  -7.121 1.43e-12 ***
model$IMP_TEAM_BATTING_SO    -0.027077   0.005635  -4.805 1.65e-06 ***
model$TRIM_TEAM_BATTING_H     0.049592   0.003604  13.761  < 2e-16 ***
model$TRIM_TEAM_BATTING_3B    0.127181   0.016197   7.852 6.27e-15 ***
model$TRIM_TEAM_BATTING_BB    0.026716   0.003266   8.180 4.67e-16 ***
model$TRIM_TEAM_FIELDING_E   -0.074771   0.005166 -14.473  < 2e-16 ***
model$TRIM_IMP_TEAM_BASERUN_CS -0.098158  0.022718  -4.321 1.62e-05 ***
model$TRIM_IMP_TEAM_BASERUN_SB  0.080052  0.006438  12.435  < 2e-16 ***
model$TRIM_TEAM_PITCHING_H   -0.013200   0.002696  -4.897 1.04e-06 ***
model$TRIM_IMP_TEAM_PITCHING_SO 0.013352  0.005493   2.430   0.0152 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.36 on 2261 degrees of freedom
Multiple R-squared:  0.388,     Adjusted R-squared:  0.3842
F-statistic: 102.4 on 14 and 2261 DF,  p-value: < 2.2e-16
```

By doing so, above is the results of the model. Overall, the model is significant at 95% confidence level since its p-value is 2.2e-16, which is less than alpha of 0.05. Each of the predictor is significant at 95% confidence level as well with p-values less than 0.05 alpha. The adjusted R-squared drops slightly to 0.3842, which means that 38.42% of the variation in TARGET_WINS can be explained by the model. The betas for each predictor show the marginal change in the response variable associated with the predictor. For example, TRIM_IMP_TEAM_PITCHING_SO beta of 0.013352 means that for every one unit increase in strikeout by pitchers, the team's total win in the season will increase by 0.01. The M_TEAM_BASERUN_SB beta of 41.843653 means that when there's a missing value, the total win will increase by 41.84, which is an interesting finding that might need to be confirmed with industry expert.

As mentioned earlier, multicollinearity may be a big issued in the model development process as we see correlation among predictors in the correlation plot. One technical metric to determine multicollinearity is VIF value. If VIF value is 0, it means there's absolutely no correlation

between the variables. The higher the VIF value, the stronger the correlation. The industry benchmark for multicollinearity detection is the VIF of 10. In other words, if a variable has a VIF above 10, it's highly correlated with other variables.

model$TEAM_PITCHING_HR	4.235266
model$M_TEAM_FIELDING_DP	4.241224
model$M_TEAM_BASERUN_SB	3.814096
model$M_TEAM_BATTING_SO	1.455194
model$IMP_TEAM_FIELDING_DP	1.768045
model$IMP_TEAM_BATTING_SO	27.898615
model$TRIM_TEAM_BATTING_H	3.435806
model$TRIM_TEAM_BATTING_3B	2.854030
model$TRIM_TEAM_BATTING_BB	2.223363
model$TRIM_TEAM_FIELDING_E	11.581986
model$TRIM_IMP_TEAM_BASERUN_CS	1.355810
model$TRIM_IMP_TEAM_BASERUN_SB	2.997585
model$TRIM_TEAM_PITCHING_H	10.256083
model$TRIM_IMP_TEAM_PITCHING_SO	20.368285

From the table above, TRIM_IMP_TEAM_PITCHING_SO and IMP_TEAM_BATTING_SO have severe multicollinearity issue with VIF above 20. In addition, TRIM_TEAM_PITCHING_H and TRIM_TEAM_FIELDING_E have moderate multicollinearity issue with VIF above 10. The remaining predictors don't have multicollinearity concern since their VIF values are below 10.

Subsection 3.2: Backward Model #2

Using the backward selection method, we have the following result.

```
Call:
lm(formula = model$TARGET_WINS ~ model$TEAM_PITCHING_HR + model$M_TEAM_FIELDING_
DP +
    model$M_TEAM_BASERUN_SB + model$M_TEAM_BATTING_SO + model$IMP_TEAM_FIELDING_
DP +
    model$IMP_TEAM_BATTING_SO + model$TRIM_TEAM_BATTING_H + model$TRIM_TEAM_BATT
ING_3B +
    model$TRIM_TEAM_BATTING_BB + model$TRIM_TEAM_FIELDING_E +
    model$TRIM_IMP_TEAM_BASERUN_CS + model$TRIM_IMP_TEAM_BASERUN_SB +
    model$TRIM_TEAM_PITCHING_H + model$TRIM_IMP_TEAM_PITCHING_SO)

Residuals:
    Min      1Q  Median      3Q     Max
-63.078  -7.810   0.160   7.707  66.935

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  34.898297   5.379472   6.487 1.07e-10 ***
model$TEAM_PITCHING_HR        0.067848   0.008700   7.798 9.51e-15 ***
model$M_TEAM_FIELDING_DP      6.925695   1.609775   4.302 1.76e-05 ***
model$M_TEAM_BASERUN_SB      41.843653   2.172579  19.260  < 2e-16 ***
model$M_TEAM_BATTING_SO       6.690207   1.510636   4.429 9.93e-06 ***
model$IMP_TEAM_FIELDING_DP   -0.100069   0.014052  -7.121 1.43e-12 ***
model$IMP_TEAM_BATTING_SO    -0.027077   0.005635  -4.805 1.65e-06 ***
model$TRIM_TEAM_BATTING_H     0.049592   0.003604  13.761  < 2e-16 ***
model$TRIM_TEAM_BATTING_3B    0.127181   0.016197   7.852 6.27e-15 ***
model$TRIM_TEAM_BATTING_BB    0.026716   0.003266   8.180 4.67e-16 ***
model$TRIM_TEAM_FIELDING_E   -0.074771   0.005166 -14.473  < 2e-16 ***
model$TRIM_IMP_TEAM_BASERUN_CS -0.098158   0.022718  -4.321 1.62e-05 ***
model$TRIM_IMP_TEAM_BASERUN_SB  0.080052   0.006438  12.435  < 2e-16 ***
model$TRIM_TEAM_PITCHING_H   -0.013200   0.002696  -4.897 1.04e-06 ***
model$TRIM_IMP_TEAM_PITCHING_SO  0.013352   0.005493   2.430   0.0152 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.36 on 2261 degrees of freedom
Multiple R-squared:  0.388,     Adjusted R-squared:  0.3842
F-statistic: 102.4 on 14 and 2261 DF,  p-value: < 2.2e-16
```

This is the same as the result we have in model #1 using stepwise selection method.

Subsection 3.3: Forward Method #3

Using the forward selection method, we have the following result.

```
Call:
lm(formula = model$TARGET_WINS ~ model$TEAM_PITCHING_HR + model$M_TEAM_FIELDING_
DP +
    model$M_TEAM_BASERUN_SB + model$M_TEAM_BATTING_SO + model$IMP_TEAM_FIELDING_
DP +
    model$IMP_TEAM_BATTING_SO + model$TRIM_TEAM_BATTING_H + model$TRIM_TEAM_BATT
ING_3B +
    model$TRIM_TEAM_BATTING_BB + model$TRIM_TEAM_FIELDING_E +
    model$TRIM_IMP_TEAM_BASERUN_CS + model$TRIM_IMP_TEAM_BASERUN_SB +
    model$TRIM_TEAM_PITCHING_H + model$TRIM_IMP_TEAM_PITCHING_SO)

Residuals:
    Min      1Q  Median      3Q     Max
-63.078  -7.810   0.160   7.707  66.935

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  34.898297   5.379472   6.487 1.07e-10 ***
model$TEAM_PITCHING_HR        0.067848   0.008700   7.798 9.51e-15 ***
model$M_TEAM_FIELDING_DP      6.925695   1.609775   4.302 1.76e-05 ***
model$M_TEAM_BASERUN_SB      41.843653   2.172579  19.260  < 2e-16 ***
model$M_TEAM_BATTING_SO       6.690207   1.510636   4.429 9.93e-06 ***
model$IMP_TEAM_FIELDING_DP   -0.100069   0.014052  -7.121 1.43e-12 ***
model$IMP_TEAM_BATTING_SO    -0.027077   0.005635  -4.805 1.65e-06 ***
model$TRIM_TEAM_BATTING_H     0.049592   0.003604  13.761  < 2e-16 ***
model$TRIM_TEAM_BATTING_3B    0.127181   0.016197   7.852 6.27e-15 ***
model$TRIM_TEAM_BATTING_BB    0.026716   0.003266   8.180 4.67e-16 ***
model$TRIM_TEAM_FIELDING_E   -0.074771   0.005166 -14.473  < 2e-16 ***
model$TRIM_IMP_TEAM_BASERUN_CS -0.098158   0.022718  -4.321 1.62e-05 ***
model$TRIM_IMP_TEAM_BASERUN_SB  0.080052   0.006438  12.435  < 2e-16 ***
model$TRIM_TEAM_PITCHING_H   -0.013200   0.002696  -4.897 1.04e-06 ***
model$TRIM_IMP_TEAM_PITCHING_SO  0.013352   0.005493   2.430   0.0152 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.36 on 2261 degrees of freedom
Multiple R-squared:  0.388,     Adjusted R-squared:  0.3842
F-statistic: 102.4 on 14 and 2261 DF,  p-value: < 2.2e-16
```

The backward and forward selection approaches yield the same result, which is also the same as the revised stepwise method. Therefore, we can conclude that we get the same model using three different methods: stepwise, backward, and forward.

Subsection 3.4: VIF Model #4

Model #4 will address the VIF issues identified by the first three models. Specifically, since IMP_TEAM_BATTING_SO has the highest VIF value, we will start model #4 by dropping that from the equation.

```
          model$TEAM_PITCHING_HR          model$M_TEAM_FIELDING_DP
                        4.204179                          4.128343
        model$M_TEAM_BASERUN_SB          model$M_TEAM_BATTING_SO
                        3.676679                          1.449028
     model$IMP_TEAM_FIELDING_DP       model$TRIM_TEAM_BATTING_H
                        1.765605                          3.058534
     model$TRIM_TEAM_BATTING_3B      model$TRIM_TEAM_BATTING_BB
                        2.852876                          2.189362
     model$TRIM_TEAM_FIELDING_E  model$TRIM_IMP_TEAM_BASERUN_CS
                       10.618913                          1.352495
 model$TRIM_IMP_TEAM_BASERUN_SB      model$TRIM_TEAM_PITCHING_H
                        2.922840                          5.455572
model$TRIM_IMP_TEAM_PITCHING_SO
                        2.901464
```

Dropping IMP_TEAM_BATTING_SO significantly reduces the VIF values in other predictors. However, TRIM_TEAM_FIELDING_E still has a VIF value above 10. Thus, we will take an additional step to drop this variable out of the model.

```
          model$TEAM_PITCHING_HR          model$M_TEAM_FIELDING_DP
                        4.000203                          2.602455
        model$M_TEAM_BASERUN_SB          model$M_TEAM_BATTING_SO
                        3.099494                          1.440590
     model$IMP_TEAM_FIELDING_DP       model$TRIM_TEAM_BATTING_H
                        1.726342                          2.976734
     model$TRIM_TEAM_BATTING_3B      model$TRIM_TEAM_BATTING_BB
                        2.772319                          2.086327
 model$TRIM_IMP_TEAM_BASERUN_CS  model$TRIM_IMP_TEAM_BASERUN_SB
                        1.309087                          2.818838
     model$TRIM_TEAM_PITCHING_H model$TRIM_IMP_TEAM_PITCHING_SO
                        4.597373                          2.894778
```

None of the VIF value now is above 10, so we have successfully and completely addressed multicollinearity issue in the model. Below is the result of model #4 after these changes.

```
Call:
lm(formula = model$TARGET_WINS ~ model$TEAM_PITCHING_HR + model$M_TEAM_FIELDING_
DP +
    model$M_TEAM_BASERUN_SB + model$M_TEAM_BATTING_SO + model$IMP_TEAM_FIELDING_
DP +
    model$TRIM_TEAM_BATTING_H + model$TRIM_TEAM_BATTING_3B +
    model$TRIM_TEAM_BATTING_BB + model$TRIM_IMP_TEAM_BASERUN_CS +
    model$TRIM_IMP_TEAM_BASERUN_SB + model$TRIM_TEAM_PITCHING_H +
    model$TRIM_IMP_TEAM_PITCHING_SO, data = model)

Residuals:
    Min      1Q  Median      3Q     Max
-64.506  -8.038   0.436   8.318  62.931

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   9.663028   5.253546   1.839 0.065997 .
model$TEAM_PITCHING_HR        0.090364   0.008835  10.227  < 2e-16 ***
model$M_TEAM_FIELDING_DP     -7.533538   1.317640  -5.717 1.22e-08 ***
model$M_TEAM_BASERUN_SB      28.310585   2.046495  13.834  < 2e-16 ***
model$M_TEAM_BATTING_SO       5.590368   1.570562   3.559 0.000379 ***
model$IMP_TEAM_FIELDING_DP   -0.068939   0.014509  -4.752 2.15e-06 ***
model$TRIM_TEAM_BATTING_H     0.051455   0.003505  14.679  < 2e-16 ***
model$TRIM_TEAM_BATTING_3B    0.088423   0.016681   5.301 1.26e-07 ***
model$TRIM_TEAM_BATTING_BB    0.034385   0.003306  10.401  < 2e-16 ***
model$TRIM_IMP_TEAM_BASERUN_CS -0.037199   0.023326  -1.595 0.110911
model$TRIM_IMP_TEAM_BASERUN_SB  0.058778   0.006523   9.010  < 2e-16 ***
model$TRIM_TEAM_PITCHING_H   -0.014997   0.001886  -7.952 2.86e-15 ***
model$TRIM_IMP_TEAM_PITCHING_SO -0.009731   0.002164  -4.497 7.25e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.92 on 2263 degrees of freedom
Multiple R-squared:  0.3312,    Adjusted R-squared:  0.3277
F-statistic:  93.4 on 12 and 2263 DF,  p-value: < 2.2e-16
```

By dropping the two variables mentioned above, we have a new model #4. However, the predictor TRIM_IMP_TEAM_BASERUN_CS is no longer significant in the model since its p-value is less than 0.05 alpha at 95% confidence level. Therefore, we will drop this variable out of the equation and rerun model #4. Below is the result of the new model output.

```
Call:
lm(formula = model$TARGET_WINS ~ model$TEAM_PITCHING_HR + model$M_TEAM_FIELDING_
DP +
    model$M_TEAM_BASERUN_SB + model$M_TEAM_BATTING_SO + model$IMP_TEAM_FIELDING_
DP +
    model$TRIM_TEAM_BATTING_H + model$TRIM_TEAM_BATTING_3B +
    model$TRIM_TEAM_BATTING_BB + model$TRIM_IMP_TEAM_BASERUN_SB +
    model$TRIM_TEAM_PITCHING_H + model$TRIM_IMP_TEAM_PITCHING_SO,
    data = model)

Residuals:
    Min      1Q  Median      3Q     Max
-64.102  -8.083   0.340   8.296  62.713

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   8.333419   5.188733   1.606    0.108
model$TEAM_PITCHING_HR        0.091864   0.008788  10.453  < 2e-16 ***
model$M_TEAM_FIELDING_DP     -6.858533   1.248230  -5.495 4.35e-08 ***
model$M_TEAM_BASERUN_SB      28.043702   2.040336  13.745  < 2e-16 ***
model$M_TEAM_BATTING_SO       6.055024   1.543824   3.922 9.04e-05 ***
model$IMP_TEAM_FIELDING_DP   -0.071992   0.014387  -5.004 6.05e-07 ***
model$TRIM_TEAM_BATTING_H     0.051148   0.003501  14.609  < 2e-16 ***
model$TRIM_TEAM_BATTING_3B    0.089867   0.016662   5.394 7.62e-08 ***
model$TRIM_TEAM_BATTING_BB    0.034880   0.003292  10.594  < 2e-16 ***
model$TRIM_IMP_TEAM_BASERUN_SB 0.054623  0.005983   9.130  < 2e-16 ***
model$TRIM_TEAM_PITCHING_H   -0.014893   0.001885  -7.899 4.33e-15 ***
model$TRIM_IMP_TEAM_PITCHING_SO -0.009604 0.002163 -4.439 9.46e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.92 on 2264 degrees of freedom
Multiple R-squared:  0.3305,    Adjusted R-squared:  0.3272
F-statistic: 101.6 on 11 and 2264 DF,  p-value: < 2.2e-16
```

Now all predictors are statistically significant at 95% confidence level with their p-values less than 0.05 alpha. Moreover, the overall model is statistically significant at 95% confidence level as well with p-value of 2.2e-16, below 0.05 alpha. The adjusted R-squared drops from the first three models to 0.3272, which means that 32.72% of the variation in TARGET_WINS is explained by the model.

Subsection 3.5: Industry Model #5

Some of the predictors in model #4 don’t match industry knowledge. Thus, model #5 will address this issue. Specifically, the following predictors’ betas contradict baseball industry knowledge. For example, TEAM_PITCHING_HR has a positive beta, which indicates a positive impact on TARGET_WINS. However, according to industry knowledge, it should have a negative impact on total wins.

Predictors	Betas	Industry Knowledge
TEAM_PITCHING_HR	0.091864	Negative impact
IMP_TEAM_FIELDING_DP	-0.071992	Positive impact
TRIM_IMP_TEAM_PITCHING_SO	-0.009604	Positive impact

As a result, in order to match industry knowledge, model #5 will remove these three predictors from the equation. Below is the model output.

```
Call:
lm(formula = model$TARGET_WINS ~ model$M_TEAM_FIELDING_DP + model$M_TEAM_BASERUN
_SB +
    model$M_TEAM_BATTING_SO + model$TRIM_TEAM_BATTING_H + model$TRIM_TEAM_BATTIN
G_3B +
    model$TRIM_TEAM_BATTING_BB + model$TRIM_IMP_TEAM_BASERUN_SB +
    model$TRIM_TEAM_PITCHING_H, data = model)

Residuals:
    Min      1Q  Median      3Q     Max
-58.584  -8.524   0.288   8.730  57.084

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                  -20.554988   3.677182  -5.590 2.55e-08 ***
model$M_TEAM_FIELDING_DP      -8.408647   1.210690  -6.945 4.91e-12 ***
model$M_TEAM_BASERUN_SB       25.415368   1.996387  12.731  < 2e-16 ***
model$M_TEAM_BATTING_SO        5.094446   1.510753   3.372 0.000758 ***
model$TRIM_TEAM_BATTING_H      0.063673   0.003141  20.269  < 2e-16 ***
model$TRIM_TEAM_BATTING_3B     0.025631   0.014257   1.798 0.072345 .
model$TRIM_TEAM_BATTING_BB     0.043445   0.003136  13.855  < 2e-16 ***
model$TRIM_IMP_TEAM_BASERUN_SB 0.048417   0.005443   8.895  < 2e-16 ***
model$TRIM_TEAM_PITCHING_H    -0.013416   0.001927  -6.964 4.32e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.29 on 2267 degrees of freedom
Multiple R-squared:  0.2905,    Adjusted R-squared:  0.288
F-statistic: 116.1 on 8 and 2267 DF,  p-value: < 2.2e-16
```

The adjusted R-squared drops to 0.288, which means that 28.8% of variation in TARGET_WINS can be explained by the model. However, the overall model is still significant at 95% confidence level with p-value of 2.2e-16, less than 0.05 alpha. In addition, all betas' signs match baseball industry knowledge. In model #5, TRIM_MEAN_BATTING_3B is no longer significant 95% confidence level. However, it's still significant at 90% confidence level with p-value less than 0.01. Thus, we will keep this predictor in the model.

```
      model$M_TEAM_FIELDING_DP        model$M_TEAM_BASERUN_SB
                      2.074815                       2.785365
       model$M_TEAM_BATTING_SO      model$TRIM_TEAM_BATTING_H
                      1.258754                       2.257668
    model$TRIM_TEAM_BATTING_3B     model$TRIM_TEAM_BATTING_BB
                      1.912555                       1.772509
model$TRIM_IMP_TEAM_BASERUN_SB     model$TRIM_TEAM_PITCHING_H
                      1.853161                       4.530663
```

We can also confirm that none of the predictor in model #5 has multicollinearity issue since none of them have VIF value greater than 10.

Section 4: Model Selection

Since the first three models using stepwise, backward, and forward selection methods are the same, we will group them under one model as we do model comparison in this section. Specifically, we will use the following three metrics to compare models.

- Adjusted R-squared: R-squared is known as coefficient of determination, which shows the proportion of variability in the response variable that is explained by the model. As we add more predictors into the model, R-squared will automatically increase. Thus, adjusted R-squared is used in multiple linear regression model because it's designed to adjust for the number of predictors in the model. Specifically, adjusted R-squared penalizes the model for adding insignificant predictors into the model. It will only increase if the additional predictor adds explanatory power to the model. The higher adjusted R-squared, the better the model.
- Mean square error (MSE): mean of square of error in the predictive mode. An error is calculated as actual value minus predicted value. If we square all of the error terms and take an average of them, that's the MSE. This metric assesses the accuracy of the model to determine how close the predicted values are to the true values. The smaller the MSE, the better the model.
- Akaike information criterion (AIC): model evaluation metric used for non-nested models. It assesses the model's both accuracy and precision. In other words, similar to adjusted R-squared, it penalizes the model for its complexity as we add more predictors. Similar to MSE, it also evaluates how close the predicted values are to the actual values.

modelcomp	adjr2comp	msecomp	AICcomp
"model 1/2/3"	"0.38"	"151.78"	"17922.06"
"model 4"	"0.33"	"166.06"	"18120.67"
"model5"	"0.29"	"175.96"	"18246.51"

The table above shows the adjusted R-squared, MSE, and AIC for each model in previous section. The model created by using the three variable selection method is the best one since it has the highest adjusted R-squared, the lowest MSE, and lowest AIC. Therefore, if we only use technical metrics to select the best model, the variable selection model is the chosen one. However, if we also consider other factors, this is not necessarily the case. Specifically, this model has too much multicollinearity issue, which means that it may not be reliable. In other words, the multicollinearity issue may inflate the three metrics we use to evaluate models.

That leaves us with model #4 and model #5. Between the two of these, model #4 seems better since it has higher adjusted R-squared, lower MSE, and lower AIC. However, the problem with model #4 is that some of its predictors don't match industry knowledge. On the other hand, although model #5 is the worst model using the three technical evaluation metrics, it is the best model because of the following reasons.

- All of its predictors match industry knowledge
- The model overall is still significant at 95% confidence level

- The train dataset is not clean with many missing values and outliers. Therefore, since the beginning, the dataset used in the analysis is not completely good and reliable. There are many assumptions made during the data preparation process to clean up the dataset by addressing missing value and outlier issue. Therefore, industry knowledge is more reliable than the data in this case.
- The analyst doesn't know anything about baseball. Therefore, it's better to listen to industry expert, so industry knowledge is important during the model selection process in this section.
- Model #5 has a combination of data science and industry knowledge whereas the other models don't.

As a result, the chosen model for this project is model #5. Here's the model equation to predict TARGET_WINS

P_TARGET_WINS = -20.554988 + M_TEAM_FIELDING_DP*-8.408647 + M_TEAM_BASERUN_SB*25.415368 + M_TEAM_BATTING_SO*5.094446 + TRIM_TEAM_BATTING_H*0.063673 + TRIM_TEAM_BATTING_3B*0.025631 + TRIM_TEAM_BATTING_BB*0.043445 + TRIM_IMP_TEAM_BASERUN_SB*0.048417 + TRIM_TEAM_PITCHING_H*-0.013416

CONCLUSION

In summary, the money ball OLS regression project starts with a train dataset of 2276 observations and 17 variables. The first stage of the project is data exploration, which identifies six variables with missing value issues and nine variables with outlier issues. The second stage of the project is data preparation, which is divided into two parts. The first half imputes variables with missing values. Specifically, a flag variable and an imputed variable are created for each of the six variables with missing values. If the variable has normality issue, the median is used to replace the missing records. If the variable doesn't have normality issue, the mean is used to replace the missing records. The second half of the section addresses outlier issues. For variables with moderate outlier issues, 99% trim is utilized. For variables with more severe outlier issues, a combination of z-standardization and 95% trim is used in the transformation process.

The third stage of the project is model development. The first three models using stepwise, backward, and forward selection methods yield the same result. However, this model contains variables with too much multicollinearity. Therefore, a fourth model is introduced to address the multicollinearity issue. However, this model doesn't match industry knowledge. Thus, a fifth model is introduced to address that problem. The final stage of the project is model evaluation and selection. Three technical metrics are used to evaluate the models: adjusted R-squared, MSE, and AIC. Although model #5 is ranked worst in all three metrics, it's ultimately chosen for the project because it doesn't have multicollinearity issue and also matches industry knowledge. The predictive model yields the following equation that can be applied to score new data and forecast the number of wins.

P_TARGET_WINS = -20.554988 + M_TEAM_FIELDING_DP*-8.408647 + M_TEAM_BASERUN_SB*25.415368 + M_TEAM_BATTING_SO*5.094446 + TRIM_TEAM_BATTING_H*0.063673 + TRIM_TEAM_BATTING_3B*0.025631 + TRIM_TEAM_BATTING_BB*0.043445 + TRIM_IMP_TEAM_BASERUN_SB*0.048417 + TRIM_TEAM_PITCHING_H*-0.013416

If future researchers would like to use the equation above, they must keep in mind the parameters of all variables in the train dataset. For example, the TARGET_WINS in the dataset is bound between 0 and 162 for a 162-game season. Thus, if they want to predict wins for more than one season, the predicted value will be over 162, which is inappropriate to use for this equation.

Finally, if future researchers want to continue developing this project, below are some recommendations.

- Collect better data with fewer missing values
- If the missing value issues can't be fixed during the data collection process, collect more data to increase the sample size
- Speak to industry knowledge to understand the nuances of each variable
- The adjusted R-squared for all models in this project are low, so if possible, add more variables to the dataset to develop more accurate model