Moneyball OLS Regression Project

Introduction

The purpose of this study is to create a predictive model of baseball team wins. A data set named "moneyball" consisting of 2276 observations of 14 continuous variables including 6 batting predictors, 2 baserunning predictors, 4 pitching predictors, and 2 fielding predictors is used to model the response variable TARGET_WINS. The model is prepared using linear regression in the R language.

First, EDA is conducted on the TRAINING data set. The EDA will look at the distributions of the variables, correlations, missing values, outliers, etc. After this analysis, the data will be prepared for modeling by imputing missing values, truncating or transforming variables, and creating new metrics. Linear regression is used for modeling, and various methods of variable selection will be used. After iterating through models and analysis, a champion model is selected. A scoring program is then written using the champion model to apply to a TEST set of 259 observations and output predicted team win totals. The accuracy of the scoring program will be assessed.

1. Data Exploration

The moneyball data set is visualized using summary().

```
> summary(moneyball)
  TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
 Min.   :  0.00   Min.   : 891   Min.   : 69.0   Min.   :  0.00   Min.   :  0.00   Min.   :  0.0  
 1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0   1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0  
 Median : 82.00   Median :1454   Median :238.0   Median : 47.00   Median :102.00   Median :512.0  
 Mean   : 80.79   Mean   :1469   Mean   :241.2   Mean   : 55.25   Mean   : 99.61   Mean   :501.6  
 3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0   3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0  
 Max.   :146.00   Max.   :2554   Max.   :458.0   Max.   :223.00   Max.   :264.00   Max.   :878.0  
                                                                                                  
 TEAM_BATTING_SO  TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
 Min.   :   0.0   Min.   :  0.0   Min.   :  0.0   Min.   :29.00    Min.   : 1137   Min.   :  0.0   
 1st Qu.: 548.0   1st Qu.: 66.0   1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419   1st Qu.: 50.0   
 Median : 750.0   Median :101.0   Median : 49.0   Median :58.00    Median : 1518   Median :107.0   
 Mean   : 735.6   Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779   Mean   :105.7   
 3rd Qu.: 930.0   3rd Qu.:156.0   3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682   3rd Qu.:150.0   
 Max.   :1399.0   Max.   :697.0   Max.   :201.0   Max.   :95.00    Max.   :30132   Max.   :343.0   
 NA's   :102      NA's   :131     NA's   :772     NA's   :2085                                     
 TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E  TEAM_FIELDING_DP
 Min.   :   0.0   Min.   :    0.0   Min.   :  65.0   Min.   : 52.0   
 1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0   1st Qu.:131.0   
 Median : 536.5   Median :  813.5   Median : 159.0   Median :149.0   
 Mean   : 553.0   Mean   :  817.7   Mean   : 246.5   Mean   :146.4   
 3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2   3rd Qu.:164.0   
 Max.   :3645.0   Max.   :19278.0   Max.   :1898.0   Max.   :228.0   
                  NA's   :102                        NA's   :286     
> 
```

There are 6 variables with missing values: TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_SO, and TEAM_FIELDING_DP. 92% of the TEAM_BATTING_HBP values are missing, so the variable is removed from the data set. Imputation using only 8% of the observations would be too biased. The percentages of missing values are shown below from the aggr() function.

Variable Count
TEAM_BASERUN_CS 0.33919156
TEAM_FIELDING_DP 0.12565905
TEAM_BASERUN_SB 0.05755712
TEAM_BATTING_SO 0.04481547

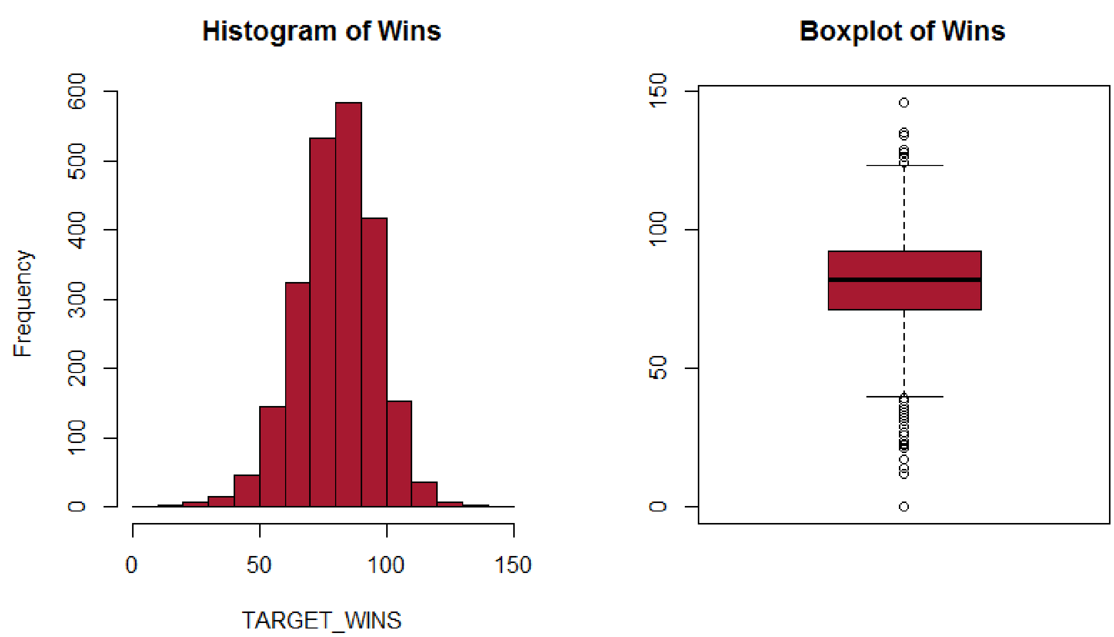
TEAM_PITCHING_SO 0.04481547

Various imputation methods for the 5 variables above will be investigated in the next section. In addition to some missing values there are 10 variables with minimum values of 0. It is not reasonable for a baseball team to have 0 of any of the variables in the moneyball set. Distribution analysis of these variables will show which method of adjustment to make.

There are also 5 variables with unfeasibly high maximum values: TARGET_WINS, TEAM_PITCHING_SO, TEAM_PITCHING_H, TEAM_PITCHING_BB, and TEAM_FIELDING_E. Similarly, these variables will be analyzed for a method of adjustment such as truncation, normalization, or trimming.

The histogram and boxplot for the response variable TARGET_WINS are generated.

Figure 1: Histogram and boxplot of TARGET_WINS



As mentioned in the summary above the outliers on either end of the distribution need to be addressed which is in the next section.

Histograms are generated for the 13 remaining predictor variables and are shown in Appendix 2. The assessment of the summary output is confirmed with 9 of the variables requiring adjustment or transformation for left-tailedness and 4 predictors with right-tailedness.

Next the correlation coefficients between the predictors and TARGET_WINS are calculated. Pairwise-complete correlation is used since the data set has not been imputed yet.

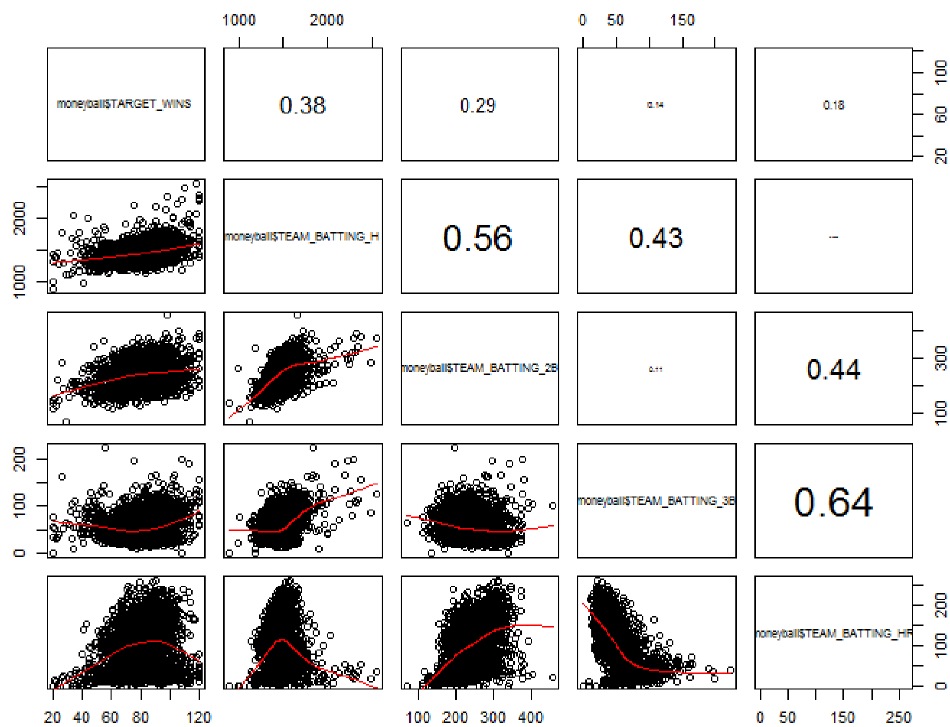
```
TARGET_WINS        1.00000000
TEAM_BATTING_H     0.38067572
TEAM_BATTING_2B    0.28555905
TEAM_BATTING_3B    0.13986105
TEAM_BATTING_HR    0.17937695
TEAM_BATTING_BB    0.23752067
TEAM_BATTING_SO   -0.03064897
```

```
TEAM_BASERUN_SB     0.13415844
TEAM_BASERUN_CS     0.01967601
TEAM_PITCHING_H    -0.10420788
TEAM_PITCHING_HR    0.19115611
TEAM_PITCHING_BB    0.12584803
TEAM_PITCHING_SO   -0.07903061
TEAM_FIELDING_E    -0.17835961
TEAM_FIELDING_DP   -0.03315797
```

None of the coefficients exceed 0.50, and TEAM_PITCHING_BB and TEAM_PITCHING_SO have the opposite coefficient sign than expected (walks seen as good and strikeouts as bad). This means none of the individual predictors will be a great predictor of TARGET_WINS on its own and that the interpretability of any generated model should be considered.

Finally, scatter plot matrices with correlation coefficients are generated for the 3 main types of variables to look at potential multicollinearity concerns as well as opportunities for transformation. First the batting variables are charted against TARGET_WINS.

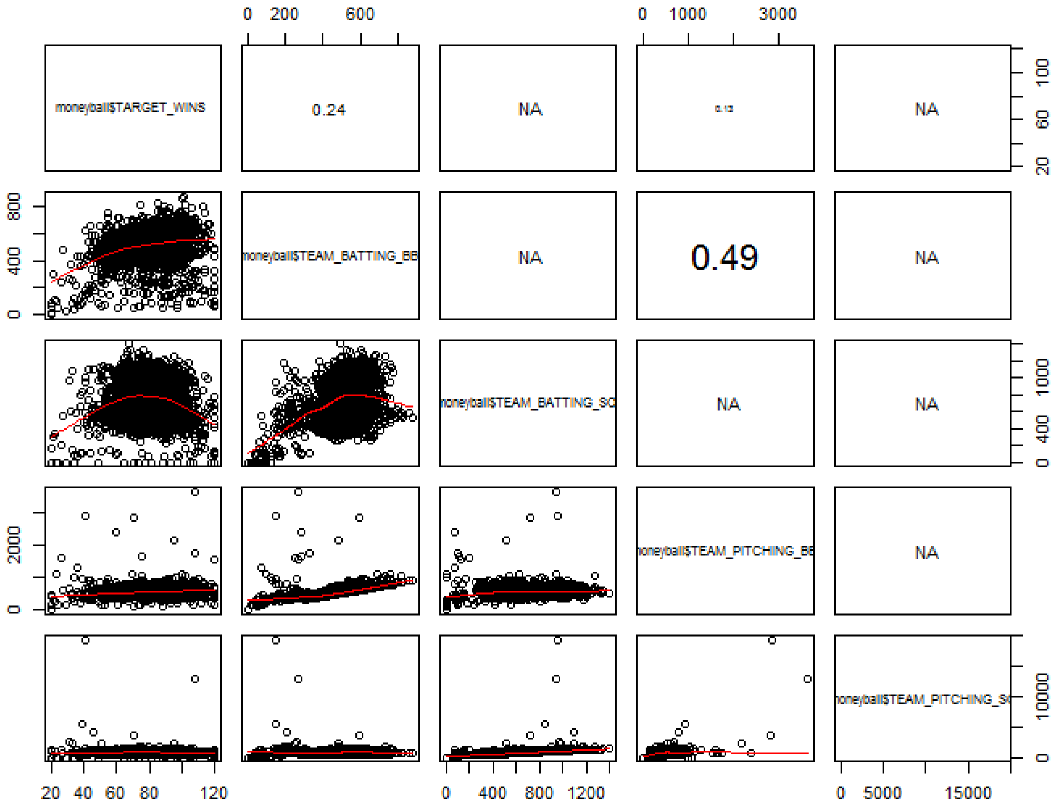
Figure 2: Scatter-correlation matrix of TARGET_WINS and batting predictors



Not surprisingly, some of the batting statistics are highly correlated with each other, such as TEAM_BATTING_3B and TEAM_BATTING_HR. Those 2 predictors would also require transformation to try and straighten the correlation with TARGET_WINS. To account for the correlation between predictors combined hitting metrics such as total hits or slugging should be considered.

Next look at the walk and strikeout predictors.

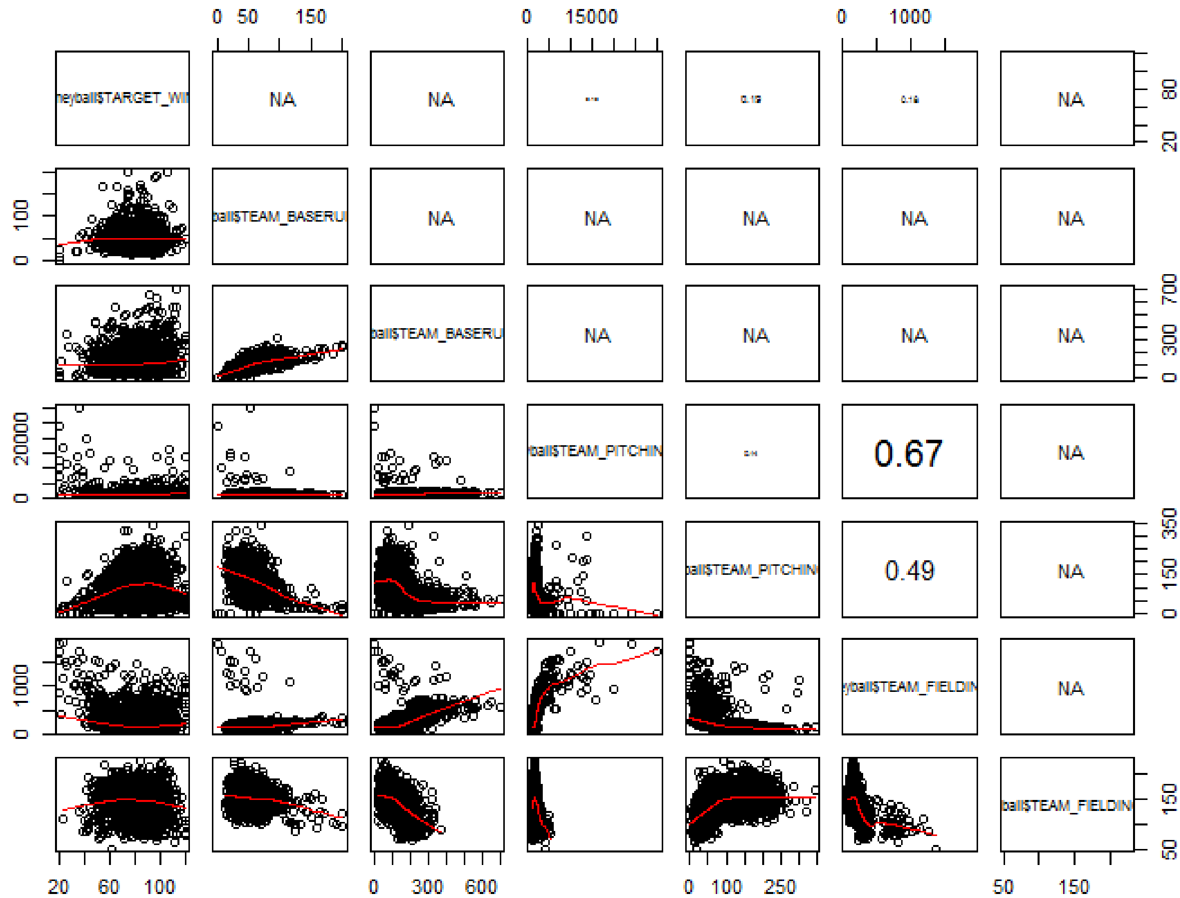
Figure 3: Scatter-correlation matrix of TARGET_WINS and BB/K predictors



The outliers in TEAM_PITCHING_BB and TEAM_PITCHING_SO mask any relationships. TEAM_BATTING_BB is the only single predictor with a relatively strong correlation to TARGET_WINS. A ratio metric of BB to SO should be considered.

The baserunning, fielding, and remaining pitching predictors are looked at next.

Figure 4: Scatter-correlation matrix of TARGET_WINS and baserunning/fielding predictors



TEAM_PITCHING_HR and the FIELDING predictors should be considered for transformations. A ratio of BASERUN predictors (SB/SB+CS) should be tested as well. There is a large correlation between TEAM_FIELDING_E and TEAM_PITCHING_H, but PITCHING_H requires outlier correction first.

Each of the predictors have been anlayzed for missing values, distributions, correlations, and relationships with the response TARGET_WINS. Next the data is prepared for modeling using the suggestions from EDA.

2. Data Preparation

The lowest win totals for a baseball team is 20 in the pre-modern era and 36 in the modern era. The highest win total is 116. A practical domain of {20, 120} is selected for TARGET_WINS and the variable is truncated. The resulting quartiles are shown below.

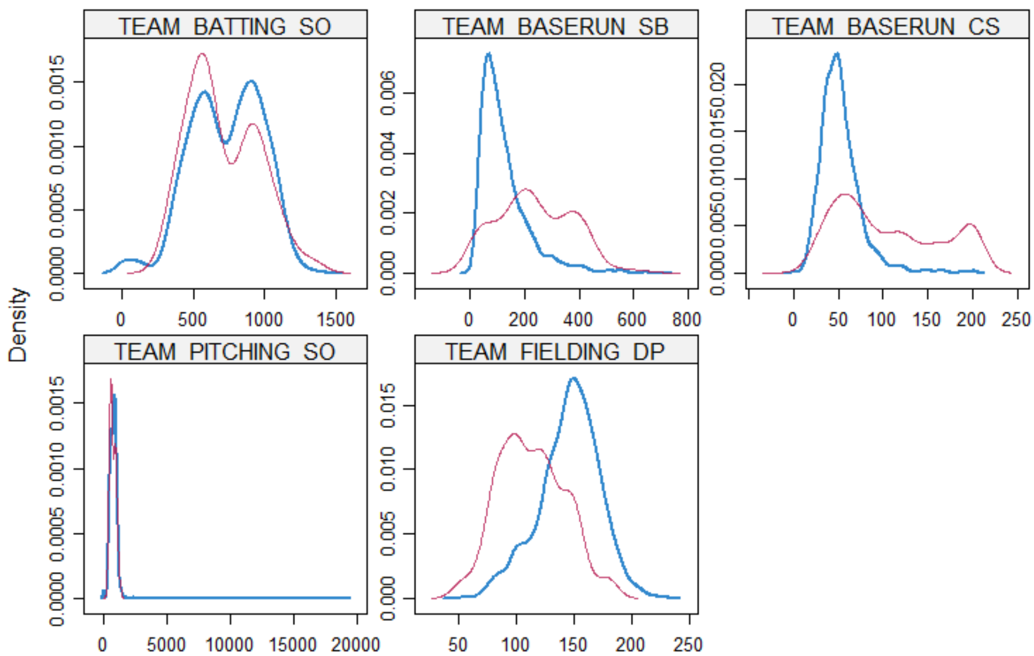
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
20.00	71.00	82.00	80.77	92.00	120.00

Prior to the distribution adjustments, transformations, and metric creations, the missing values for the 5 variables require imputation.

Missing Value Imputation

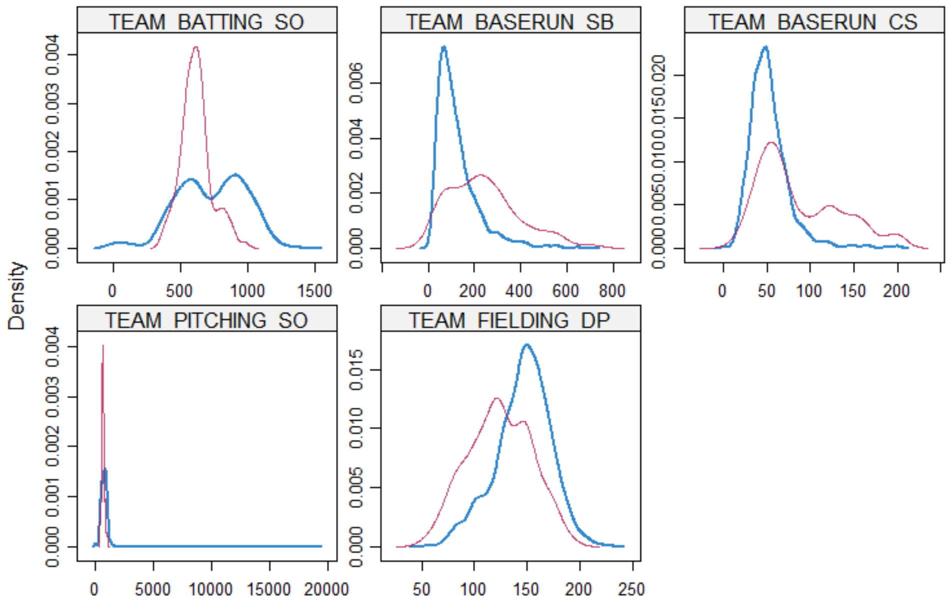
The mice() package in R conducts multiple imputation using various methods, conducts an analysis such as regression on the multiple imputations, then pools the results into a single model. The default method of imputation for continuous variables is predictive mean matching (PMM). For the purposes of creating a predictive scoring program, a single imputation set will be created for each of the 5 variables and converted to an imputed data set that can be joined with the moneyball data set. After exploring various methods, CART decision trees and random forests are chosen for imputation. The variables are imputed using mice() and density plots are generated for visualizing the appropriateness of the imputations.

Figure 5: Density plots of imputed (red) and observed (blue) distributions using CART trees



CART imputation seems appropriate for the 5 variables above excluding the BASERUN variables.

Figure 6: Density plots of imputed (red) and observed (blue) distributions using random forests



To summarize, BATTING and PITCHING variables are imputed using CART decision trees and the BASERUN and FIELDING variables are imputed using random forests. In general, random forests should provide more accurate results for prediction purposes than a single CART tree since it runs random CART trees with various states of pruning which reduces the predictive error. For each of these predictors, an imputed variable is added to the data set as well as a flag variable indicating whether the value in the original predictor was imputed or not. The aggr() function is run again to show that no missing values remain in the imputed variables.

```
IMP_TEAM_BATTING_SO 0.00000000
IMP_TEAM_PITCHING_SO 0.00000000
IMP_TEAM_BASERUN_CS 0.00000000
IMP_TEAM_BASERUN_SB 0.00000000
IMP_TEAM_FIELDING_DP 0.00000000
```

With no missing values remaining in the moneyball data set, the remaining variables are truncated.

Outlier Truncation

First the 9 remaining variables with left-tailed distributions are analyzed at the 1st and 5th percentiles as potential truncation cutoffs.

```
> quantile(moneyball2$TEAM_BATTING_3B, c(0.01, 0.05))
1% 5%
17 23
> quantile(moneyball2$TEAM_BATTING_HR, c(0.01, 0.05))
 1%     5%
 4.75 14.00
> quantile(moneyball2$TEAM_BATTING_BB, c(0.01, 0.05))
    1%      5%
 79.00 248.25
> quantile(moneyball2$IMP_TEAM_BATTING_SO, c(0.01, 0.05))
    1%      5%
 72.00 361.75
> quantile(moneyball2$IMP_TEAM_BASERUN_SB, c(0.01, 0.05))
1% 5%
```

```
21 35
> quantile(moneyball2$IMP_TEAM_BASERUN_CS, c(0.01, 0.05))
1% 5%
18 25
> quantile(moneyball2$TEAM_PITCHING_HR, c(0.01, 0.05))
1% 5%
 8 18
> quantile(moneyball2$TEAM_PITCHING_BB, c(0.01, 0.05))
 1%  5%
240 377
> quantile(moneyball2$IMP_TEAM_PITCHING_SO, c(0.01, 0.05))
 1%  5%
241 420
```

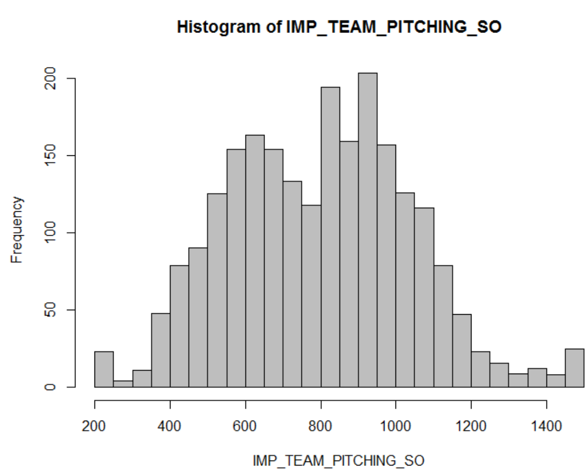
The predictors above are truncated on the lower end at either the 0.01 or 0.05 quantile based on perceived feasibility.

Next the 4 variables identified with high outliers in Section 1 have the 95th and 99th percentiles calculated.

```
> quantile(moneyball2$IMP_TEAM_PITCHING_SO, c(0.95, 0.99))
    95%     99%
1170.50 1461.75
> quantile(moneyball2$TEAM_PITCHING_BB, c(0.95, 0.99))
95% 99%
757 921
> quantile(moneyball2$TEAM_FIELDING_E, c(0.95, 0.99))
 95%  99%
 716 1228
> quantile(moneyball2$TEAM_PITCHING_H, c(0.95, 0.99))
 95%  99%
2563 7054
```

Of these predictors, the first 3 are truncated at the 99th percentile while TEAM_PITCHING_H was decided to have a hard cutoff at 3000. As a spot-check on the effectiveness of the outlier corrections the histogram for IMP_TEAM_PITCHING_SO is regenerated.

Figure 7: Histogram of TEAM_PITCHING_SO post-truncation



There is a large improvement in the feasibility of the distribution post-truncation. Next the new metrics identified during scatter matrix analysis are created. First the walk-strikeout ratios are created, TEAM_BATTING_BB_SO and TEAM_PITCHING_BB_SO. A slugging metric is approximated as a ratio of extra base hits to total hits (2B+3B+HR)/H. Next the stealing success metric is created by dividing the number of stolen bases by the combined stolen bases and caught stealing. A WHIP (walks hits per inning pitched) is approximated using TEAM_PITCHING_BB + TEAM_PITCHING_H / 1458, where 1458 is the number of innings in a 162-game season assuming each game is 9 innings long.

Prior to variable transformation, a model is fit using the data set of truncated, imputed predictors.

3. Build Models

Model 1: STEPWISE

The first model is fit by creating a subset of predictors removing INDEX and the unimputed variables using stepwise variable selection. AIC is used as the cutoff metric instead of adjusted R^2 as it is a more useful metric for prediction models. The summary() of STEPWISE model is shown below, followed by a table of the variance inflation factors (VIFs) for the various predictors.

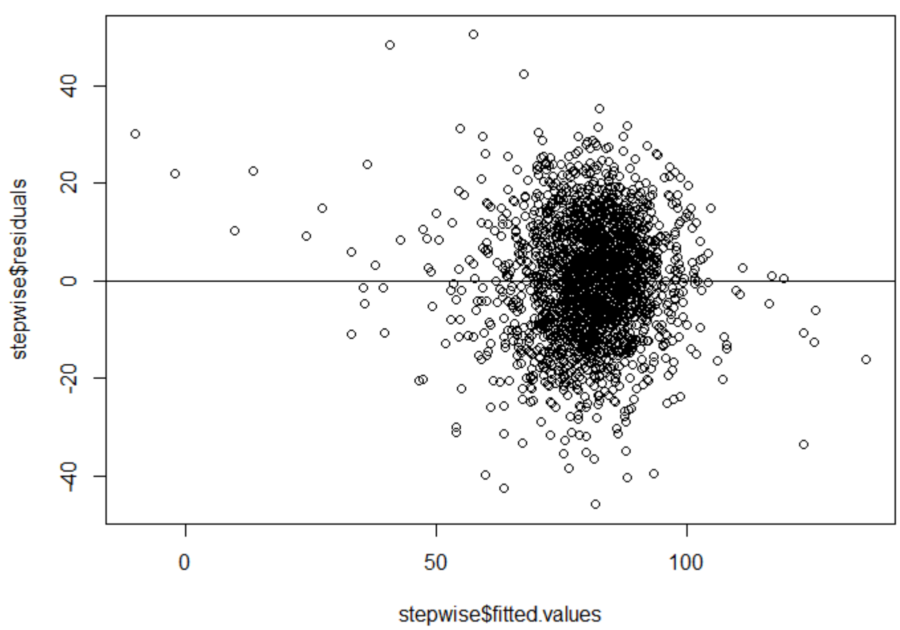
```
Residuals:
    Min      1Q  Median      3Q     Max 
-45.883  -7.778   0.124   8.056  50.489 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)          -72.984894  14.716968  -4.959 7.60e-07 ***
TEAM_BATTING_H         0.107306   0.009681  11.084  < 2e-16 ***
TEAM_BATTING_2B       -0.283834   0.038622  -7.349 2.78e-13 ***
TEAM_BATTING_3B       -0.169045   0.040662  -4.157 3.34e-05 ***
TEAM_BATTING_HR       -0.281275   0.045249  -6.216 6.05e-10 ***
TEAM_BATTING_BB        0.075058   0.006131  12.242  < 2e-16 ***
TEAM_PITCHING_H        0.009659   0.002371   4.073 4.79e-05 ***
TEAM_PITCHING_HR       0.059473   0.024462   2.431 0.015124 *  
TEAM_FIELDING_E       -0.064816   0.003561 -18.201  < 2e-16 ***
IMP_TEAM_BATTING_SO   -0.012541   0.005276  -2.377 0.017530 *  
IMP_TEAM_PITCHING_SO  -0.020452   0.003804  -5.376 8.40e-08 ***
IMP_TEAM_BASERUN_CS    0.021707   0.008423   2.577 0.010028 *  
IMP_TEAM_BASERUN_SB    0.037113   0.004370   8.493  < 2e-16 ***
IMP_TEAM_FIELDING_DP  -0.082395   0.012174  -6.768 1.66e-11 ***
M_TEAM_BASERUN_CS      2.009632   0.844494   2.380 0.017410 *  
M_TEAM_FIELDING_DP     5.512504   1.422340   3.876 0.000109 ***
M_TEAM_BATTING_SO      9.861650   1.502361   6.564 6.47e-11 ***
M_TEAM_BASERUN_SB     26.870142   1.987925  13.517  < 2e-16 ***
TEAM_BATTING_BB_SO   -29.048374   3.398568  -8.547  < 2e-16 ***
TEAM_BATTING_SLUG    397.446398  58.735822   6.767 1.67e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.81 on 2256 degrees of freedom
Multiple R-squared:  0.4289,	Adjusted R-squared:  0.4241 
F-statistic: 89.18 on 19 and 2256 DF,  p-value: < 2.2e-16
```

TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B	TEAM_BATTING_HR
31.949	53.27311	20.64222	120.712
TEAM_PITCHING_H	TEAM_PITCHING_HR	TEAM_FIELDING_E	IMP_TEAM_BATTING_SO
12.79014	36.07399	9.509457	24.53487

IMP_TEAM_BASERUN_CS	IMP_TEAM_BASERUN_SB	IMP_TEAM_FIELDING_DP	M_TEAM_BASERUN_CS
2.060714	2.574491	1.832867	2.607442
M_TEAM_BATTING_SO	M_TEAM_BASERUN_SB	TEAM_BATTING_BB_SO	TEAM_BATTING_SLUG
1.576031	3.496673	14.19949	131.3977

STEPWISE showed 19 of the 25 predictors to be significant with an adjusted R^2 of 0.42. There is significant multicollinearity in many of the batting and pitching predictors that have other metrics in the model derived from them such as TEAM_BATTING_SLUG, however, so in its current state STEPWISE should not be used for predictive purposes for fears of bias in the coefficients. In addition, this multicollinearity is affecting the interpretability of the model since the 2B, 3B, and HR predictors have negative coefficients due to the relative effects of the extremely high and positive SLUG coefficient. Generally speaking, a cutoff of 10 is used for VIF to indicate multicollinearity, and TEAM_BATTING_SLUG has a VIF of 131! An overall residual plot against the fitted values for STEPWISE is generated to assess goodness of fit (GOF).

Figure 8: Residual plot of STEPWISE



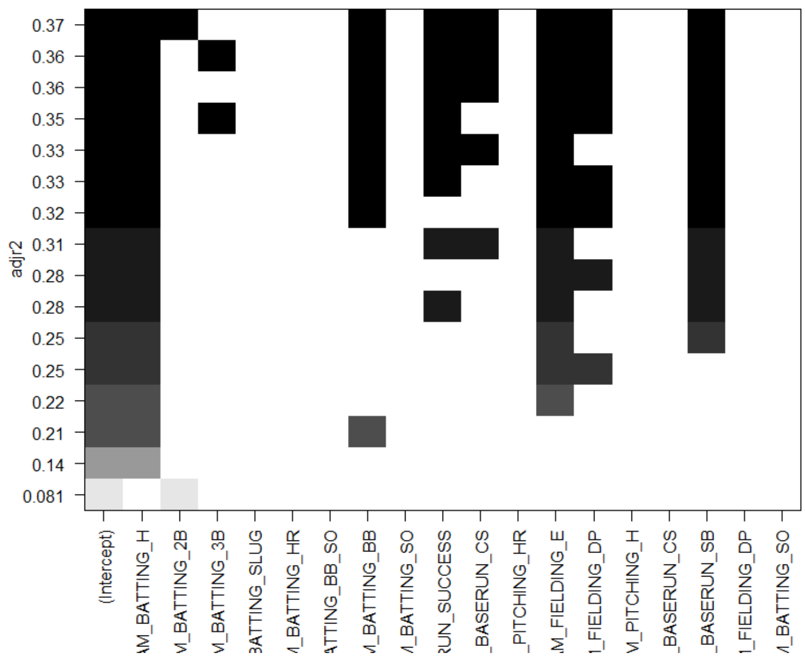
The circular or "double bow" shape of residuals indicates transformation in either the response, predictors, or both is recommended as the assumption of homoscedasticity in the error term is violated which can affect the accuracy of the model when used for prediction.

Using the results of previous EDA, the residual analysis, and additional EDA conducted on STEPWISE results, some regressors are transformed. TEAM_BATTING_SLUG and TEAM_BATTING_3B are transformed using the natural log while TEAM_BATTING_HR, TEAM_PITCHING_HR, and IMP_TEAM_FIELDING_DP are transformed using the square root. This is to "straighten" the scatter plots between these variables and the response.

Model 2: SUBSET

All subsets regression is conducted on a chosen subset of variables based on the results of variable transformation and STEPWISE.

Figure 9: Subset plot for variable selection from SUBSET



A model is fit using the recommendations of the subset plot above.

```
Residuals:
    Min      1Q  Median      3Q     Max 
-48.980  -8.137   0.170   7.959  50.102 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)               -6.904800   4.849537  -1.424    0.155    
TEAM_BATTING_H             0.066645   0.002634  25.306  < 2e-16 ***
TEAM_BATTING_2B           -0.051220   0.008125  -6.304 3.48e-10 ***
TEAM_BATTING_BB            0.035138   0.003201  10.978  < 2e-16 ***
TEAM_BASERUN_SUCCESS      27.921981   2.751741  10.147  < 2e-16 ***
IMP_TEAM_BASERUN_CS        0.067153   0.008222   8.168 5.16e-16 ***
TEAM_FIELDING_E           -0.047708   0.002240 -21.302  < 2e-16 ***
SQRT_IMP_TEAM_FIELDING_DP -2.332923   0.279500  -8.347  < 2e-16 ***
M_TEAM_BASERUN_SB         25.218476   1.628833  15.483  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.39 on 2267 degrees of freedom
Multiple R-squared:  0.3683,	Adjusted R-squared:  0.3661 
F-statistic: 165.2 on 8 and 2267 DF,  p-value: < 2.2e-16
```

The multicollinearity issues from STEPWISE are not present in SUBSET (since SLUG is removed).

TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_BB
2.147864	2.14207	1.810775
TEAM_BASERUN_SUCCESS	IMP_TEAM_BASERUN_CS	TEAM_FIELDING_E
1.193591	1.394471	3.416902
SQRT_IMP_TEAM_FIELDING_DP	M_TEAM_BASERUN_SB	
1.706428	2.132677	

Model 3: CHECK

Taking lessons from the first two methods, a third model is created in hopes of improving the performance metrics of SUBSET while resolving the multicollinearity of STEPWISE. Interaction terms are also included in this custom model CHECK for SQRT_TEAM_BATTING_HR, SQRT_TEAM_BATTING_SLUG, and IMP_TEAM_PITCHING_SO to see if the high correlation between the 3 predictors can be accounted for.

```
Residuals:
    Min      1Q  Median      3Q     Max 
-47.578  -8.168   0.015   8.085  45.370 

Estimate
(Intercept)                                                     20.733756
TEAM_BATTING_H                                                   0.078203
TEAM_BATTING_2B                                                 -0.166170
SQRT_TEAM_BATTING_HR                                           -20.637954
SQRT_TEAM_BATTING_SLUG                                         -93.593577
IMP_TEAM_PITCHING_SO                                            -0.149718
TEAM_FIELDING_E                                                 -0.049285
TEAM_BASERUN_SUCCESS                                            31.380093
IMP_TEAM_BASERUN_CS                                              0.080886
M_TEAM_BASERUN_SB                                               20.437989
TEAM_PITCHING_H                                                  0.013826
TEAM_BATTING_BB                                                  0.036124
TEAM_PITCHING_WHIP                                             -11.947897
SQRT_TEAM_BATTING_HR:SQRT_TEAM_BATTING_SLUG                     39.059317
SQRT_TEAM_BATTING_HR:IMP_TEAM_PITCHING_SO                        0.019516
SQRT_TEAM_BATTING_SLUG:IMP_TEAM_PITCHING_SO                      0.321157
SQRT_TEAM_BATTING_HR:SQRT_TEAM_BATTING_SLUG:IMP_TEAM_PITCHING_SO  -0.040964
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.25 on 2259 degrees of freedom
Multiple R-squared:  0.3848,  Adjusted R-squared:  0.3805 
F-statistic: 88.32 on 16 and 2259 DF,  p-value: < 2.2e-16
```

VIFs

```
       TEAM_BATTING_H
             6.149550
      TEAM_BATTING_2B
             8.364124
 SQRT_TEAM_BATTING_HR
          2091.353002
SQRT_TEAM_BATTING_SLUG
           109.091599
 IMP_TEAM_PITCHING_SO
           787.263817
      TEAM_FIELDING_E
             5.870373
 TEAM_BASERUN_SUCCESS
             1.359127
  IMP_TEAM_BASERUN_CS
```

```
                                              1.871733
                                    M_TEAM_BASERUN_SB
                                              3.672410
                                       TEAM_PITCHING_H
                                            113.836198
                                       TEAM_BATTING_BB
                                              8.913124
                                    TEAM_PITCHING_WHIP
                                             77.466135
          SQRT_TEAM_BATTING_HR:SQRT_TEAM_BATTING_SLUG
                                           2888.112918
            SQRT_TEAM_BATTING_HR:IMP_TEAM_PITCHING_SO
                                           3929.960073
          SQRT_TEAM_BATTING_SLUG:IMP_TEAM_PITCHING_SO
                                           1378.533196
SQRT_TEAM_BATTING_HR:SQRT_TEAM_BATTING_SLUG:IMP_TEAM_PITCHING_SO
                                           4915.618279
```

As expected, the interaction terms cause huge VIFs as well as reducing the significance of the other variables in the model. 3-way interaction variables and subsets, although significant in CHECK, cause too much complexity to the model and muddle the interpretability and are thus not recommended.

Model 4: STEPWISE2

Iterating back to the first selection method, stepwise variable selection is used with the new transformed variables.

```
Residuals:
    Min      1Q  Median      3Q     Max 
-42.941  -7.724   0.234   7.854  46.273 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)               131.091255  18.182532   7.210 7.61e-13 ***
TEAM_BATTING_H              0.077651   0.005935  13.083  < 2e-16 ***
TEAM_BATTING_2B            -0.163661   0.021704  -7.541 6.74e-14 ***
TEAM_BATTING_BB             0.071679   0.006127  11.699  < 2e-16 ***
TEAM_PITCHING_H             0.009609   0.002368   4.057 5.14e-05 ***
TEAM_FIELDING_E            -0.063336   0.003655 -17.331  < 2e-16 ***
IMP_TEAM_BATTING_SO        -0.008387   0.005201  -1.613  0.10698    
IMP_TEAM_PITCHING_SO       -0.021715   0.003738  -5.809 7.19e-09 ***
IMP_TEAM_BASERUN_SB         0.038583   0.003913   9.859  < 2e-16 ***
M_TEAM_BASERUN_CS           1.754124   0.836895   2.096  0.03619 *  
M_TEAM_FIELDING_DP          3.960386   1.473045   2.689  0.00723 ** 
M_TEAM_BATTING_SO           8.583396   1.494094   5.745 1.04e-08 ***
M_TEAM_BASERUN_SB          24.023441   1.959115  12.262  < 2e-16 ***
TEAM_BATTING_BB_SO        -27.024285   3.415518  -7.912 3.93e-15 ***
SQRT_TEAM_BATTING_HR       -3.570774   0.663154  -5.385 8.02e-08 ***
LOG_TEAM_BATTING_3B        -2.896404   1.448717  -1.999  0.04570 *  
LOG_TEAM_BATTING_SLUG      54.276401   8.164040   6.648 3.71e-11 ***
SQRT_IMP_TEAM_FIELDING_DP  -2.147572   0.279934  -7.672 2.51e-14 ***
SQRT_TEAM_PITCHING_HR       1.723302   0.565286   3.049  0.00233 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.81 on 2257 degrees of freedom
Multiple R-squared:  0.4289,	Adjusted R-squared:  0.4243 
F-statistic: 94.17 on 18 and 2257 DF,  p-value: < 2.2e-16
```

VIFs

TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_BB
12.01287	16.83045	7.306152

TEAM_PITCHING_H	TEAM_FIELDING_E	IMP_TEAM_BATTING_SO
12.76638	10.01965	23.90951
IMP_TEAM_PITCHING_SO	IMP_TEAM_BASERUN_SB	M_TEAM_BASERUN_CS
13.24274	2.27975	2.561822
M_TEAM_FIELDING_DP	M_TEAM_BATTING_SO	M_TEAM_BASERUN_SB
3.890388	1.5594	3.397509
TEAM_BATTING_BB_SO	SQRT_TEAM_BATTING_HR	LOG_TEAM_BATTING_3B
14.24817	77.52774	7.689154
LOG_TEAM_BATTING_SLUG	SQRT_IMP_TEAM_FIELDING_DP	SQRT_TEAM_PITCHING_HR
38.233	1.884965	52.16105

As shown in the following section, the performance metrics have been improved from the first model STEPWISE. However, the multicollinearity effects for the BATTING variables still need to be addressed which continue to affect the signs of the coefficients such as TEAM_BATTING_2B.

<u>Model 5: CHECK2</u>

STEPWISE2 was re-fit with an interaction term for LOG_TEAM_BATTING_SLUG and SQRT_TEAM_BATTING_HR. The interaction term was not significant, so to address multicollinearity, STEPWISE2 was adjusted by removing either HR or SLUG. Removing HR produced a better model per the metrics in the following section. Multicollinearity concerns were also mostly addressed.

```
 Residuals:
    Min      1Q  Median      3Q     Max 
-44.465  -7.861   0.102   7.971  49.074 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                62.183607   9.477662   6.561 6.60e-11 ***
TEAM_BATTING_H              0.059383   0.003855  15.403  < 2e-16 ***
TEAM_BATTING_2B            -0.090742   0.011890  -7.632 3.40e-14 ***
LOG_TEAM_BATTING_SLUG      23.466395   3.380652   6.941 5.05e-12 ***
TEAM_BATTING_BB             0.067962   0.006050  11.233  < 2e-16 ***
TEAM_PITCHING_H             0.011002   0.002262   4.865 1.22e-06 ***
TEAM_FIELDING_E            -0.059614   0.003616 -16.488  < 2e-16 ***
IMP_TEAM_BATTING_SO        -0.012769   0.005009  -2.549  0.01087 *  
IMP_TEAM_PITCHING_SO       -0.016688   0.003525  -4.735 2.33e-06 ***
IMP_TEAM_BASERUN_SB         0.040516   0.003809  10.638  < 2e-16 ***
M_TEAM_BASERUN_CS           1.721478   0.842179   2.044  0.04106 *  
M_TEAM_FIELDING_DP          3.870643   1.474431   2.625  0.00872 ** 
M_TEAM_BATTING_SO           9.110315   1.496438   6.088 1.34e-09 ***
M_TEAM_BASERUN_SB          21.998534   1.842000  11.943  < 2e-16 ***
TEAM_BATTING_BB_SO        -26.057849   3.378475  -7.713 1.83e-14 ***
LOG_TEAM_BATTING_3B         2.010233   0.926033   2.171  0.03005 *  
SQRT_IMP_TEAM_FIELDING_DP  -2.283288   0.280130  -8.151 5.92e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.89 on 2259 degrees of freedom
Multiple R-squared:  0.4208,  Adjusted R-squared:  0.4167 
F-statistic: 102.6 on 16 and 2259 DF,  p-value: < 2.2e-16
```

<u>VIFs</u>

```
           TEAM_BATTING_H            TEAM_BATTING_2B      LOG_TEAM_BATTING_SLUG 
                 5.002784                   4.985498                   6.470418 
          TEAM_BATTING_BB            TEAM_PITCHING_H            TEAM_FIELDING_E 
```

```
                7.031675                  11.488414                  9.679007
      IMP_TEAM_BATTING_SO       IMP_TEAM_PITCHING_SO       IMP_TEAM_BASERUN_SB
               21.886937                  11.618005                  2.131260
        M_TEAM_BASERUN_CS         M_TEAM_FIELDING_DP         M_TEAM_BATTING_SO
                2.560465                   3.846922                  1.543912
        M_TEAM_BASERUN_SB         TEAM_BATTING_BB_SO       LOG_TEAM_BATTING_3B
                2.964307                  13.759114                  3.100754
SQRT_IMP_TEAM_FIELDING_DP
                1.863007
```

The coefficients in CHECK2 are more intuitive, TEAM_BATTING_2B is still slightly negative due to the large coefficient in SLUG. TEAM_BATTING_BB_SO has a large negative coefficient which is not what would be expected. The 3 flag variables including BATTING_SO have large and positive coefficients which may have affected the BB_SO ratio coefficient. Additional study into the signs of the coefficients of these variables would be recommended prior to using the model in production.

4. Select Models

The Akaike Information Criterion (AIC), Mean Square Error (MSE), PRESS, and adjusted R^2 were selected as the four performance metrics to calculate for model comparison. These metrics cover a range of insight, with adjusted R^2 providing a sense of how much of the variation in TARGET_WINS in the TRAINING set can be explained from the model while punishing for complexity. The higher the adjusted R^2 the better. AIC, MSE, and PRESS are all important for prediction modeling, and the lower the values the better. The metrics for each of the 5 models are shown below.

```
> AIC(stepwise)
[1] 17720.46
> mse(stepwise)
[1] 138.3051
> PRESS(stepwise)
[1] 323663.5
```

adjR2=0.4241

```
> AIC(subset)
[1] 17927.98
> mse(subset)
[1] 152.9796
> PRESS(subset)
[1] 352062.7
```

adjR2=0.3661

```
> AIC(check)
[1] 17883.71
> mse(check)
[1] 148.9814
> PRESS(check)
[1] 346877.2
```

adjR2=0.3805

```
> AIC(stepwise2)
[1] 17718.5
> mse(stepwise2)
[1] 138.3073
```

```
> PRESS(stepwise2)
[1] 323130.8
```
adjR2 = 0.4243

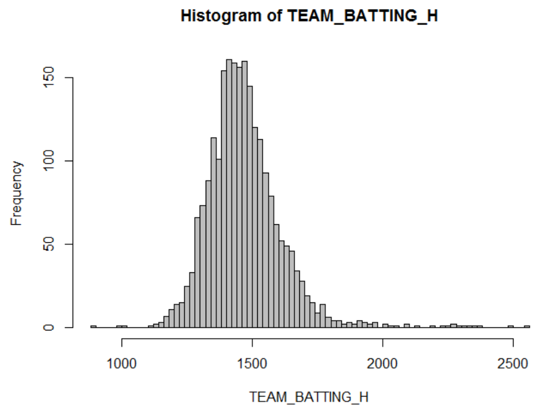
```
> AIC(check2)
[1] 17746.37
> mse(check2)
[1] 140.2576
> PRESS(check2)
[1] 326464.1
> vif(check2)
```
adjR2=0.4167

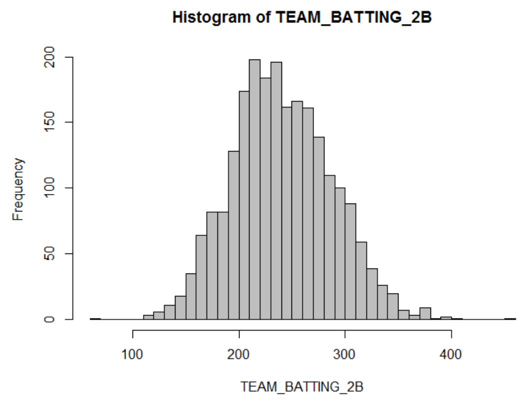
The final model, CHECK2, is selected as the champion model for a scoring program since although the performance metrics aren't as low as the two larger STEPWISE models, the large multicollinearity issues in the STEPWISE models are mostly resolved in CHECK2 while retaining relatively strong performance.

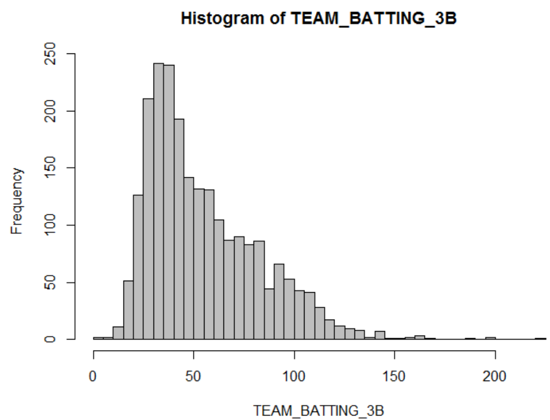
Conclusion

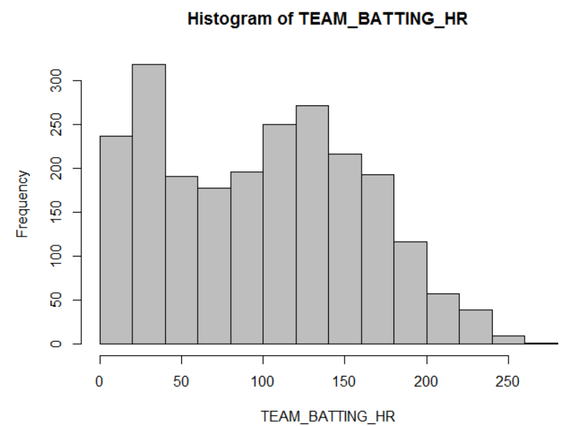
In summary, the moneyball data set was used to build a predictive scoring program for team wins. EDA was conducted, predictors were imputed, truncated, transformed, and new metrics created. Multiple variable selection methods were used and a couple custom models built to try and predict wins. A champion model was selected and used to create a scoring program which was tested by using a separate data set to predict a column of team wins.

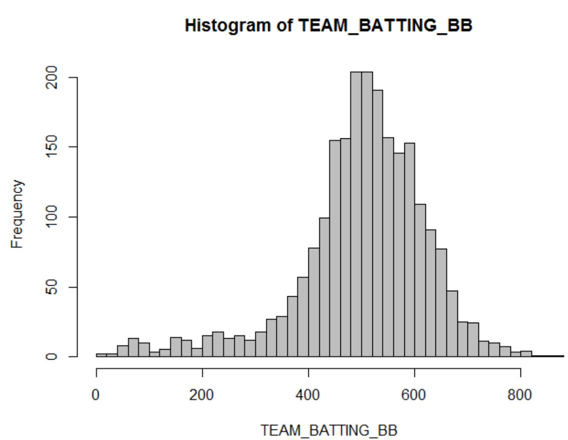
Appendix 1: Predictor Variable Histograms

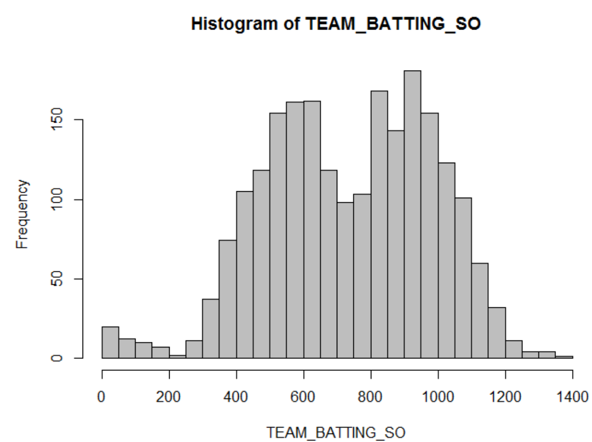


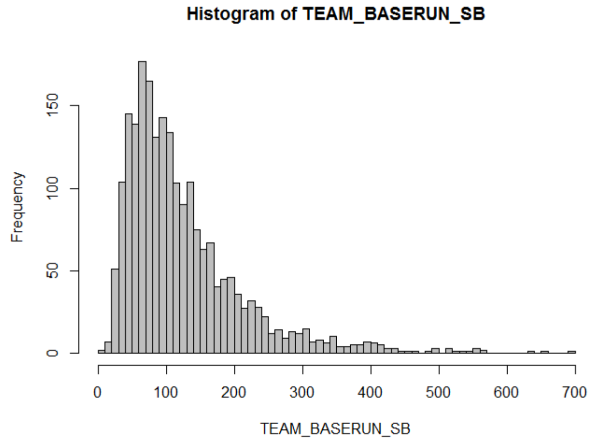


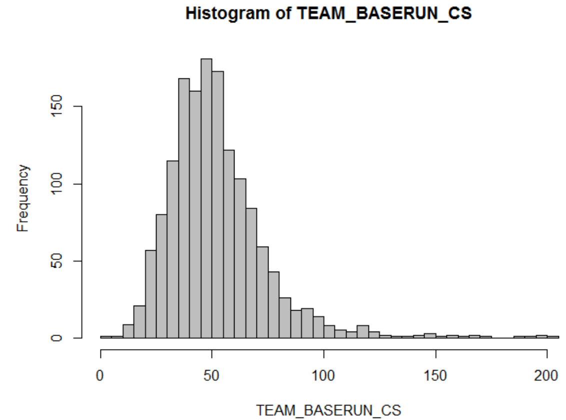


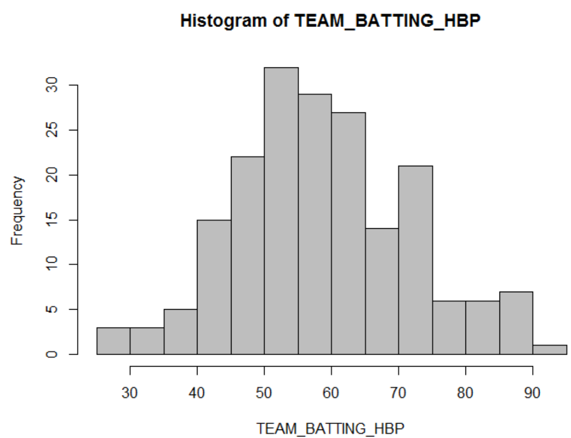


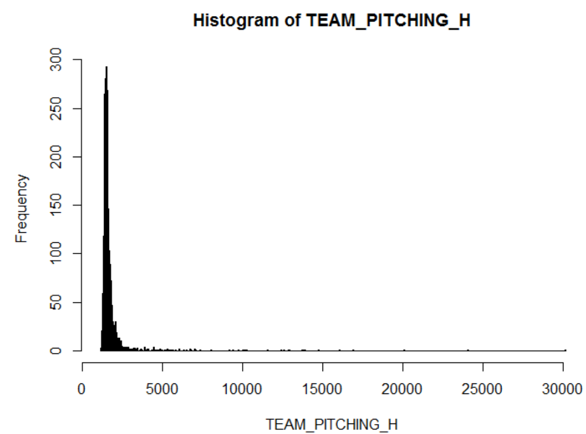


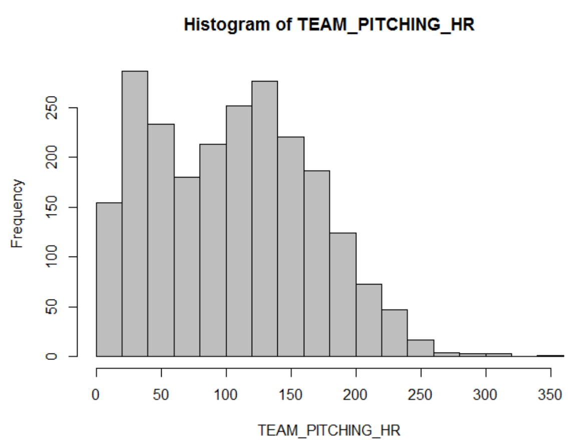


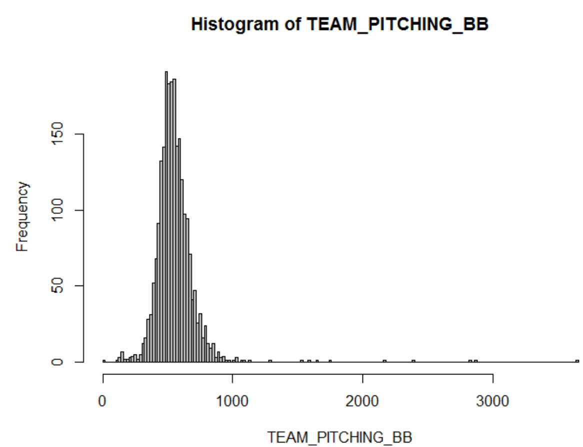

Histogram of TEAM_BASERUN_SB
Frequency
0
50
100
150
0
100
200
300
400
500
600
700
TEAM_BASERUN_SB

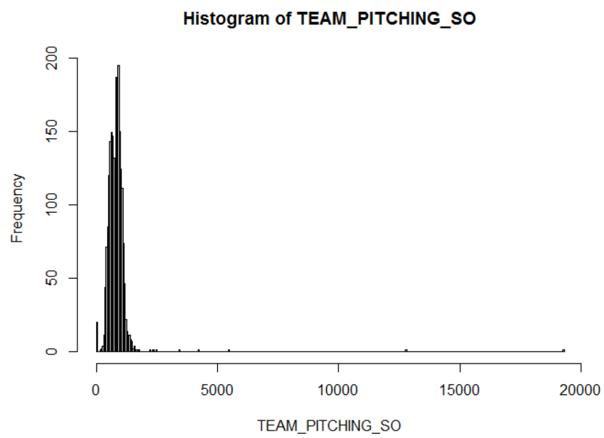

Histogram of TEAM_BASERUN_CS
Frequency
0
50
100
150
0
50
100
150
200
TEAM_BASERUN_CS

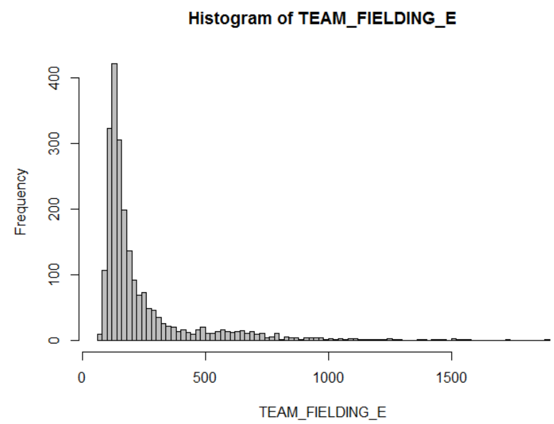

Histogram of TEAM_BATTING_HBP
Frequency
0
5
10
15
20
25
30
30
40
50
60
70
80
90
TEAM_BATTING_HBP

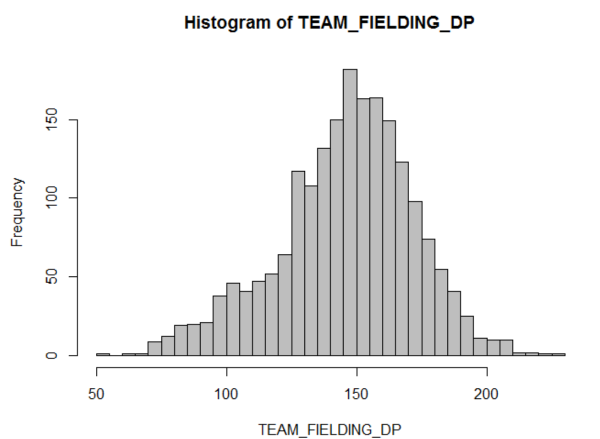

Histogram of TEAM_PITCHING_H
Frequency
0
50
100
150
200
250
300
0
5000
10000
15000
20000
25000
30000
TEAM_PITCHING_H


Histogram of TEAM_PITCHING_HR
Frequency
0
50
100
150
200
250
0
50
100
150
200
250
300
350
TEAM_PITCHING_HR


Histogram of TEAM_PITCHING_BB
Frequency
0
50
100
150
0
1000
2000
3000
TEAM_PITCHING_BB


Histogram of TEAM_PITCHING_SO
Frequency
TEAM_PITCHING_SO
0
50
100
150
200
0
5000
10000
15000
20000


Histogram of TEAM_FIELDING_E
Frequency
TEAM_FIELDING_E
0
100
200
300
400
0
500
1000
1500


Histogram of TEAM_FIELDING_DP
Frequency
TEAM_FIELDING_DP
0
50
100
150
50
100
150
200

Code:

```
#Unit 1 Assignment: Moneyball OLS Regression Project


#Import the data set

moneyball <- read.csv("C:/Users/Desktop/MSPA 411/unit 1 Weeks 1 to 3/Unit 1 - Moneyball/4 Homework/moneyball.csv", header=TRUE)

head(moneyball, 10)


#Get any packages necessary

install.packages("rJava")

install.packages("readr")

install.packages("leaps")

install.packages("xlsxjars")

install.packages("xlsx")

install.packages("party")

install.packages("Hmisc")

install.packages("mice")

install.packages("VIM")

install.packages("randomForest")

install.packages("MASS")

install.packages("MPV")

install.packages("scales")

install.packages("ggplot2")

install.packages("rlang")


library(ggplot2)

library(MPV)

library(randomForest)

library(mice)

library(VIM)
```

```
library(Hmisc)

library(party)

library(rJava)

library(readr)

library(pbkrtest)

library(car)

library(leaps)

library(MASS)

library(xlsxjars)

library(xlsx)


##########################################################################################
##########################

#1) DATA EXPLORATION

##########################################################################################
##########################

#Check the summary descriptive statistics for each variable

summary(moneyball)


#Summarize missing values

aggr_plot <- aggr(moneyball, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
labels=names(data), cex.axis=.7)


#Remove TEAM_BATTING_HBP variable due to 92% missing values

moneyball <- subset(moneyball, select = -TEAM_BATTING_HBP)


#Visualize TARGET_WINS using histogram and boxplot

par(mfrow=c(1,2))

hist(moneyball$TARGET_WINS, col = "#A71930", xlab = "TARGET_WINS", main = "Histogram of Wins")

boxplot(moneyball$TARGET_WINS, col = "#A71930", main = "Boxplot of Wins")

par(mfrow = c(1,1))
```

```
#Create histograms of each variable
with(moneyball, hist(TEAM_BATTING_H, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_BATTING_2B, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_BATTING_3B, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_BATTING_HR, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_BATTING_BB, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_BATTING_SO, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_BASERUN_SB, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_BASERUN_CS, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_PITCHING_H, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_PITCHING_HR, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_PITCHING_BB, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_PITCHING_SO, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_FIELDING_E, breaks="FD", col="gray"))
with(moneyball, hist(TEAM_FIELDING_DP, breaks="FD", col="gray"))

#Use pairwise complete correlation method to look at the predictor coefficients with TARGET_WINS
corr <- cor(moneyball, moneyball$TARGET_WINS, use = "pairwise.complete.obs")

#Create scatter plots against TARGET_WINS
panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
```

```
}

pairs(~ moneyball$TARGET_WINS + moneyball$TEAM_BATTING_H + moneyball$TEAM_BATTING_2B + moneyball$TEAM_BATTING_3B + moneyball$TEAM_BATTING_HR, lower.panel = panel.smooth, upper.panel = panel.cor)

pairs(~ moneyball$TARGET_WINS + moneyball$TEAM_BATTING_BB + moneyball$TEAM_BATTING_SO + moneyball$TEAM_PITCHING_BB + moneyball$TEAM_PITCHING_SO, lower.panel = panel.smooth, upper.panel = panel.cor)

pairs(~ moneyball$TARGET_WINS + moneyball$TEAM_BASERUN_CS + moneyball$TEAM_BASERUN_SB + moneyball$TEAM_PITCHING_H + moneyball$TEAM_PITCHING_HR + moneyball$TEAM_FIELDING_E + moneyball$TEAM_FIELDING_DP, lower.panel = panel.smooth, upper.panel = panel.cor)

############################################################################################
##########################

#2) DATA PREPARATION

############################################################################################
##########################

#Truncate TARGET_WINS to a domain of {20,120}

moneyball$TARGET_WINS[(moneyball$TARGET_WINS < 20)] = 20

moneyball$TARGET_WINS[(moneyball$TARGET_WINS > 120)] = 120

summary(moneyball$TARGET_WINS)

#MISSING VALUE IMPUTATION

#5 variables require imputation, use a CART decision tree method

#Try the mice() package for multiply imputation- use only one imputation and export the results as a data set

test <- mice(moneyball, method = "cart", minbucket = 4, m=1)   #m is the number of imputations run, default is 5

test2 <- mice(moneyball, method = "rf", ntree = 10, m=1)
```

#Compare the known observations to the imputation distributions

densityplot(test)

densityplot(test2)

#Prepare the imputed variables as dataframes

imp_df1 <- complete(test, "broad")

imp_df2 <- complete(test2, "broad")

#Join the imputed variable columns with the moneyball dataframe

moneyball2 <- cbind(moneyball, imp_df1$TEAM_BATTING_SO.1, imp_df1$TEAM_PITCHING_SO.1, imp_df2$TEAM_BASERUN_CS.1, imp_df2$TEAM_BASERUN_SB.1, imp_df2$TEAM_FIELDING_DP.1)

#Create flag variables for each of the imputed variables

moneyball2$M_TEAM_BASERUN_CS <- as.numeric(is.na(moneyball2$TEAM_BASERUN_CS))

moneyball2$M_TEAM_FIELDING_DP <- as.numeric(is.na(moneyball2$TEAM_FIELDING_DP))

moneyball2$M_TEAM_BATTING_SO <- as.numeric(is.na(moneyball2$TEAM_BATTING_SO))

moneyball2$M_TEAM_BASERUN_SB <- as.numeric(is.na(moneyball2$TEAM_BASERUN_SB))

moneyball2$M_TEAM_PITCHING_SO <- as.numeric(is.na(moneyball2$TEAM_PITCHING_SO))

old <- c('imp_df1$TEAM_BATTING_SO.1', 'imp_df1$TEAM_PITCHING_SO.1', 'imp_df2$TEAM_BASERUN_CS.1', 'imp_df2$TEAM_BASERUN_SB.1', 'imp_df2$TEAM_FIELDING_DP.1')

new <- c('IMP_TEAM_BATTING_SO', 'IMP_TEAM_PITCHING_SO', 'IMP_TEAM_BASERUN_CS', 'IMP_TEAM_BASERUN_SB', 'IMP_TEAM_FIELDING_DP')

#Rename the imputed columns with "IMP" nomenclature

setnames(moneyball2, old = old, new = new)

#Ensure no missing values remain

aggr_plot <- aggr(moneyball2, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(data), cex.axis=.7)

```
#OUTLIER TRUNCATION

#Check the lower percentiles with variables that have 0

quantile(moneyball2$TEAM_BATTING_3B, c(0.01, 0.05))

quantile(moneyball2$TEAM_BATTING_HR, c(0.01, 0.05))

quantile(moneyball2$TEAM_BATTING_BB, c(0.01, 0.05))

quantile(moneyball2$IMP_TEAM_BATTING_SO, c(0.01, 0.05))

quantile(moneyball2$IMP_TEAM_BASERUN_SB, c(0.01, 0.05))

quantile(moneyball2$IMP_TEAM_BASERUN_CS, c(0.01, 0.05))

quantile(moneyball2$TEAM_PITCHING_HR, c(0.01, 0.05))

quantile(moneyball2$TEAM_PITCHING_BB, c(0.01, 0.05))

quantile(moneyball2$IMP_TEAM_PITCHING_SO, c(0.01, 0.05))


#Select a percentile for each and truncate

moneyball2$TEAM_BATTING_3B[(moneyball2$TEAM_BATTING_3B <
quantile(moneyball2$TEAM_BATTING_3B, 0.05))] = quantile(moneyball2$TEAM_BATTING_3B, 0.05)

moneyball2$TEAM_BATTING_HR[(moneyball2$TEAM_BATTING_HR <
quantile(moneyball2$TEAM_BATTING_HR, 0.05))] = quantile(moneyball2$TEAM_BATTING_HR, 0.05)

moneyball2$TEAM_BATTING_BB[(moneyball2$TEAM_BATTING_BB <
quantile(moneyball2$TEAM_BATTING_BB, 0.05))] = quantile(moneyball2$TEAM_BATTING_BB, 0.05)

moneyball2$IMP_TEAM_BATTING_SO[(moneyball2$IMP_TEAM_BATTING_SO <
quantile(moneyball2$IMP_TEAM_BATTING_SO, 0.05))] =
quantile(moneyball2$IMP_TEAM_BATTING_SO, 0.05)

moneyball2$IMP_TEAM_BASERUN_SB[(moneyball2$IMP_TEAM_BASERUN_SB <
quantile(moneyball2$IMP_TEAM_BASERUN_SB, 0.05))] =
quantile(moneyball2$IMP_TEAM_BASERUN_SB, 0.05)

moneyball2$IMP_TEAM_BASERUN_CS[(moneyball2$IMP_TEAM_BASERUN_CS <
quantile(moneyball2$IMP_TEAM_BASERUN_CS, 0.05))] =
quantile(moneyball2$IMP_TEAM_BASERUN_CS, 0.05)

moneyball2$TEAM_PITCHING_HR[(moneyball2$TEAM_PITCHING_HR <
quantile(moneyball2$TEAM_PITCHING_HR, 0.05))] = quantile(moneyball2$TEAM_PITCHING_HR, 0.05)

moneyball2$TEAM_PITCHING_BB[(moneyball2$TEAM_PITCHING_BB <
quantile(moneyball2$TEAM_PITCHING_BB, 0.01))] = quantile(moneyball2$TEAM_PITCHING_BB, 0.01)

moneyball2$IMP_TEAM_PITCHING_SO[(moneyball2$IMP_TEAM_PITCHING_SO <
quantile(moneyball2$IMP_TEAM_PITCHING_SO, 0.01))] =
quantile(moneyball2$IMP_TEAM_PITCHING_SO, 0.01)
```

```
#Spot check that the 0's were removed
summary(moneyball2)
#Only 0's remaining are for un-imputed variables or flag variables

#Next, truncate variables with obvious outliers on the high end by first looking at upper percentiles
quantile(moneyball2$IMP_TEAM_PITCHING_SO, c(0.95, 0.99))
quantile(moneyball2$TEAM_PITCHING_BB, c(0.95, 0.99))
quantile(moneyball2$TEAM_FIELDING_E, c(0.95, 0.99))
quantile(moneyball2$TEAM_PITCHING_H, c(0.95, 0.99))
#Decide an appropriate truncation based on charts and percentiles
moneyball2$IMP_TEAM_PITCHING_SO[(moneyball2$IMP_TEAM_PITCHING_SO >
quantile(moneyball2$IMP_TEAM_PITCHING_SO, 0.99))] =
quantile(moneyball2$IMP_TEAM_PITCHING_SO, 0.99)
moneyball2$TEAM_PITCHING_BB[(moneyball2$TEAM_PITCHING_BB >
quantile(moneyball2$TEAM_PITCHING_BB, 0.99))] = quantile(moneyball2$TEAM_PITCHING_BB, 0.99)
moneyball2$TEAM_FIELDING_E[(moneyball2$TEAM_FIELDING_E >
quantile(moneyball2$TEAM_FIELDING_E, 0.99))] = quantile(moneyball2$TEAM_FIELDING_E, 0.99)
moneyball2$TEAM_PITCHING_H[(moneyball2$TEAM_PITCHING_H > 3000)] = 3000

#Recheck distributions
with(moneyball2, hist(IMP_TEAM_PITCHING_SO, breaks="FD", col="gray"))

#NEW VARIABLES, consider calculated measures that could improve predictive accuracy
#Walk to strikeout ratios
moneyball2$TEAM_BATTING_BB_SO <- moneyball2$TEAM_BATTING_BB /
moneyball2$IMP_TEAM_BATTING_SO
moneyball2$TEAM_PITCHING_BB_SO <- moneyball2$TEAM_PITCHING_BB /
moneyball2$IMP_TEAM_PITCHING_SO

#Create modified slugging percentage based on available data
```

moneyball2$TEAM_BATTING_SLUG <- (moneyball2$TEAM_BATTING_2B + moneyball2$TEAM_BATTING_3B + moneyball2$TEAM_BATTING_HR) / moneyball2$TEAM_BATTING_H

#Create stealing success rate

moneyball2$TEAM_BASERUN_SUCCESS <- moneyball2$IMP_TEAM_BASERUN_SB / (moneyball2$IMP_TEAM_BASERUN_SB + moneyball2$IMP_TEAM_BASERUN_CS)

#Create WHIP-like metric using 1458 as denominator (assumes 9 innings over 162 games, ignores extra innings)

moneyball2$TEAM_PITCHING_WHIP <- (moneyball2$TEAM_PITCHING_BB + moneyball2$TEAM_PITCHING_H) / 1458

#TRANSFORMATIONS

#Transform some predictors after initial modeling attempts

moneyball2$SQRT_TEAM_BATTING_SLUG <- sqrt(moneyball2$TEAM_BATTING_SLUG)

moneyball2$LOG_TEAM_BATTING_SLUG <- log(moneyball2$TEAM_BATTING_SLUG)

moneyball2$SQRT_TEAM_BATTING_HR <- sqrt(moneyball2$TEAM_BATTING_HR)

moneyball2$LOG_TEAM_BATTING_3B <- log(moneyball2$TEAM_BATTING_3B)

moneyball2$SQRT_TEAM_PITCHING_HR <- sqrt(moneyball2$TEAM_PITCHING_HR)

moneyball2$SQRT_IMP_TEAM_FIELDING_DP <- sqrt(moneyball2$IMP_TEAM_FIELDING_DP)

#Rerun some correlations after imputing and transformations

pairs(~ moneyball2$TARGET_WINS + moneyball2$TEAM_BATTING_H + moneyball2$TEAM_BATTING_2B + moneyball2$TEAM_BATTING_3B + moneyball2$TEAM_BATTING_HR + moneyball2$TEAM_BATTING_SLUG, lower.panel = panel.smooth, upper.panel = panel.cor)

pairs(~ moneyball2$TARGET_WINS + moneyball2$TEAM_PITCHING_H + moneyball2$TEAM_PITCHING_HR + moneyball2$IMP_TEAM_PITCHING_SO + moneyball2$TEAM_BATTING_BB_SO + moneyball2$LOG_TEAM_BATTING_SLUG, lower.panel = panel.smooth, upper.panel = panel.cor)

```
pairs(~ moneyball2$TARGET_WINS + moneyball2$IMP_TEAM_BASERUN_CS +
moneyball2$IMP_TEAM_BASERUN_SB + moneyball2$TEAM_BASERUN_SUCCESS +
moneyball2$INV_IMP_TEAM_FIELDING_DP, lower.panel = panel.smooth, upper.panel = panel.cor)



##########################################################################################
##########################

#3) BUILD MODELS

##########################################################################################
##########################

#Create a full model of 25 predictors

mb_full <- subset(moneyball2, select = c(-TEAM_BASERUN_CS, -TEAM_FIELDING_DP, -
TEAM_BATTING_SO, -TEAM_PITCHING_SO, -INDEX, -TEAM_BASERUN_SB))


#MODEL 1: STEPWISE REGRESSION

stepwise_model <- lm(TARGET_WINS ~ ., data = mb_full)

stepwise <- stepAIC(stepwise_model, direction = "both")

summary(stepwise)


#Look at residual plot

plot(stepwise$fitted.values, stepwise$residuals)

abline(0, 0)

#Generate partial residual plot of BATTING_SLUG

plot(moneyball2$TEAM_BATTING_SLUG, stepwise$residuals)

abline(0, 0)



#MODEL 2: ALL SUBSETS REGRESSION

subsets <- regsubsets(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
LOG_TEAM_BATTING_3B + SQRT_TEAM_BATTING_SLUG + SQRT_TEAM_BATTING_HR +
```

```
                    TEAM_BATTING_BB_SO + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO +
TEAM_BASERUN_SUCCESS + IMP_TEAM_BASERUN_CS + SQRT_TEAM_PITCHING_HR +

                    TEAM_FIELDING_E + SQRT_IMP_TEAM_FIELDING_DP + TEAM_PITCHING_H +
M_TEAM_BASERUN_CS +

                    M_TEAM_BASERUN_SB + M_TEAM_FIELDING_DP + M_TEAM_BATTING_SO,  data =
moneyball2, nbest = 2)

plot(subsets, scale="adjr2")

#Use the results of the subset plot to fit the recommended model

subset <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_BB +

                    TEAM_BASERUN_SUCCESS + IMP_TEAM_BASERUN_CS + TEAM_FIELDING_E +
SQRT_IMP_TEAM_FIELDING_DP +

                    M_TEAM_BASERUN_SB, data = moneyball2)

summary(subset)



#MODEL 3: FIRST CUSTOM MODEL

check <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + SQRT_TEAM_BATTING_HR *
SQRT_TEAM_BATTING_SLUG * IMP_TEAM_PITCHING_SO +

                    TEAM_FIELDING_E + TEAM_BASERUN_SUCCESS + IMP_TEAM_BASERUN_CS +
M_TEAM_BASERUN_SB + TEAM_PITCHING_H +

                    TEAM_BATTING_BB + TEAM_PITCHING_WHIP, data = moneyball2)

summary(check)



#MODEL 4: STEPWISE2

mb_full2 <- subset(moneyball2, select = c(-TEAM_BASERUN_CS, -TEAM_FIELDING_DP, -
TEAM_BATTING_SO, -TEAM_PITCHING_SO, -INDEX, -TEAM_BASERUN_SB, -TEAM_BATTING_SLUG, -
TEAM_BATTING_3B, -TEAM_BATTING_HR, -TEAM_PITCHING_HR, -IMP_TEAM_FIELDING_DP))

stepwise_model2 <- lm(TARGET_WINS ~ ., data = mb_full2)

stepwise2 <- stepAIC(stepwise_model2, direction = "both")

summary(stepwise2)
```

```
#MODEL 5: SECOND CUSTOM MODEL

check2 <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + LOG_TEAM_BATTING_SLUG
+ TEAM_BATTING_BB +

          TEAM_PITCHING_H + TEAM_FIELDING_E + IMP_TEAM_BATTING_SO +
IMP_TEAM_PITCHING_SO + IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_CS +

          M_TEAM_FIELDING_DP + M_TEAM_BATTING_SO + M_TEAM_BASERUN_SB +
TEAM_BATTING_BB_SO + LOG_TEAM_BATTING_3B +

          SQRT_IMP_TEAM_FIELDING_DP, data = moneyball2)

summary(check2)




###########################################################################################
##########################

#4) SELECT MODELS

###########################################################################################
##########################

#Function for Mean Square Error Calculation

mse <- function(sm)

  mean(sm$residuals^2)


#Check some performance metrics

AIC(stepwise)

mse(stepwise)

PRESS(stepwise)

vif(stepwise)


AIC(subset)

mse(subset)

PRESS(subset)

vif(subset)
```

```
AIC(check)
mse(check)
PRESS(check)
vif(check)

AIC(stepwise2)
mse(stepwise2)
PRESS(stepwise2)
vif(stepwise2)

AIC(check2)
mse(check2)
PRESS(check2)
vif(check2)



#################################################################################################
#############################
#5) SCORING PROGRAM
#################################################################################################
#############################
#Create a scoring program
#First pull in the TEST file
moneyball_test <- read.csv("C:/Users/Desktop/MSPA 411/unit 1 Weeks 1 to 3/Unit 1 - Moneyball/4 Homework/moneyball_test.csv", header=TRUE)
head(moneyball_test, 10)
summary(moneyball_test)
```

```
#Remove TEAM_BATTING_HBP variable due to missing values
moneyball_test <- subset(moneyball_test, select = -TEAM_BATTING_HBP)



#IMPUTATION
test_bf <- mice(moneyball_test, method = "rf", ntree = 10, m=1)
test_bp <- mice(moneyball_test, method = "cart", minbucket = 4, m=1)
densityplot(test_bf)
densityplot(test_bp)

#Prepare the imputed variables as dataframes
test_bf2 <- complete(test_bf, "broad")

#Join the imputed variable columns with the test dataframe index
moneyball_test2 <- cbind(moneyball_test$INDEX, test_bf2)
moneyball_test3 <- subset(moneyball_test, select = INDEX)

#Create flag variables for each of the potentially imputed variables
moneyball_test3$M_TEAM_BASERUN_CS <- as.numeric(is.na(moneyball_test$TEAM_BASERUN_CS))
moneyball_test3$M_TEAM_BASERUN_SB <- as.numeric(is.na(moneyball_test$TEAM_BASERUN_SB))
moneyball_test3$M_TEAM_BATTING_2B <- as.numeric(is.na(moneyball_test$TEAM_BATTING_2B))
moneyball_test3$M_TEAM_BATTING_3B <- as.numeric(is.na(moneyball_test$TEAM_BATTING_3B))
moneyball_test3$M_TEAM_BATTING_BB <- as.numeric(is.na(moneyball_test$TEAM_BATTING_BB))
moneyball_test3$M_TEAM_BATTING_H <- as.numeric(is.na(moneyball_test$TEAM_BATTING_H))
moneyball_test3$M_TEAM_BATTING_HR <- as.numeric(is.na(moneyball_test$TEAM_BATTING_HR))
moneyball_test3$M_TEAM_BATTING_SO <- as.numeric(is.na(moneyball_test$TEAM_BATTING_SO))
moneyball_test3$M_TEAM_FIELDING_DP <- as.numeric(is.na(moneyball_test$TEAM_FIELDING_DP))
moneyball_test3$M_TEAM_FIELDING_E <- as.numeric(is.na(moneyball_test$TEAM_FIELDING_E))
moneyball_test3$M_TEAM_PITCHING_H <- as.numeric(is.na(moneyball_test$TEAM_PITCHING_H))
moneyball_test3$M_TEAM_PITCHING_SO <- as.numeric(is.na(moneyball_test$TEAM_PITCHING_SO))
```

```
#Find and create a vector of flag variables that had missing values
flag = apply(moneyball_test3, MARGIN=2, function(col){any(col==1)})
flag2 = as.vector(which(flag==TRUE))

#Subset the flag variable dataframe to only variables that had missing values
moneyball_test3 <- moneyball_test3[flag2]

#Join the flag variables with the imputed test variables
moneyball_test2 <- cbind(moneyball_test2, moneyball_test3)


#TRUNCATION
moneyball_test2$TEAM_BATTING_3B.1[(moneyball_test2$TEAM_BATTING_3B.1 <
quantile(moneyball_test2$TEAM_BATTING_3B.1, 0.05))] =
quantile(moneyball_test2$TEAM_BATTING_3B.1, 0.05)

moneyball_test2$TEAM_BATTING_HR.1[(moneyball_test2$TEAM_BATTING_HR.1 <
quantile(moneyball_test2$TEAM_BATTING_HR.1, 0.05))] =
quantile(moneyball_test2$TEAM_BATTING_HR.1, 0.05)

moneyball_test2$TEAM_BATTING_BB.1[(moneyball_test2$TEAM_BATTING_BB.1 <
quantile(moneyball_test2$TEAM_BATTING_BB.1, 0.05))] =
quantile(moneyball_test2$TEAM_BATTING_BB.1, 0.05)

moneyball_test2$TEAM_BATTING_SO.1[(moneyball_test2$TEAM_BATTING_SO.1 <
quantile(moneyball_test2$TEAM_BATTING_SO.1, 0.05))] =
quantile(moneyball_test2$TEAM_BATTING_SO.1, 0.05)

moneyball_test2$TEAM_BASERUN_SB.1[(moneyball_test2$TEAM_BASERUN_SB.1 <
quantile(moneyball_test2$TEAM_BASERUN_SB.1, 0.05))] =
quantile(moneyball_test2$TEAM_BASERUN_SB.1, 0.05)

moneyball_test2$TEAM_BASERUN_CS.1[(moneyball_test2$TEAM_BASERUN_CS.1 <
quantile(moneyball_test2$TEAM_BASERUN_CS.1, 0.05))] =
quantile(moneyball_test2$TEAM_BASERUN_CS.1, 0.05)

moneyball_test2$TEAM_PITCHING_SO.1[(moneyball_test2$TEAM_PITCHING_SO.1 <
quantile(moneyball_test2$TEAM_PITCHING_SO.1, 0.01))] =
quantile(moneyball_test2$TEAM_PITCHING_SO.1, 0.01)
```

```
moneyball_test2$TEAM_PITCHING_SO.1[(moneyball_test2$TEAM_PITCHING_SO.1 >
quantile(moneyball_test2$TEAM_PITCHING_SO.1, 0.99))] =
quantile(moneyball_test2$TEAM_PITCHING_SO.1, 0.99)

moneyball_test2$TEAM_FIELDING_E.1[(moneyball_test2$TEAM_FIELDING_E.1 >
quantile(moneyball_test2$TEAM_FIELDING_E.1, 0.99))] =
quantile(moneyball_test2$TEAM_FIELDING_E.1, 0.99)

moneyball_test2$TEAM_PITCHING_H.1[(moneyball_test2$TEAM_PITCHING_H.1 > 3000)] = 3000

summary(moneyball_test2)



#NEW VARIABLES

moneyball_test2$TEAM_BATTING_BB_SO <- moneyball_test2$TEAM_BATTING_BB.1 /
moneyball_test2$TEAM_BATTING_SO.1

moneyball_test2$TEAM_BATTING_SLUG <- (moneyball_test2$TEAM_BATTING_2B.1 +
moneyball_test2$TEAM_BATTING_3B.1 + moneyball_test2$TEAM_BATTING_HR.1) /
moneyball_test2$TEAM_BATTING_H.1



#TRANSFORMATION

moneyball_test2$TEAM_BATTING_3B.1 <- log(moneyball_test2$TEAM_BATTING_3B.1)

moneyball_test2$TEAM_BATTING_SLUG <- log(moneyball_test2$TEAM_BATTING_SLUG)

moneyball_test2$TEAM_FIELDING_DP.1 <- sqrt(moneyball_test2$TEAM_FIELDING_DP.1)



#STAND ALONE SCORING OF CHECK2

moneyball_test2$P_TARGET_WINS <- 62.183607 + 0.059383 * moneyball_test2$TEAM_BATTING_H.1 -
  0.090742* moneyball_test2$TEAM_BATTING_2B.1 +
  23.466395* moneyball_test2$TEAM_BATTING_SLUG +
  0.067962* moneyball_test2$TEAM_BATTING_BB.1 +
  0.011002* moneyball_test2$TEAM_PITCHING_H.1 -
  0.059614* moneyball_test2$TEAM_FIELDING_E.1 -
  0.012769* moneyball_test2$TEAM_BATTING_SO.1 -
  0.016688* moneyball_test2$TEAM_PITCHING_SO.1 +
```

```
  0.040516* moneyball_test2$TEAM_BASERUN_SB.1 +
  1.721478* moneyball_test2$M_TEAM_BASERUN_CS +
  3.870643* moneyball_test2$M_TEAM_FIELDING_DP +
  9.110315* moneyball_test2$M_TEAM_BATTING_SO +
  21.998534* moneyball_test2$M_TEAM_BASERUN_SB -
  26.057849* moneyball_test2$TEAM_BATTING_BB_SO +
  2.010233* moneyball_test2$TEAM_BATTING_3B.1 -
  2.283288* moneyball_test2$TEAM_FIELDING_DP.1


#Check the distribution of target wins
moneyball_test2$P_TARGET_WINS[(moneyball_test2$P_TARGET_WINS < 20)] = 20
moneyball_test2$P_TARGET_WINS[(moneyball_test2$P_TARGET_WINS > 120)] = 120
summary(moneyball_test2$P_TARGET_WINS)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 20.00   74.46   80.71   79.96   86.61  111.30


#SCORING FILE PREPARATION
#subset of data set for the deliverable "Scored data file"
setnames(moneyball_test2, old = "moneyball_test$INDEX", new = "INDEX")
prediction <- moneyball_test2[c("INDEX","P_TARGET_WINS")]


#PREDICTION FILE
setwd("C:/Users/Desktop/MSPA 411/unit 1 Weeks 1 to 3/Unit 1 - Moneyball/4 Homework/")
write.xlsx(prediction, file = "write.xlsx", sheetName = "Predictions",
        col.names = TRUE)
```