Assignment #1

Introduction:
The objective of this analysis is to develop a model that is capable of predicting Wins for a Baseball team given performance statistics in a variety of offensive and defensive categories. In order to build this model several steps have been taken. The first step involved an exploratory data analysis to understand what the data "looks" like. Steps were then taken to prepare the data for modeling. These steps including imputing missing data and transforming data to improve normality of the distributions and to adjust for outlier data. A predictive modeling framework was then used along with automated variable selection techniques for two of the models created. The third model was created using Random Forest Regression. The models created were compared using cross-validation comparison, which leveraged both statistical validation (performance) and business usability metrics (simplicity). The results and key findings of the modelling analysis is discussed through the body of this. The modeling and analytics was completed with R. The code can be found in the appendix.

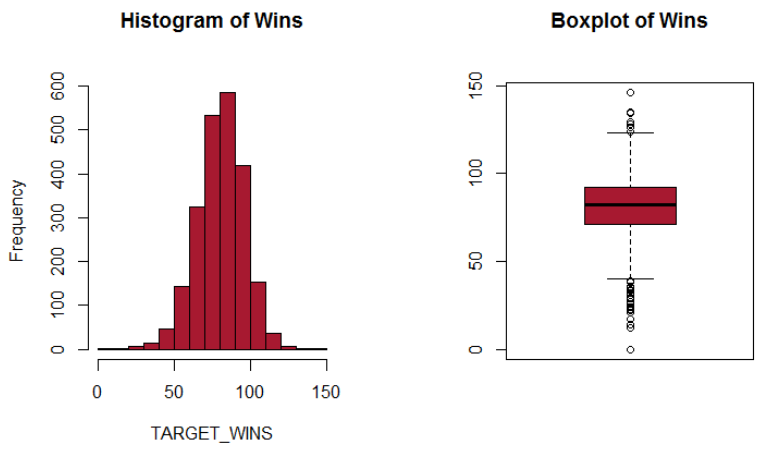
RESULTS:
Data Exploration:
The data used for this analysis consists of 16 original variables from the moneyball data set, each with 2,276 observations pertaining to baseball team performance between 1900 – 1950. These data are a record of a team's wins as well as its performance in four separate areas including batting (offense), base running (offense), pitching (defense) and fielding (defense). Prior to leveraging the data to build a predictive model a review of the data was completed. Note that a 17th variable was added to the data for this analysis. The variable is single base hits and was derived by subtracting doubles, triples and home runs from the hits variable.

Team Wins Overview: Team wins indicate the number of baseball games won in a single season. The Mean of 82 is only slightly larger than the Median of 80.79, which would indicate a slight skewness to the left. This is confirmed given the skewness result of -0.40 and can be seen in the Histogram and Boxplot in Figure 1. Given this a transformation is likely not required as the distribution is relatively normal. Finally, the data does not contain any missing values.

Table 1: Team Wins – Summary

Variable Name	Variable Type	Mean	Median	STD	Skewness	NAs
Team Wins	Response	82.00	80.79	15.75	-0.40	0

Figure 1: Team Wins – Histogram and Box Plot



Team Batting Overview: Team Batting data are offensive statistics for a baseball team. These variables can be used as predictor variables for forecasting team wins. With the exception of “strike outs” each of the offensive variables should have a positive influence on wins (e.g. more home runs should mean more wins). Table 2 shows some of the key statistics for each of these variables. The variables “Hits”, “Singles”, “Triples”, and “Walks” all have skewness results greater than abs(1), which indicates highly skewed distributions. The Mean and Median values for these variables however, do not appear to be significantly different, which could indicate that there are outliers in the data. In looking at the histogram and boxplots in Figure 2 we can see that this appears to be the case for “Hits” and “Singles”, which have long tails down the right side of the histograms. “Triples” and “Walks” simply appear to have a higher frequency of data points above or below the mean. All four variables may require some form of transformation to normalize the distributions. The variables “Strike Out” and “Hit by Pitch” have missing data, which will need to be addressed. Given “strike outs” appears to have a relatively low number of missing data a form of imputation could be used to fill in the missing values. The “hit by pitch” variable however, has a significant number of missing data, which means the variable may need to be excluded from the analysis. Figure 3 is a scatter plot matrix that displays the bivariate scatter plots for each of the batting variables on the lower panel and the correlations between each of the variables on the upper panel. The variables “Hits”, “doubles” and “walks” all have a positive relationship with wins, which indicates they could be good predictors for team wins. The triples variable appears to influence wins only after a certain threshold. The home runs and strike outs variables both have a positive relationship with wins until a certain threshold and then it becomes negative. This is interesting, but makes sense as there have been many power hitting teams that don’t win consistently and a team with too many strikeouts could mean a poor hitting team, but could also indicate more batting opportunities assuming a high team batting average. Figure 3 also shows some high correlations between some of the predictor variables in the upper panel, which could mean multicollinearity is present. A VIF test will need to be completed following the model building phase to confirm.

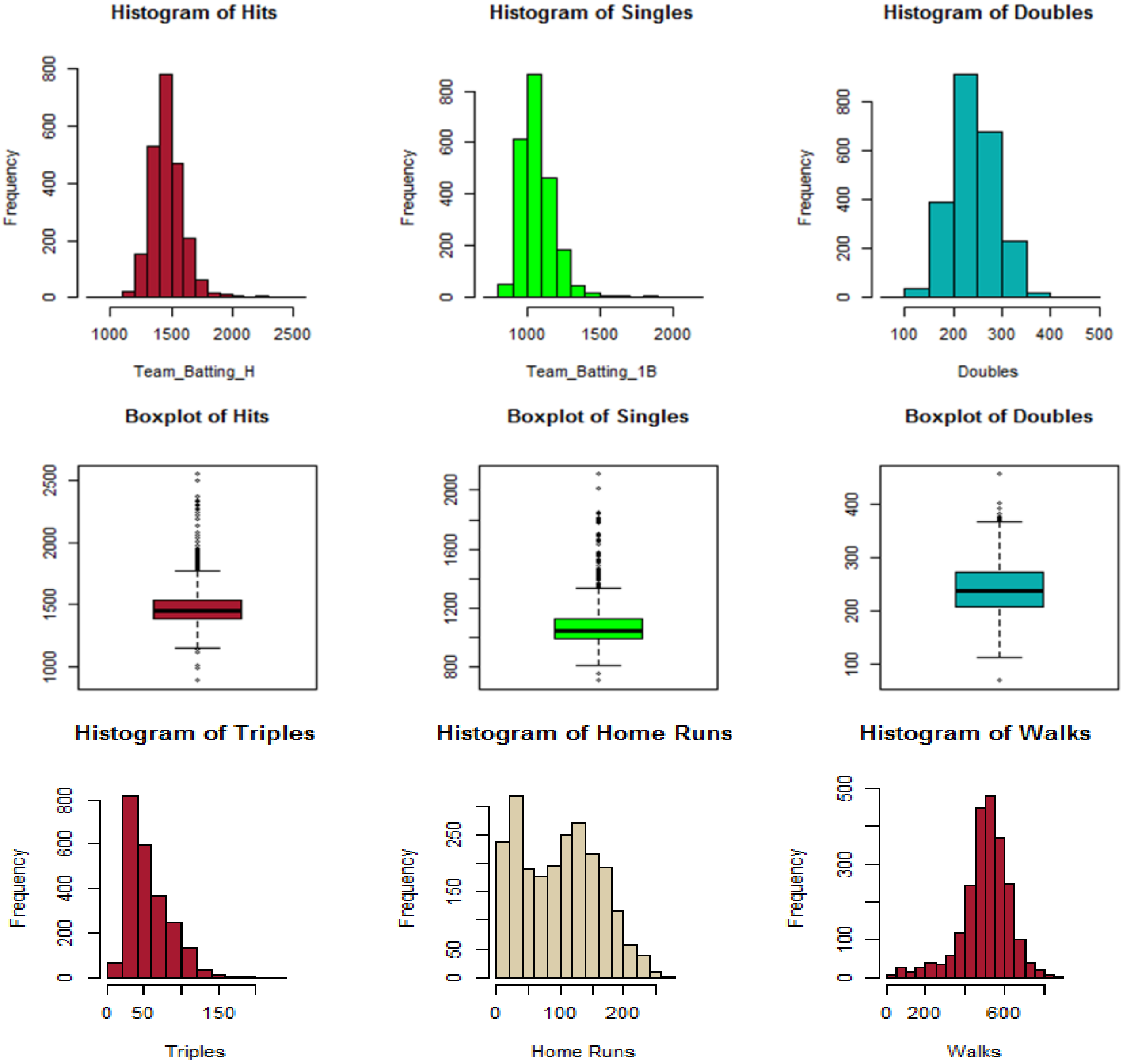
Table 2: Team Batting – Summary

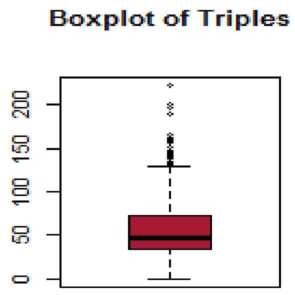
Variable Name	Variable Type	Mean	Median	STD	Skewness	NAs
Hits	Predictor	1469.00	1454	144.59	1.57	0
Singles	Predictor	1073.20	1050	128.92	2.05	0
Doubles	Predictor	241.20	238	46.8	0.22	0
Triples	Predictor	55.25	47	27.94	1.11	0

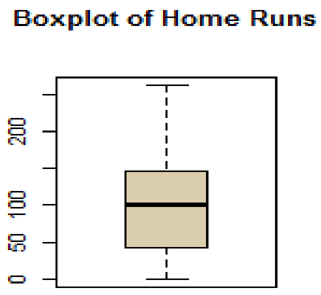
Home Runs	Predictor	99.61	102	60.55	0.19	0
Strike Outs	Predictor	735.6	750	104.16*	0.4*	102
Walks	Predictor	501.6	512	122.67	-1.03	0
Hit By Pitch	Predictor	59.36	58	12.97*	0.32*	2085

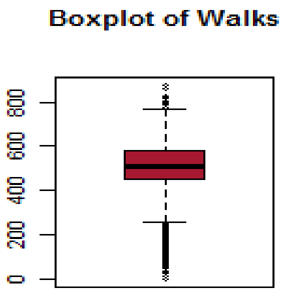
* excludes NAs

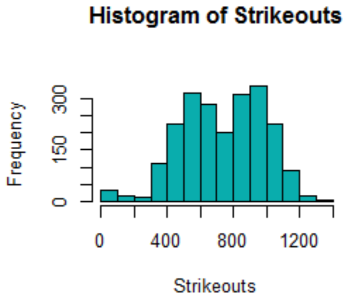
Figure 2: Team Batting – Histogram and Box Plot

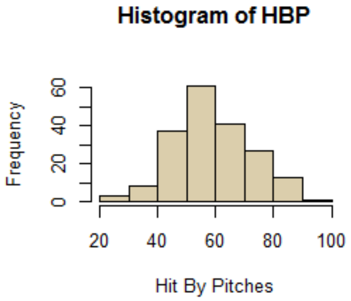


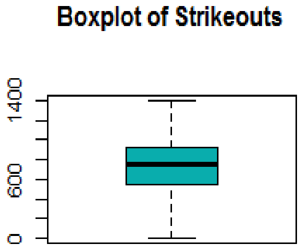












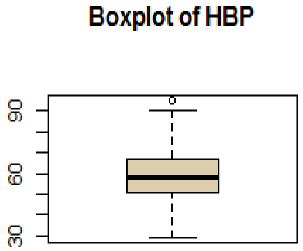
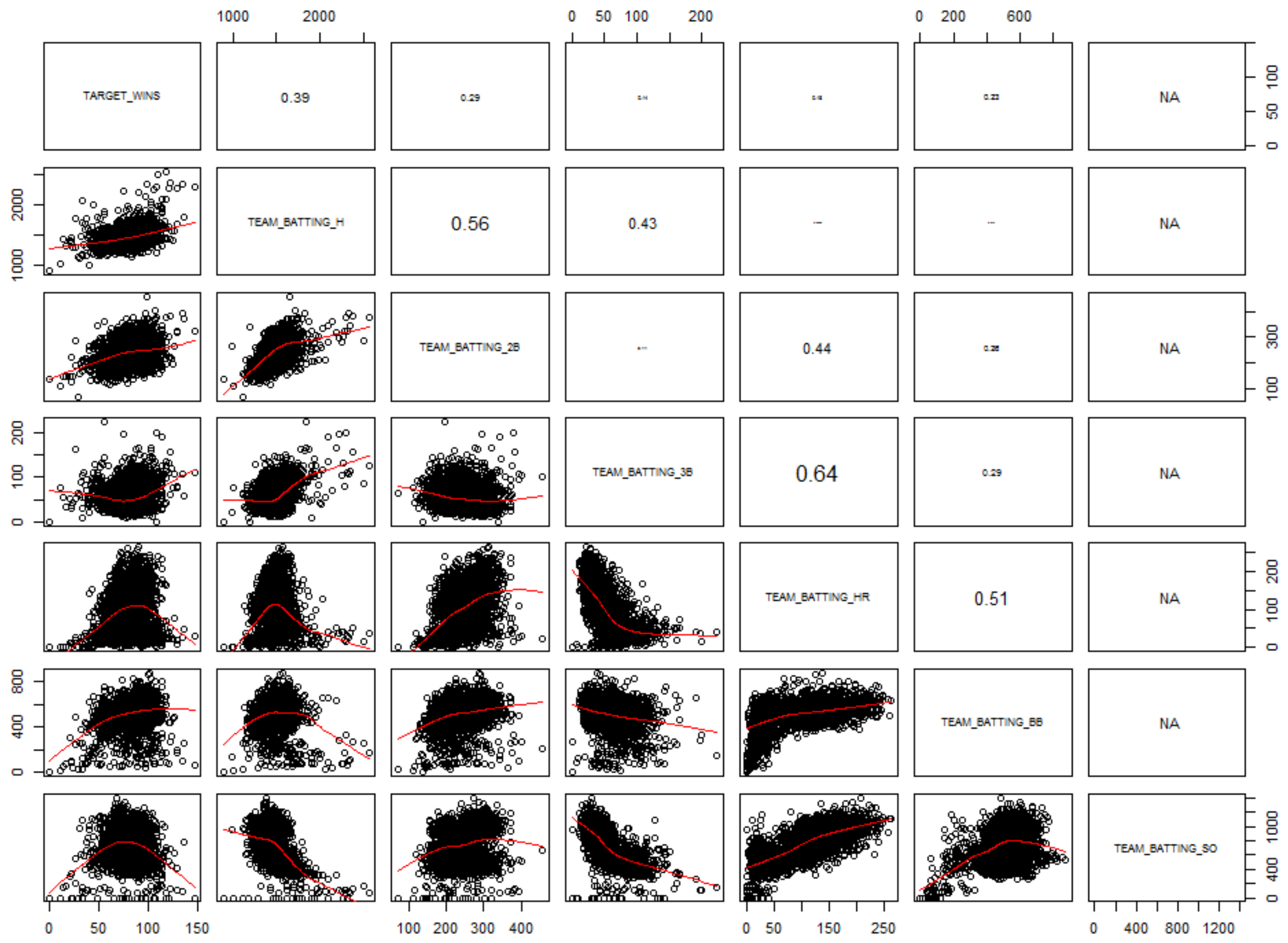


Figure 3: Team Batting – Scatter Plot Matrix w/ Correlation and Smoothed Line



Team Base Running Overview: Team Base Running data are offensive statistics for a baseball team and are potential predictor variables. Both variables have several missing data points as can be seen in Table 3. The "Caught Stealing" variable may have to be removed given the number of missing data is quite large while the missing values in "Stolen Bases" should be able to be imputed. The distributions for each of the variables can be seen in Figure 4. Both distributions appear to be skewed to the right due to outlier or highly influential data points. This may need to be addressed in order to have these variables as viable predictors of team wins. Figure 5 shows that there is a slight positive correlation between team wins and stolen bases, which makes sense given it advances the runner into a scoring position. The caught stealing variable appears to have a positive correlation for lower values, but flattens out and looks to turn into a slight negative correlation. This also makes sense as stealing bases can help win games, but even the best base stealers get caught sometimes. Getting caught too often however can cost a team runs and potentially games.

Table 3: Team Base Running – Summary

Variable Name	Variable Type	Mean	Median	STD	Skewness	NAs
Stolen Bases	Predictor	124.80	101	29.92*	0.56*	131
Caught Stealing	Predictor	52.80	49	11.90*	0.35*	772

* excludes NAs

Figure 4: Team Base Running – Histogram and Box Plot

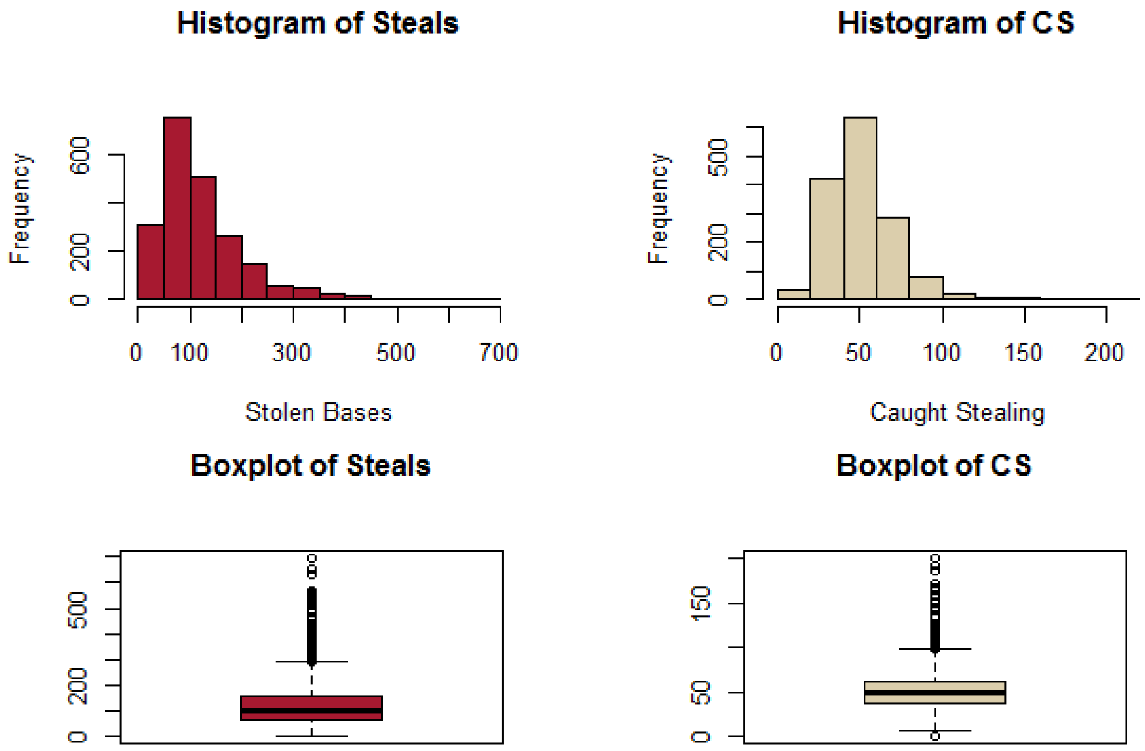
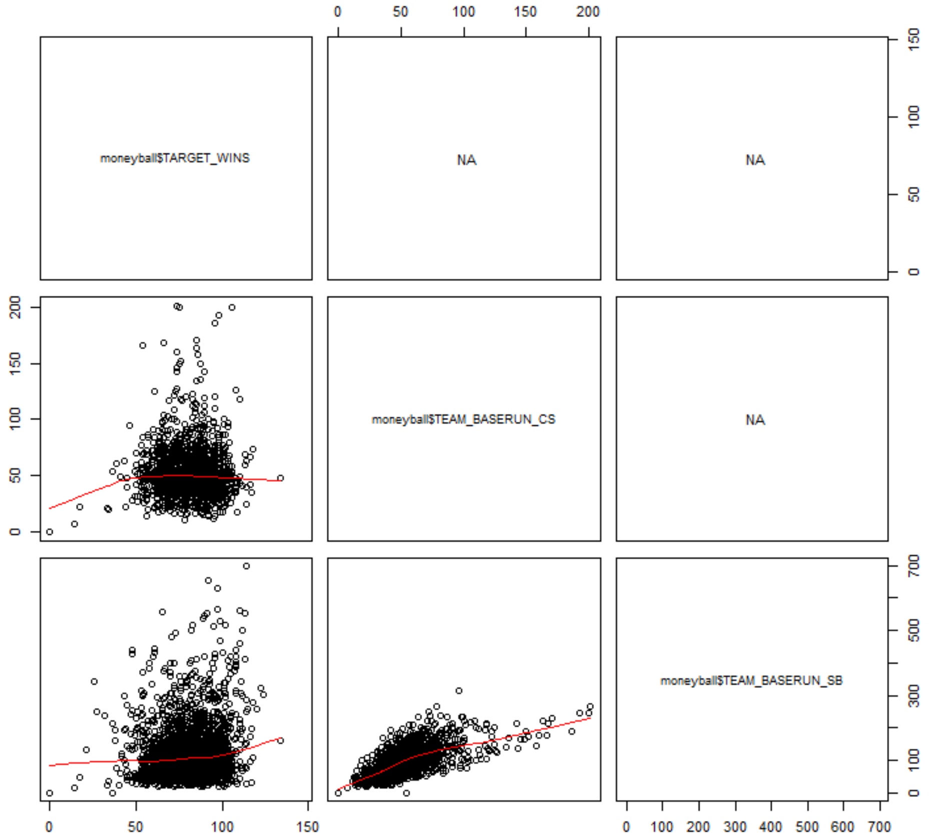

Histogram of Steals
Frequency
0
200
600
0
100
300
500
700
Stolen Bases
Histogram of CS
Frequency
0
200
500
0
50
100
150
200
Caught Stealing
Boxplot of Steals
0
200
500
Boxplot of CS
0
50
150

Figure 5: Team Base Running – Scatter Plot Matrix w/ Correlation and Smoothed Line



Team Pitching Overview: Team Pitching data are defensive statistics for a baseball team that can be used as predictor variables for wins. As can be seen in Table 4, the variables “Hits” and “Walks” have significant skewness, which can also be seen in the histograms for each variable in Figure 6. Both variables seem to be influenced by outlier data, which may need to be addressed before using as predictor variables. Interestingly, the “hits” variable doesn’t appear to have a strong relationship with team wins as the fitted line in Figure 7 is relatively flat. “Walks” however appears to have a slight positive correlation, which is counter intuitive. The “strike out” variable appears to have a normal distribution given its mean and median are relatively close and the skewness result is close to zero, however the histogram and boxplot shows an extremely tight curve with some large outlier values. Given this it’s not likely going to be a very strong predictor for Wins. The scatter plot in Figure 7 confirms this may be the case as the fitted line appears to be flat. Finally, the home runs variable has a distribution that is very similar to the team batting home run variable with modest skewness to left. In addition, the scatter plot matrix shows its influence on team wins is also similar to the batting curve, which is interesting as home runs equate to runs or points against a team so having any positive influence on wins is somewhat counter intuitive.

Table 4: Team Pitching – Summary

Variable Name	Variable Type	Mean	Median	STD	Skewness	NAs
Hits	Predictor	1779.00	1518	1406.84	10.33	0
Home Runs	Predictor	105.70	107	61.3	0.29	0
Walks	Predictor	553.00	536.5	166.36	6.74	0

Strike Outs	Predictor	817.70	813.5	104.35*	0.39*	102

* excludes NAs

Figure 6: Team Pitching – Histogram and Box Plot

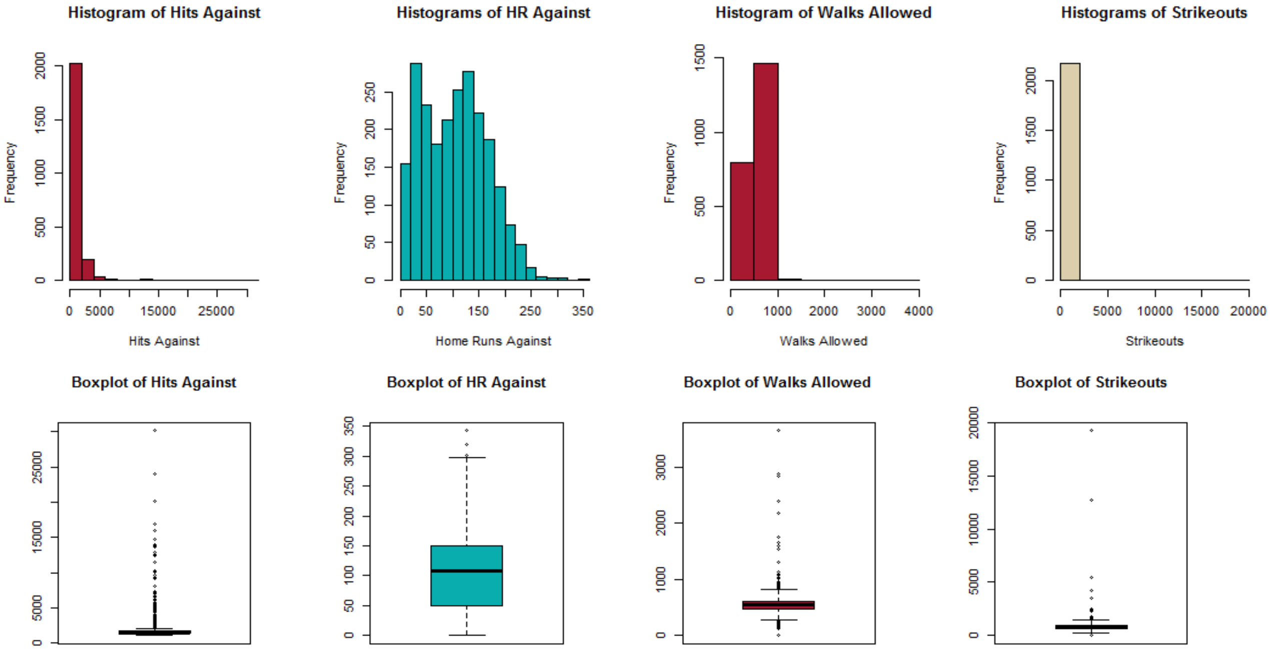
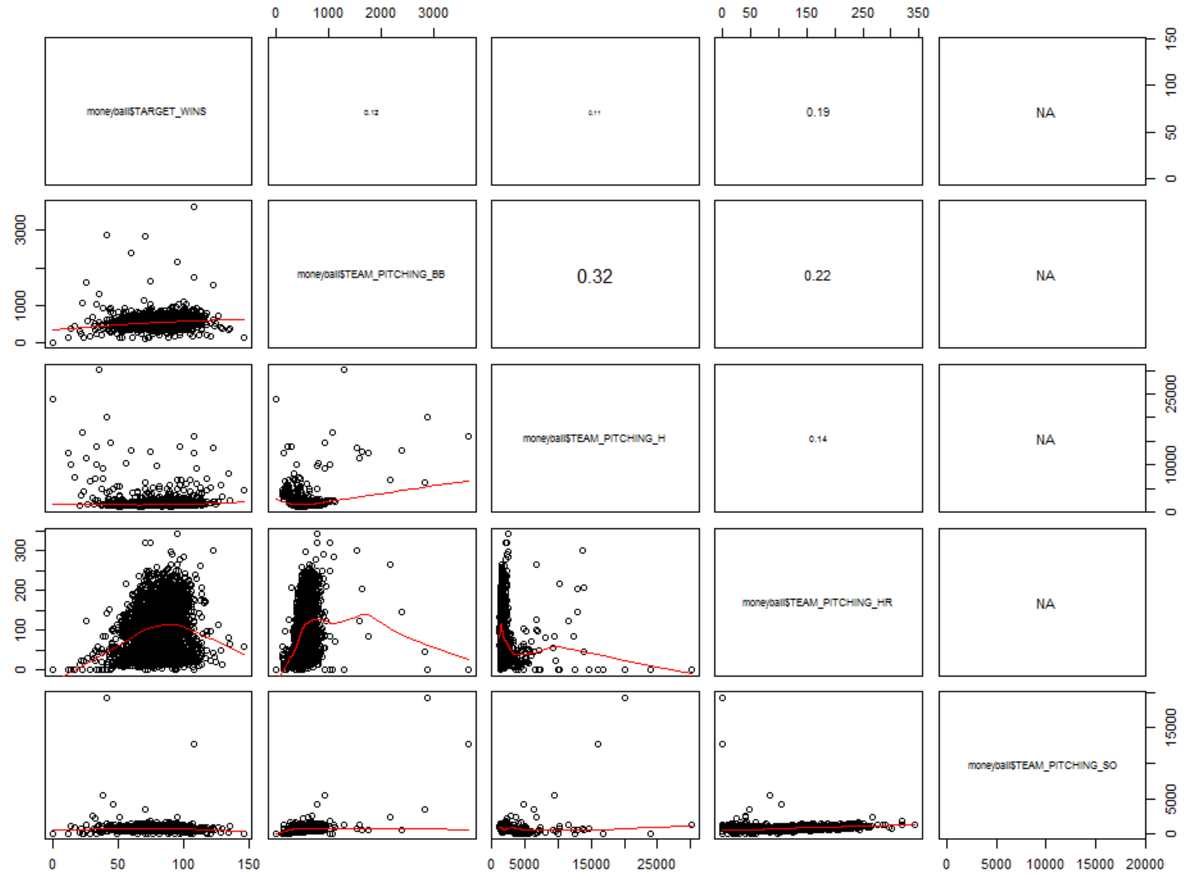


Figure 7: Team Pitching – Scatter Plot Matrix w/ Correlation and Smoothed Line



Team Fielding Overview: The final variables in the data set are for Team Fielding. These data are defensive statistics for a baseball team, which are also potential predictor variables for wins. The "Double Plays" variable appears to have a relatively normal distribution given the skewness result of 0.22 and given the mean and median are quite close. This can also be seen in the histogram and box plot of the double play variable in Figure 8. This variable does have some missing data however, which will need to be addressed before including it as predictor variable. The scatter plot in Figure 9 shows there is an initial positive relationship with team wins followed by a negative relationship after a certain threshold. This is logical given a team with a lot of DPs means they have had a lot of base runners and likely runs against scored against them. The "Errors" variable does not have any missing values, but is highly skewed to the right, which can be seen in Figure 8. The distribution appears to be a Poisson distribution with high occurrences at lower error rates that rapidly decline in frequency. The distribution does however appear to have outliers pulling the tail quite far to the right in the histogram. This can also be seen in the box plot as there are several data points above the whisker of the box. A transformation of this data may be required. Figure 9 shows that there is a negative correlation with wins meaning the more errors the less wins a team has.

Table 5: Team Fielding – Summary

Variable Name	Variable Type	Mean	Median	STD	Skewness	NAs
Errors	Predictor	246.50	159	227.77	2.99	0
Double Plays	Predictor	146.40	149	17.61*	0.22*	286

* excludes NAs

Figure 8: Team Fielding – Histogram and Box Plot

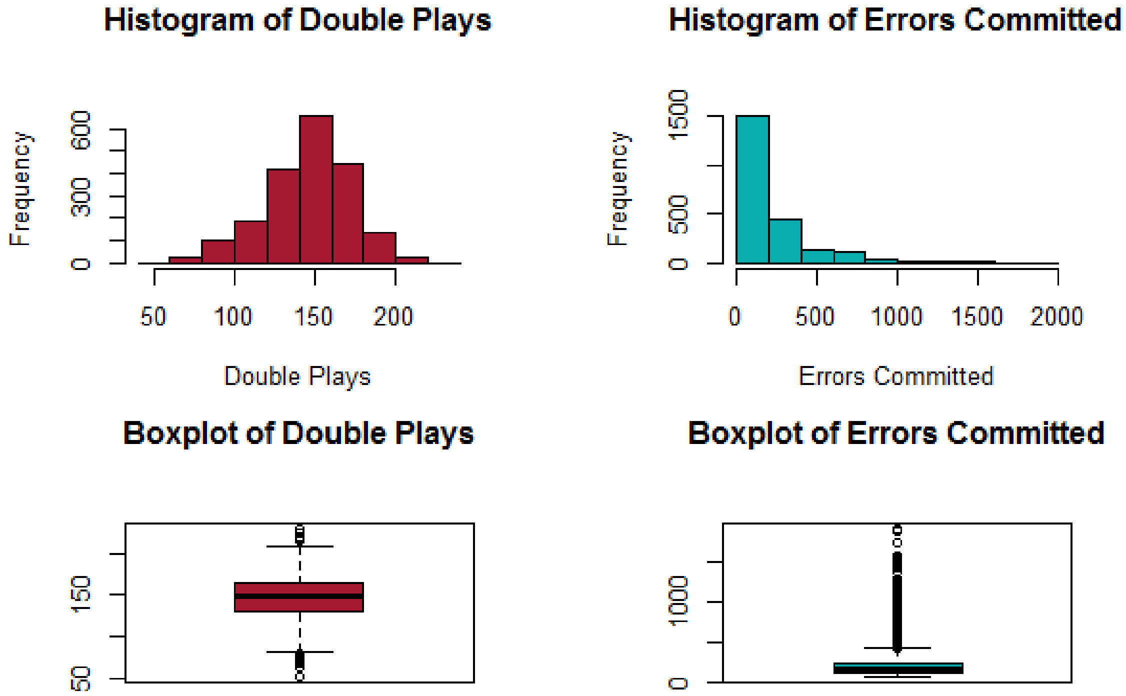
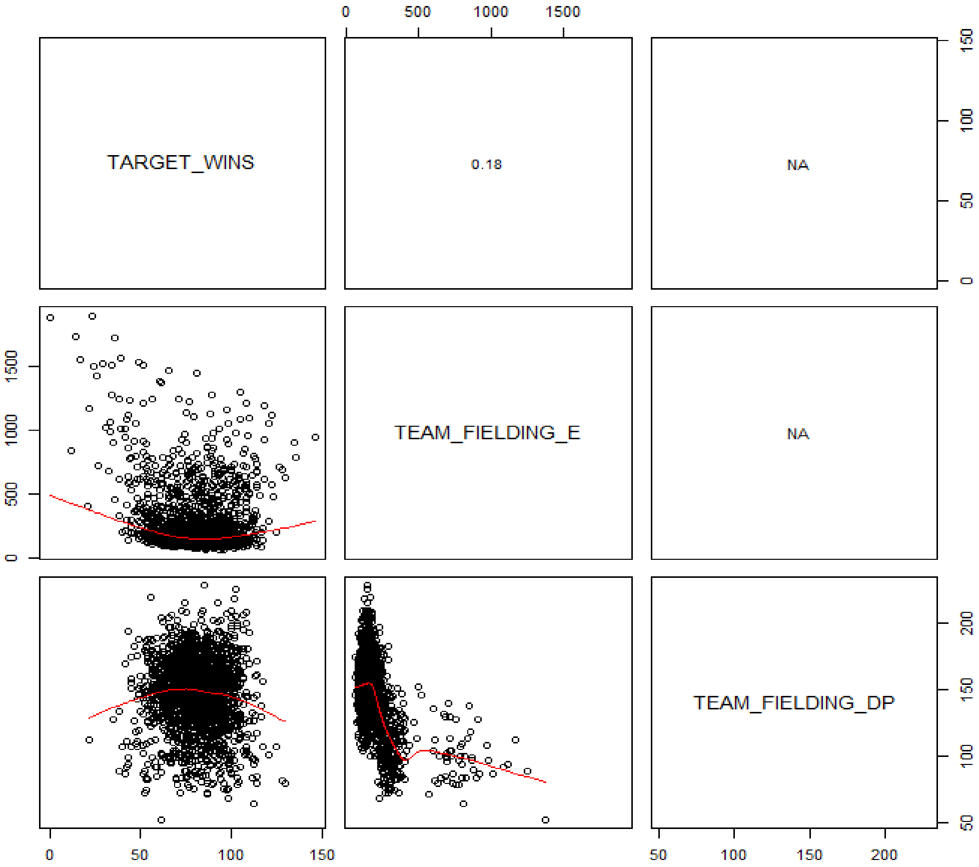


Figure 9: Team Fielding – Scatter Plot Matrix w/ Correlation and Smoothed Line



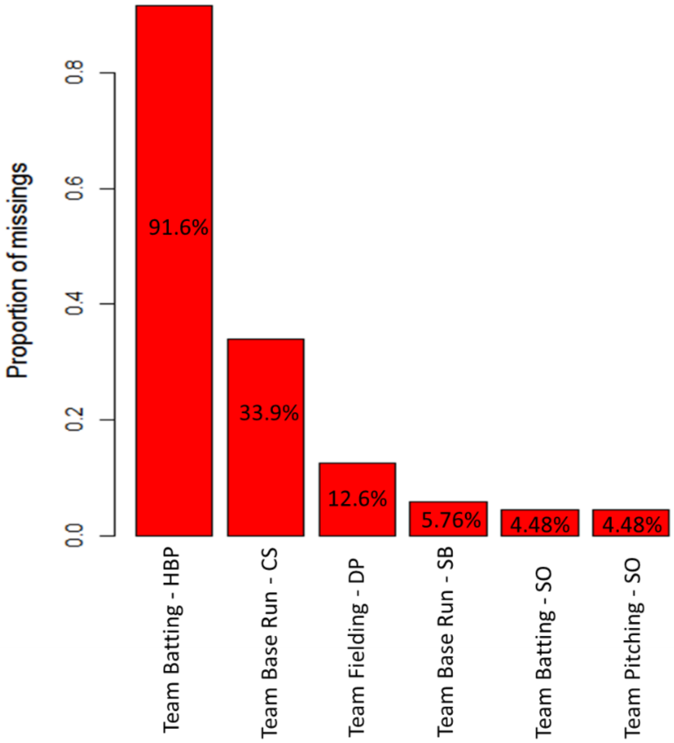
DATA PREPARATION:

Leveraging the insights obtained from the data analysis section several steps were taken to prepare the data for use in the modeling phase. These steps include addressing missing values and transforming the data to reduce the influence of outliers and obtain a more normal distribution for some of the predictor variables.

Missing Values:

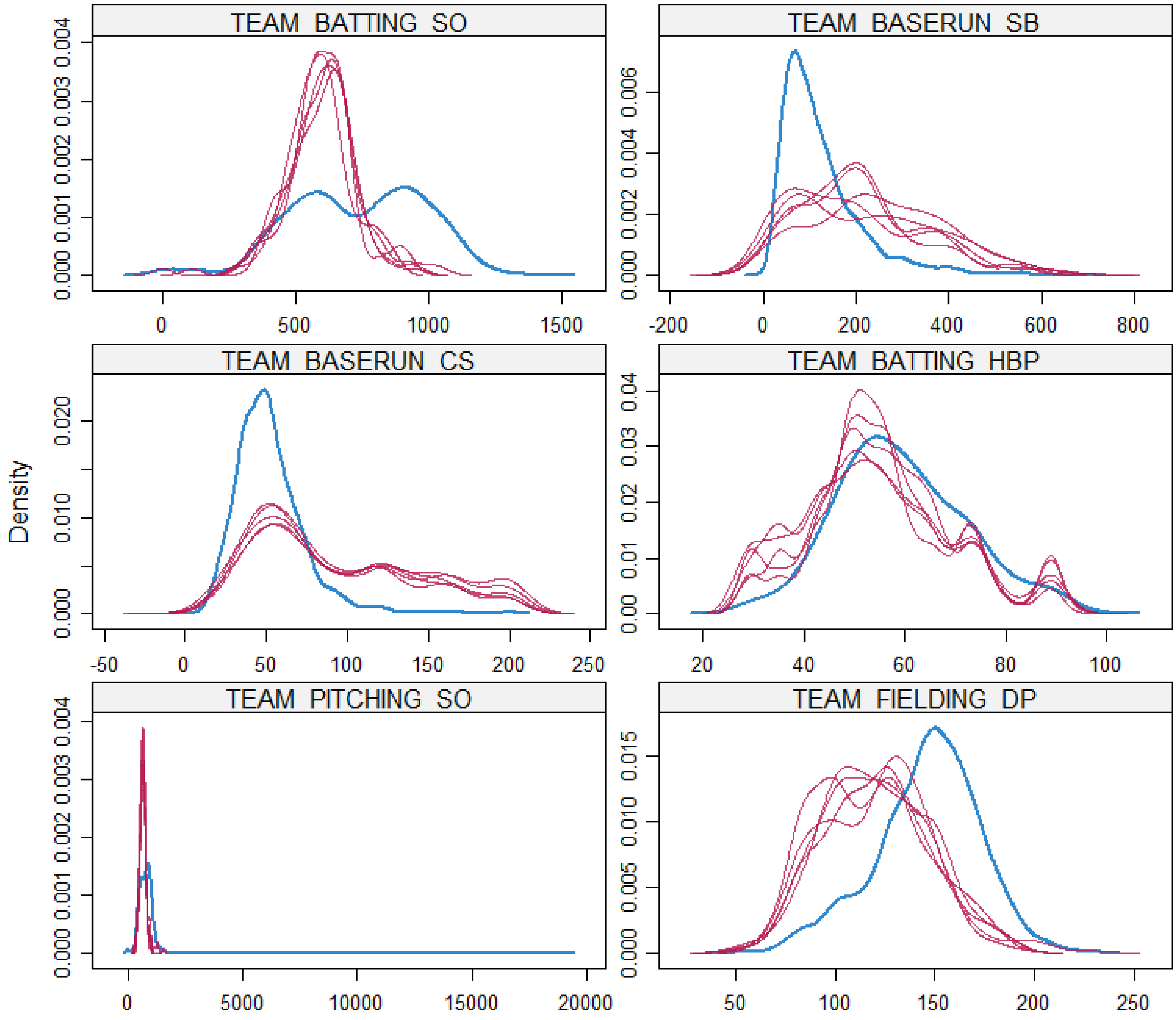
There are 6 variables in the data set that contain some percentage of missing variables. These variables and percentages are shown in Figure 10.

Figure 10: Missing Value Proportion Graph



The Team Batting – Hit by Pitch and the Team Base Run – Caught Stealing variables have a large percentage of missing data with 91.6% and 33.9% respectively. Therefore, using a simple imputation methods such as mean, median or mode replacement with this much missing data will likely misrepresent the true distributions. Given this, the Random Forecast imputation method was leveraged in the MICE imputation package. The output shown in Figure 11 shows the distributions of the five imputed data sets for each of the six variables compared to the original distribution. These imputed data sets were obtained using the "mice" function with 50 iterations and the random forest imputation method. Each of the imputed data sets has a similar distribution, but in most instances the imputed distributions aren't consistent with the original distribution. One reason for this could be that several of the variables contain outliers that skew the original distribution. Given each of the five imputed distributions are relatively similar the first of the five sets was extracted and used for the rest of the analysis. In addition, flag variables were created for each of the six variables with imputed data.

Figure 11: Random Forecast Imputation – MICE



Data Transformation:
In order to correct for skewness and address outlier's data transformations were completed on 8 of the variables in the data set. Four variables leveraged a log or square root transformation to normalize the distribution and four variables were truncated with a quartile calculation to adjust for outliers.

Skewness Adjustments:
To select the appropriate transformation several methods were explored including "log", "sqrt" and "z-transform". The transformed data were plotted and compared to the original distribution. In addition, the skewness was calculated for each transformed variable. The transformation method that produced the most normal distribution and had the lowest skewness was selected. Table 5 and Figure 12 display the results of this analysis along with the selected transformations.

Figure 12: Transformation Comparison

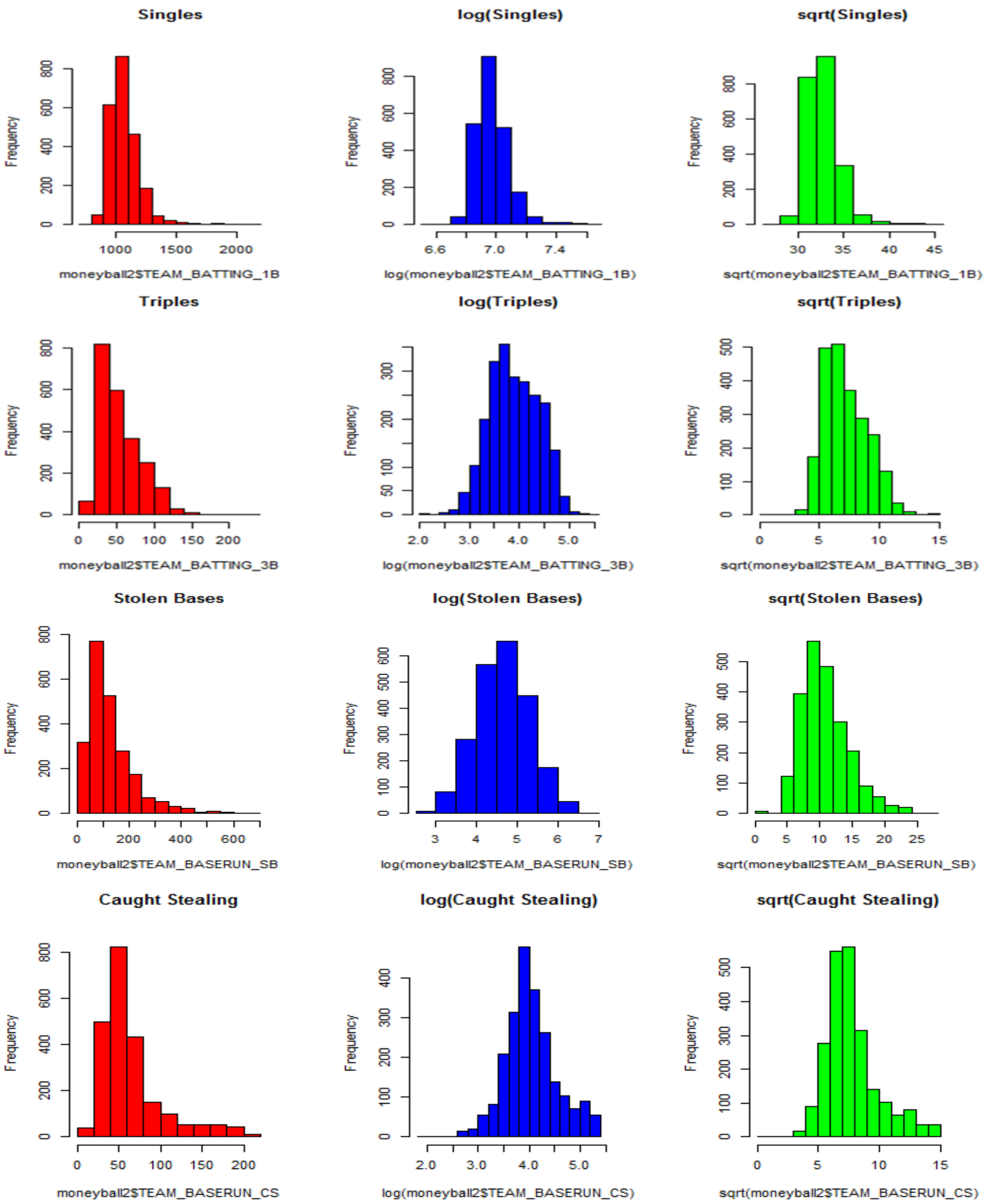


Table 6: Transformation Comparison - Skewness

Skewness			
Variable	Original	log	sqrt
Singles	2.05	1.24	1.61
Triples	1.11	-0.36	0.51
Stolen Bases	1.87	-0.8	0.84
Caught Stealing	1.71	0.08	1.05

Trimming Data – Outlier Management:

Four variables contained outliers that highly influenced the distribution of the data. These variables were transformed by trimming the data to either the 95th or 99th percentile. The same approach used for addressing skewness was used here. Each of the variables was trimmed to the 95th and 99th percentile and plotted in a histogram for visual comparison. These graphs can be seen in Figure 13 below. In addition, the skewness was calculated for each of the trimmed data as well as the original data and populated into Table 7, which can be seen below. The trimming method that produced the most normal distribution and had the lowest skewness was selected. Table 7 and Figure 13 display the results of this analysis along with the selected transformations.

Figure 13: Transformation – Trimming

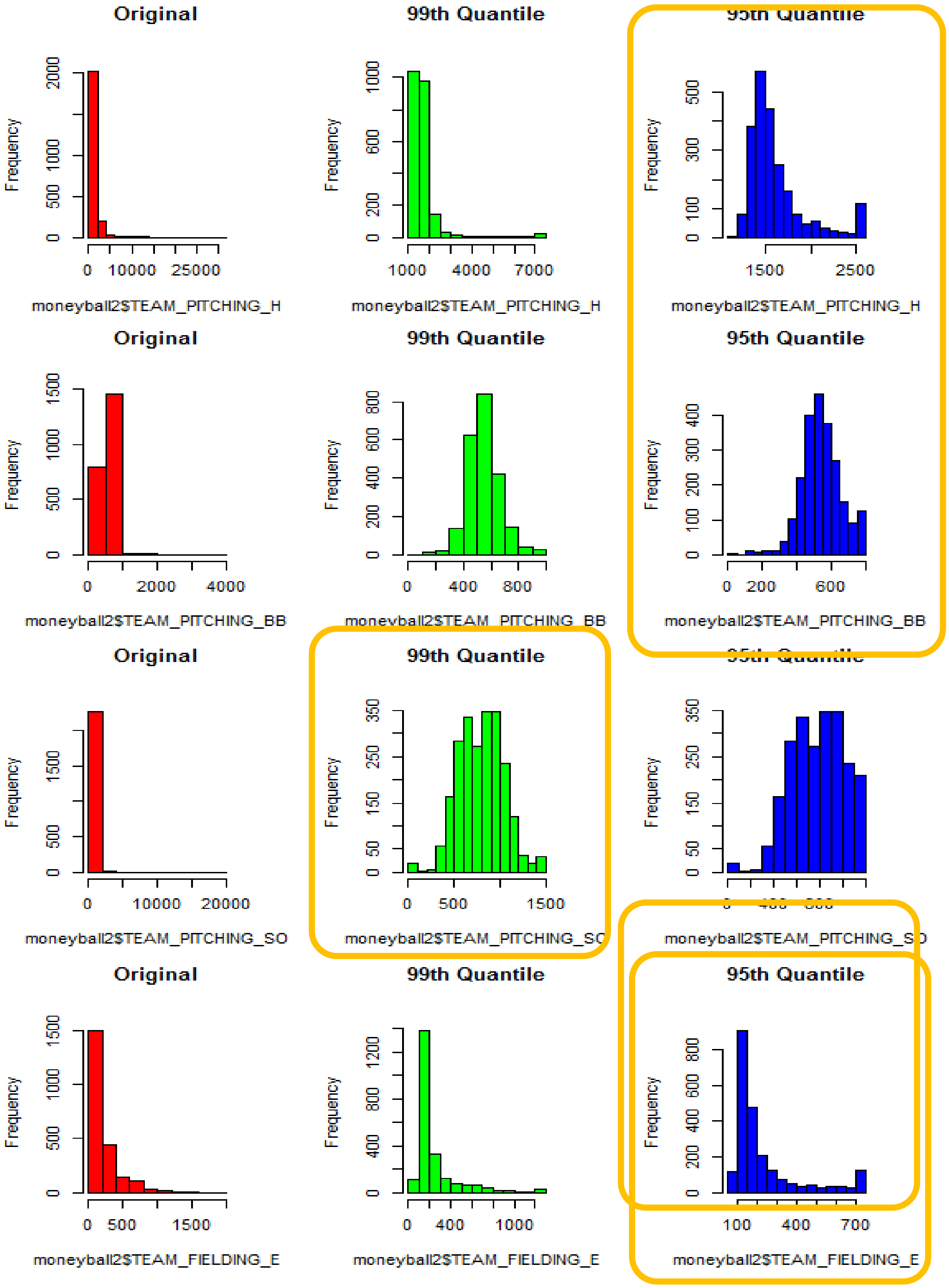


Table 7: Trimming Transformation - Skewness

Skewness			
Variable	**Original**	**95th**	**99th**
Pitching - Hits	10.33	1.76	5.03
Pitching - Walks	6.74	-0.2	0.29
Pitching - SO	22.53	-0.32	0.02
Fielding - Errors	2.99	1.78	2.54

BUILD MODELS:

Three techniques were used to in the model creation process in this section, which include the stepwise regression, all subsets regression and random forest regression.

Stepwise Regression:

The stepwise regression technique creates multiple subsets of the variables by adding and removing each one and testing each possible subset for the lowest AIC. The subset of variables that produce the lowest AIC is selected. Using this selection process 16 or the 20 predictor variables available were selected. The Team Batting Hits and the Team Pitching HR were removed from the available predictor variables during the modelling process due to high VIF values. The list of variables selected can be seen in the model summary output in Figure 14 along with the summary statistics. Reviewing the variables and the coefficients there are a few that appear questionable. First, the intercept is negative, which is unusual given a team can't have less than zero wins at any time during the season. The flag variables created for "team batting strike out", "team base running – caught stealing" and "team fielding – double plays" all have positive coefficients, which indicates a positive relationship with wins and is the opposite of their main variables. Recall however that the in the data analysis the smoothed line in the scatter plot matrix was curved for each variable vs. team wins. This implied that there is an initial positive or negative relationship up to a certain point which then turned positive. The model created is likely adjusting for and trying to include this phenomena with these coefficients. Overall, the model appears to perform well. The adjusted R-squared result of 0.384 means that 38.4% of the variation in Team Wins can be explained by this model and the VIF results in Figure 15 indicate that there is not significant collinearity between the predictor variables as the VIF results are largely less than 11.

Figure 14: Stepwise Selection Summary Output:

```
Residuals:
    Min      1Q  Median      3Q     Max 
-53.157  -8.219   0.146   8.147  59.054 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)        -3.533e+02  3.029e+01 -11.661  < 2e-16 ***
TEAM_BATTING_1B     5.596e+01  4.465e+00  12.533  < 2e-16 ***
TEAM_BATTING_2B     3.542e-02  7.727e-03   4.584 4.81e-06 ***
TEAM_BATTING_3B     6.400e+00  8.024e-01   7.976 2.38e-15 ***
TEAM_BATTING_HR     1.117e-01  9.042e-03  12.356  < 2e-16 ***
TEAM_BATTING_BB     3.036e-02  3.187e-03   9.526  < 2e-16 ***
TEAM_BATTING_SO    -8.267e-03  2.307e-03  -3.583 0.000346 ***
TEAM_BASERUN_SB     6.432e+00  5.445e-01  11.813  < 2e-16 ***
TEAM_BASERUN_CS    -2.880e+00  7.119e-01  -4.046 5.38e-05 ***
TEAM_PITCHING_H    -3.338e-03  1.936e-03  -1.724 0.084873 .  
TEAM_FIELDING_E    -6.457e-02  4.888e-03 -13.209  < 2e-16 ***
TEAM_FIELDING_DP   -1.113e-01  1.255e-02  -8.870  < 2e-16 ***
TEAM_BATTING_SO_M   7.201e+00  1.491e+00   4.829 1.47e-06 ***
TEAM_BASERUN_SB_M   3.069e+01  1.996e+00  15.370  < 2e-16 ***
TEAM_BASERUN_CS_M   3.158e+00  8.680e-01   3.638 0.000281 ***
TEAM_BATTING_HBP_M  4.536e+00  1.116e+00   4.066 4.96e-05 ***
TEAM_FIELDING_DP_M  6.953e+00  1.471e+00   4.728 2.41e-06 ***

Residual standard error: 12.38 on 2259 degrees of freedom
Multiple R-squared:  0.3866,	Adjusted R-squared:  0.3823 
F-statistic:    89 on 16 and 2259 DF,  p-value: < 2.2e-16
```

Figure 15: Stepwise Regression – VIF Table

```
   TEAM_BATTING_1B    TEAM_BATTING_2B    TEAM_BATTING_3B    TEAM_BATTING_HR 
          3.631956           1.941109           2.442922           4.448392 
   TEAM_BATTING_BB    TEAM_BATTING_SO    TEAM_BASERUN_SB    TEAM_BASERUN_CS 
          2.269271           4.779305           2.177998           2.040806 
   TEAM_PITCHING_H    TEAM_FIELDING_E   TEAM_FIELDING_DP  TEAM_BATTING_SO_M 
          5.353561          10.381864           1.801834           1.413714 
 TEAM_BASERUN_SB_M  TEAM_BASERUN_CS_M TEAM_BATTING_HBP_M TEAM_FIELDING_DP_M 
          3.210610           2.507753           1.421372           3.528491 
```

All Subsets Regression:

In all subsets regression, every possible variable grouping is inspected and analyzed for the best combination. For this selection process all of the 22 predictor variables were included in the analysis, with the exception of Team Batting Hits, which produces high VIF values given it's correlation with the other team hitting variables. Based on the subset selection process 9 of the 21 variables were included in the subset regression model. The plot in Figure 16 displays the selected variables, which can be identified by looking at the top row of the graph. Any variable on the x-axis with a black highlighted section in the top row is selected for the top model. The summary output shown in Figure 17 contains the statistics for the linear regression model using the variables selected. Like the stepwise model, the intercept is a negative number, which is unexpected. The "team pitching – HR" and the "team fielding – double play" variables have a positive and negative coefficient respectfully, which is also unusual. Again, recall that the curves of the smoothed lines in the scatter plots had both a positive and a negative relationship at different values of each variable. Given this, the model created is linear in its parameters, these along with the other coefficient estimations will consider this. In terms of performance, the adjusted R-squared result of 0.3619 means that 36.2% of the variation in Team Wins can be explained by this model. Finally, the VIF results in Figure 18 indicate that there is not significant collinearity between the predictor variables as the VIF results are largely less than 11.

Figure 16: Plot of All Subset Selection

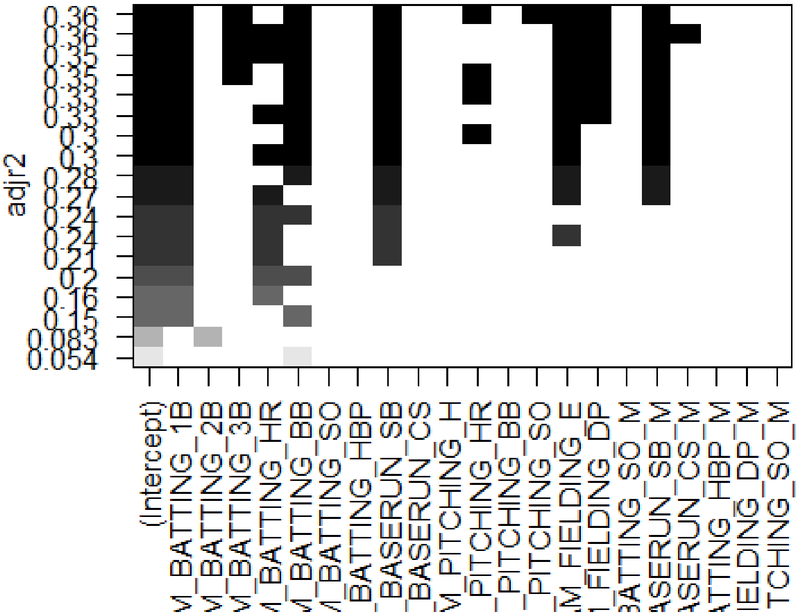


Figure 17: All Selections Regression Model – Summary Output

```
Residuals:
    Min      1Q  Median      3Q     Max 
-61.524  -8.096   0.232   7.991  73.982 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -3.297e+02  2.551e+01 -12.922  < 2e-16 ***
TEAM_BATTING_1B    5.236e+01  3.688e+00  14.195  < 2e-16 ***
TEAM_BATTING_3B    6.179e+00  7.771e-01   7.951 2.88e-15 ***
TEAM_BATTING_BB    3.606e-02  3.137e-03  11.494  < 2e-16 ***
TEAM_BASERUN_SB    5.989e+00  4.648e-01  12.886  < 2e-16 ***
TEAM_PITCHING_HR   1.073e-01  7.032e-03  15.256  < 2e-16 ***
TEAM_PITCHING_SO  -1.037e-02  1.751e-03  -5.923 3.65e-09 ***
TEAM_FIELDING_E   -5.110e-02  3.030e-03 -16.867  < 2e-16 ***
TEAM_FIELDING_DP  -1.214e-01  1.217e-02  -9.975  < 2e-16 ***
TEAM_BASERUN_SB_M  2.840e+01  1.748e+00  16.242  < 2e-16 ***
---
Residual standard error: 12.58 on 2266 degrees of freedom
Multiple R-squared:  0.3645,    Adjusted R-squared:  0.3619 
F-statistic: 144.4 on 9 and 2266 DF,  p-value: < 2.2e-16
```

Figure 18: All Subsets Regression – VIF Table

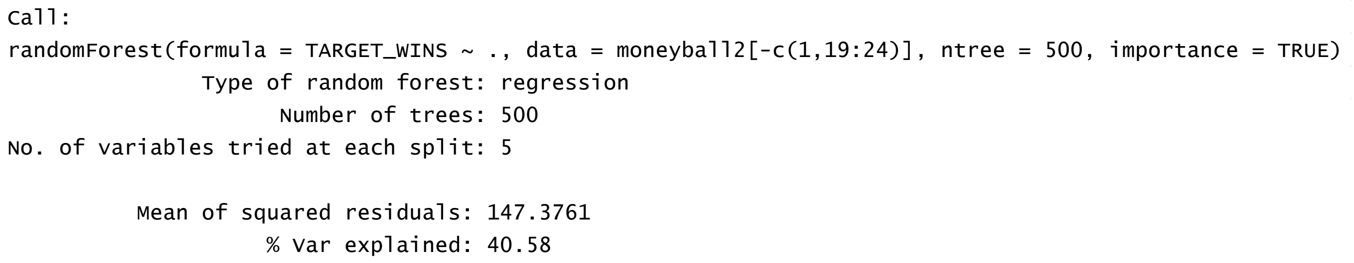
```
  TEAM_BATTING_1B   TEAM_BATTING_3B   TEAM_BATTING_BB   TEAM_BASERUN_SB 
         2.399279          2.218554          2.128468          1.536392 
 TEAM_PITCHING_HR  TEAM_PITCHING_SO   TEAM_FIELDING_E  TEAM_FIELDING_DP 
         2.669945          2.631338          3.860313          1.641076 
TEAM_BASERUN_SB_M 
 2.38364
```

Random Forest Regression:

In Random Forest regression multiple predictive models are developed and the results are aggregated together to improve accuracy. This is accomplished by sampling a subset of variables and creating a large number of decision trees. The trees are then aggregated to form a complete model for predicting the response variable. An advantage of random forecast regression is that you don't need to adjust for data issues such as missing values or outliers. In this analysis however, the same data set used for the stepwise and subset regression was used to ensure consistency and comparability of the models. The only exception is that the flag variables created were removed given they provided no value or showed no importance in the model when initially ran with them. This is consistent with the point made about random forest regression not needing an adjustment for data issues. Furthermore, no variables were excluded due to multicollinearity. The summary output in Figure 18 shows the selections made for this regression as well as the performance statistics of the model. The number of trees grown was set to 500, which was to minimize the error without adding complexity (more trees). The error vs. trees curve is Figure 20 displays this relationship. The variables tried at each split as well as the MSE and the variance explained by the model (pseudo R^2 are shown in Figure 18. The "R-squared" result of 41.72

means that 41.72% of the variation in Team Wins can be explained by this model. Figure 19 shows the variables and the level of importance. The higher the number the more important the variable. "Team Batting – Hits" and "Team Fielding – Errors" have the highest level of importance. The majority of the remaining variables appear to have an equal level of importance, which implies it's not one thing that wins baseball games, but requires doing many things well. This is consistent with the other models and does aligns well with reality.

Figure 18: Random Forest Regression – Summary Output



```
Call:
randomForest(formula = TARGET_WINS ~ ., data = moneyball2[-c(1,19:24)], ntree = 500, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 5

          Mean of squared residuals: 147.3761
                    % Var explained: 40.58
```

Figure 19: Random Forest Regression - Importance

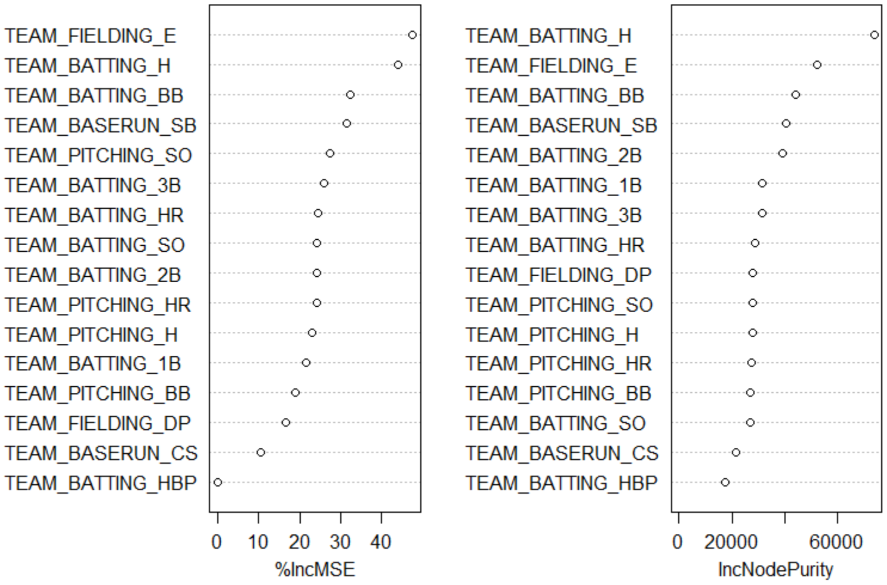
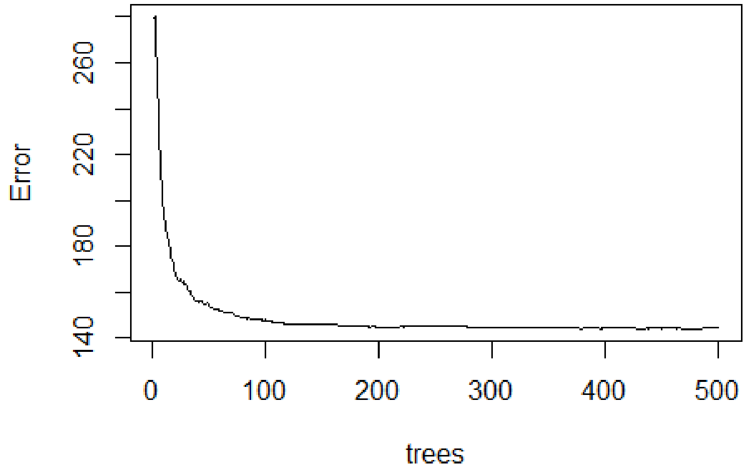


Figure 20: Random Forest Regression – Error Plot



MODEL SELECTION:

To compare and select the best model, metrics that allowed for a direct comparison of the models given availability in the summary output statistics. Given this, the adjusted R^2 or pseudo R^2 (Random Forest) and the MSE for each model was selected and compared. In addition, consideration is given to the simplicity of the model or the number of variables selected, which equates to "explain-ability". Table 8 outlines the key statistics for each of the models. From a performance perspective the Random Forest Model performs the best given the strongest Adj. R^2 and lowest MSE result. The simplest model is the Subset model which uses only 9 variables and produces a respectable Adj. ^{2} and MSE given its size relative to the others. Overall, the Random Forest Model appears to be the best model considering performance and simplicity. With the same number of variables as the stepwise model it has a better overall performance. Compared to the Subsets model the Random Forest Model is better even when considering the additional complexity.

Table 8: Model Comparison Metrics

Metric	Stepwise	Subsets	RF - Imp.
Adj. R^2	38.2%	36.2%	40.6%
MSE	152.1	157.6	147.4
Variables	16	9	16

CONCLUSION:

The objective of this analysis was to build a model that could be used to accurately predict the number of wins for a baseball team given performance metrics of teams between 1900 and 1950. The analysis completed leveraged three different linear regression techniques to create models which were compared in both performance and simplicity. Overall, the Random Forest model was selected based on its strong performance relative to the others, which more than offset its complexity.

LM vs. GLM – BONUS:

Given my model of choice was the random forest regression for this bonus section I selected the subsets model to compare the LM() function with the GLM() function in R as the data and variables were already available. The results of the LM() function are above in the "Build Model" section and will be referenced for this bonus review. Leveraging the GLM() function the results are indeed similar. The summary output can be seen in Figure B1. The output below is very similar to the output from the LM() function. The residual distribution is identical as is the coefficients data including the estimate through to the P-Value. The difference in the output is the GLM summary does not contain an R^2 or adj. R^2 value. Comparing the AIC output in Figure B2 shows an identical result. All of this is because Linear Regression is a special case of GLM therefore the results are the same.

Figure B1: GLM Summary Output – Subsets Model

```
glm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + 
    TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_HR + TEAM_PITCHING_SO + 
    TEAM_FIELDING_E + TEAM_FIELDING_DP + TEAM_BASERUN_SB_M, data = moneyball2)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-61.524   -8.096    0.232    7.991   73.982  

Coefficients:
                    Estimate  Std. Error t value Pr(>|t|)    
(Intercept)       -329.682685   25.513551 -12.922  < 2e-16 ***
TEAM_BATTING_1B     52.359101    3.688494  14.195  < 2e-16 ***
TEAM_BATTING_3B      6.179363    0.777144   7.951 2.88e-15 ***
TEAM_BATTING_BB      0.036061    0.003137  11.494  < 2e-16 ***
TEAM_BASERUN_SB      5.989401    0.464784  12.886  < 2e-16 ***
TEAM_PITCHING_HR     0.107278    0.007032  15.256  < 2e-16 ***
TEAM_PITCHING_SO    -0.010370    0.001751  -5.923 3.65e-09 ***
TEAM_FIELDING_E     -0.051098    0.003030 -16.867  < 2e-16 ***
TEAM_FIELDING_DP    -0.121387    0.012169  -9.975  < 2e-16 ***
TEAM_BASERUN_SB_M   28.396808    1.748344  16.242  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 158.3211)

    Null deviance: 564496  on 2275  degrees of freedom
Residual deviance: 358756  on 2266  degrees of freedom
```

Figure B2: AIC Comparison

```
       df      AIC
bonus1 11 17998.07
bonus2 11 17998.07
```

RATTLE – BONUS:

Rattle was used as an additional option to complete this analysis. There are some limitations that Rattle has that had to be considered.

1) Any additional variables need to be created outside of RATTLE and then uploaded in a complete data set

2) To address missing variables there are limited options, which include zeroing, imputing the mean, median or mode. A random forest option for example is not presented. That said, the random forest model option does allow for imputation.

DATA ANALYSIS:

The same steps that were followed in the main report analysis were followed with this analysis. The Histograms for the data are shown in Figure R1.

Figure R1: Histograms of Money Ball Variables

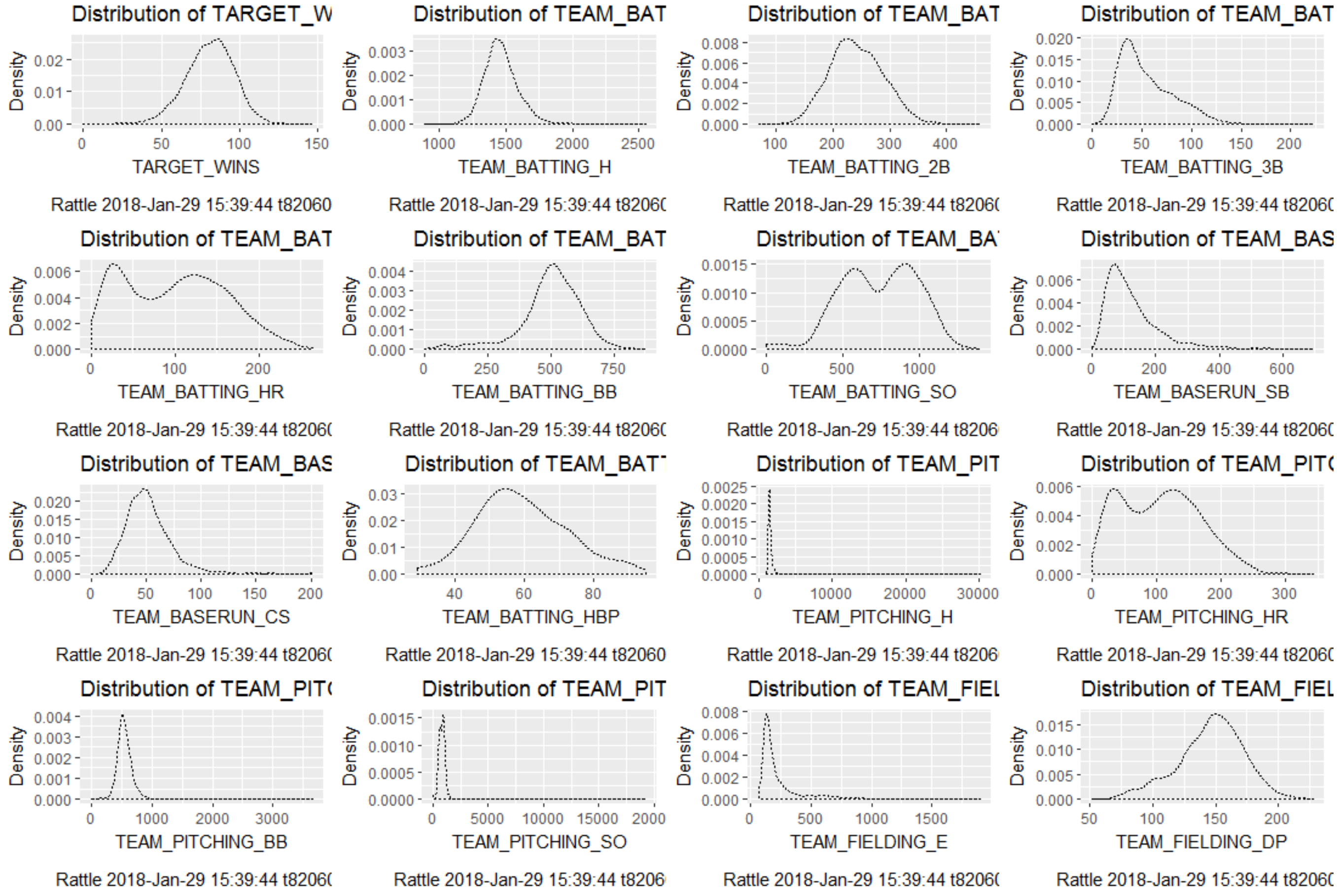


Figure R2 provides an example output of the data available in the "Explore" table of Rattle. The information is produced quickly and also adds key metrics like Kurtosis and skewness in a single report. The numbers are all consistent with what was produced in the main report however.

Figure R2: Explore Data – Tables

```
Storage  NAs
TEAM_BATTING_H  integer   0
TEAM_BATTING_2B  integer   0
TEAM_BATTING_3B  integer   0
TEAM_BATTING_HR  integer   0
TEAM_BATTING_BB  integer   0
TEAM_BATTING_SO  integer  65
TEAM_BASERUN_SB  integer  88
TEAM_BASERUN_CS  integer  500
TEAM_BATTING_HBP integer 1455
TEAM_PITCHING_H  integer   0
TEAM_PITCHING_HR integer   0
TEAM_PITCHING_BB integer   0
TEAM_PITCHING_SO integer  65
TEAM_FIELDING_E  integer   0
TEAM_FIELDING_DP integer  199
TARGET_WINS     integer   0
```

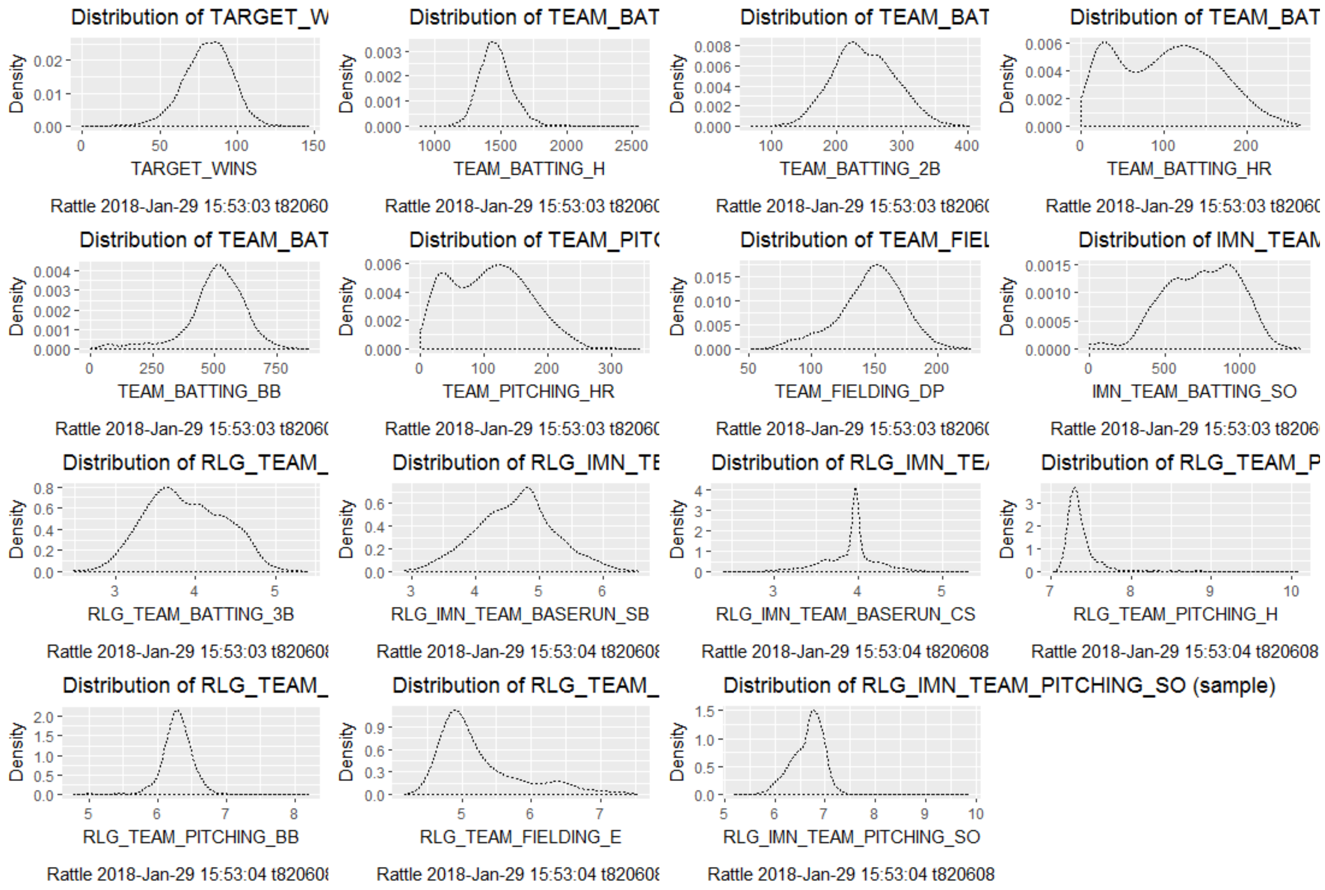
Basic statistics for each numeric variable of the dataset.

```
$TEAM_BATTING_H
nobs          1593.000000
NAs              0.000000
Minimum        891.000000
Maximum       2554.000000
1. Quartile   1381.000000
3. Quartile   1539.000000
Mean          1470.234777
Median        1457.000000
Sum        2342084.000000
SE Mean          3.687268
LCL Mean      1463.002366
UCL Mean      1477.467188
Variance     21658.340574
Stdev          147.167729
Skewness         1.530201
Kurtosis         6.985371
```

Data Preparation:

Similar data preparation techniques were used in rattle with missing values being imputed and outliers, skewness and kurtosis being addressed with transformations. For Missing value imputation however, mean imputation was used given the limited options available (no trees). Figure R3 shows the distributions of the transformed data set that will be used in the modeling phase. The HBP variable was also dropped from the set given the significant number of missing variables.

Figure R3: Prepared Data Sets



Build Models:
For the modeling building phase a regression model leveraging the transformed data set a linear regression model was used as well a Random Forest regression model. The analysis and outputs of these models are below.

Linear Regression Model: The output can be seen in Figure R4. Similar to the linear models created in the main report, the intercept is a negative number, which is not possible as a team can't have less than zero wins. With the exception of doubles all the coefficients appear to have proper signs (+/-) for each of the variables. Doubles has a negative coefficient, which is interesting as offensive statistics should have a positive relationship with wins. From a performance perspective the model has an adjusted R^2 of 36.6%, which is similar to the other regression models created.

Figure R4 - Regression Model

```
Call:
lm(formula = TARGET_WINS ~ ., data = crs$dataset[crs$train, c(crs$input,
   crs$target)])

Residuals:
   Min     1Q  Median
-43.713  -7.306   0.033
    3Q    Max
  7.228  42.532

Coefficients:                        Estimate  Std. Error  t value    Pr(>|t|)
(Intercept)                          -158.982   32.70567    -4.861   1.30E-06
TEAM_BATTING_H                       0.026817   0.006844     3.918   9.35E-05
TEAM_BATTING_2B                      -0.04719   0.010558    -4.469   8.50E-06
TEAM_BATTING_HR                      0.364408   0.049117     7.419   2.05E-13
TEAM_BATTING_BB                      0.033484   0.012401       2.7   0.007016
TEAM_PITCHING_HR                     -0.28792   0.045968    -6.264   5.02E-10
TEAM_FIELDING_DP                     -0.13753   0.014986    -9.177   2.00E-16
IMN_TEAM_BATTING_SO                  -0.04765   0.007587     -6.28   4.52E-10
RLG_TEAM_BATTING_3B                  8.884504   1.123473     7.908   5.32E-15
RLG_IMN_TEAM_BASERUN_SB              6.671021   0.782696     8.523   2.00E-16
RLG_IMN_TEAM_BASERUN_CS              -3.79586   1.097351    -3.459   0.000558
RLG_TEAM_PITCHING_H                  20.83836   5.536249     3.764   0.000174
RLG_TEAM_PITCHING_BB                 -1.08174   6.019141     -0.18   0.857401
RLG_TEAM_FIELDING_E                  -20.2563   1.574127   -12.868   2.00E-16
RLG_IMN_TEAM_PITCHING_SO             22.95788   5.264231     4.361   1.39E-05
---
Residual standard error: 11 on 1379 degrees of freedom
  (199 observations deleted due to missingness)
Multiple R-squared:  0.3719,              Adjusted R-squared:  0.3656
F-statistic: 58.33 on 14 and 1379 DF,  p-value: < 2.2e-16
```

Random Forest Regression Model: The output can be seen in Figure R5. From a performance perspective the model has a Pseudo R^2 of 42.77% and an MSE of 142, which is similar Random Forest Model created in the main report using the cleaned or prepared data in both instances. The top two importance metrics are also the same, which can be seen in Figure R5 and Figure R6. Figure R7 also has a very similar error curve to the RF model in the main report.

Figure R5 – Random Forest Regression

```
Call:
 randomForest(formula = TARGET_WINS ~ ., data = crs$dataset[crs$sample, c(crs$input, crs$target)],
ntree = 500, mtry = 6, importance = TRUE, replace = FALSE, na.action = randomForest::na.roughfix)
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 6
Mean of squared residuals: 141.8828
% Var explained: 42.77
                                %IncMSE  IncNodePurity
RLG_TEAM_FIELDING_E               51.91  35252.01
TEAM_BATTING_H                    48.22  39646.69
RLG_TEAM_BATTING_3B               25.75  13353.45
RLG_IMN_TEAM_PITCHING_SO          25.42   12968.2
TEAM_BATTING_BB                   25.24  17977.33
RLG_IMN_TEAM_BASERUN_SB           25.24  12896.41
RLG_TEAM_PITCHING_H               21.98  16966.44
TEAM_BATTING_HR                   21.74  11241.09
IMN_TEAM_BATTING_SO               20.88  11884.58
TEAM_BATTING_2B                   20.77  18948.93
TEAM_PITCHING_HR                  18.81  10169.02
RLG_TEAM_PITCHING_BB              17.96  11697.89
TEAM_FIELDING_DP                  17.48  11430.04
RLG_IMN_TEAM_BASERUN_CS            9.83   5657.18
```

Figure R6: Variable Importance Output

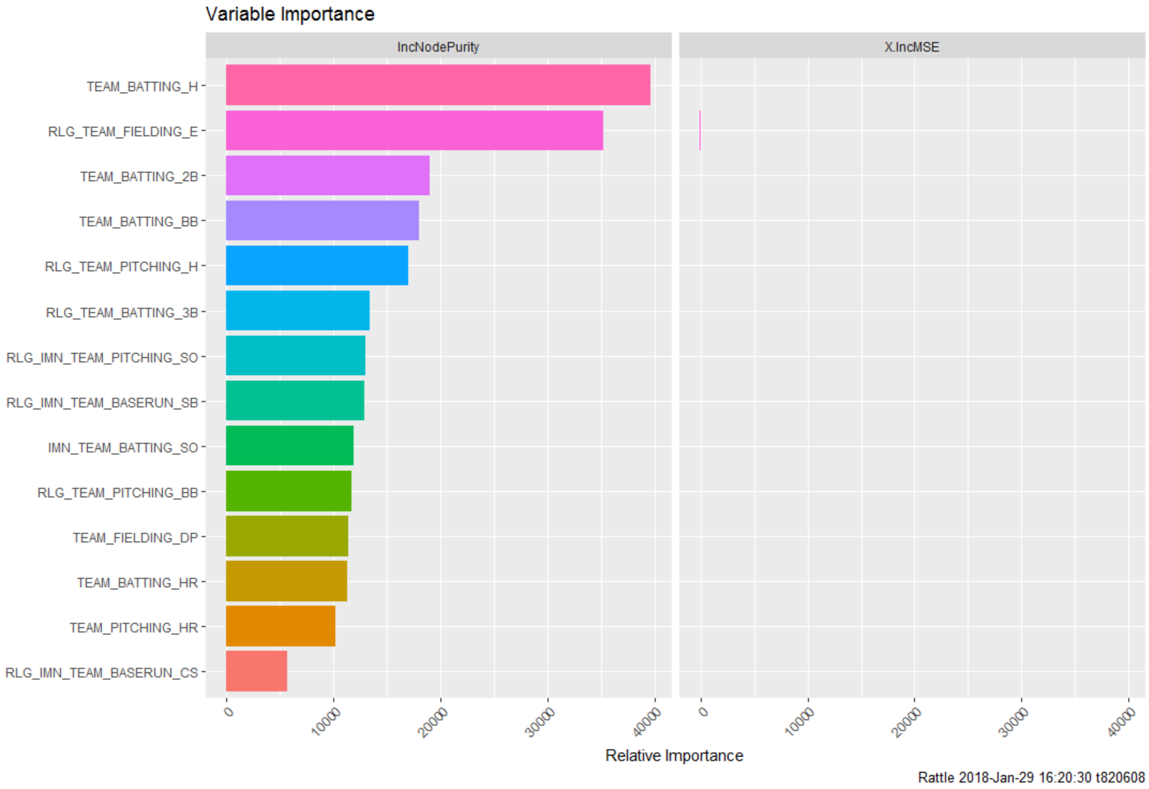
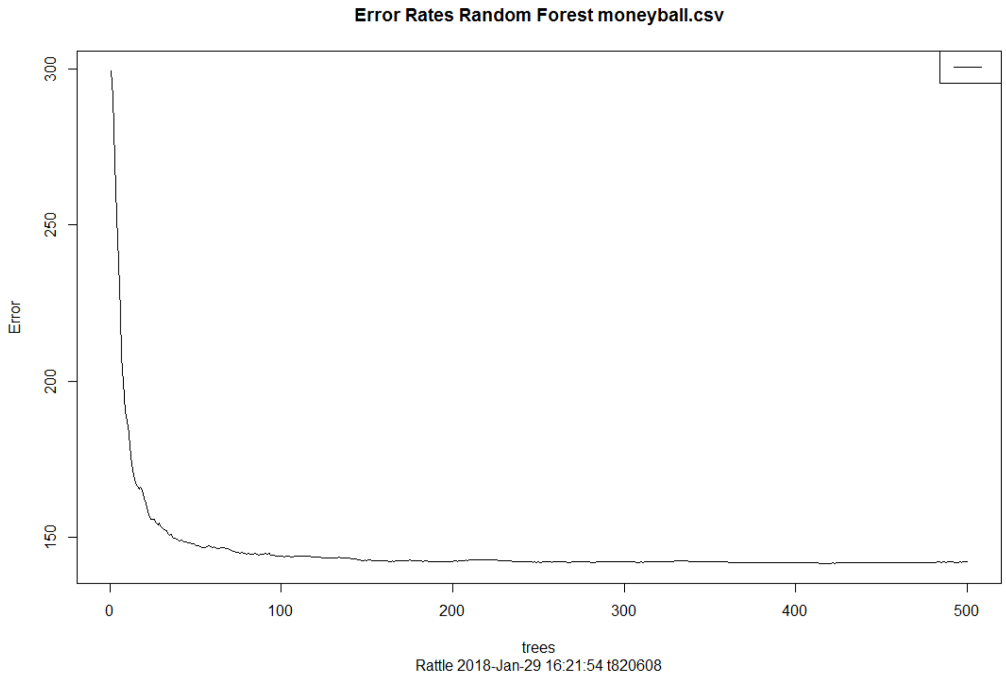


Figure R7: Error Curve



Model Selection:
In order to select a model the two models were evaluated on by scoring the test data, which is a partition of the main training data set. The partition includes 30% of the original training data set. Following the scoring, the RF model produces a MSE of 121 with the linear model coming in at 131. Given these results and considering the performance on the training data, the RF model is again the best model for predicting team wins given the money ball data (Scored Data attached in the submission).

Code:

```
library(rJava)
library(readr)
library(pbkrtest)
library(car)
library(leaps)
library(MASS)
library(xlsxjars)
library(xlsx)
library(mice)
library(VIM)
library(e1071)
library(randomForest)
library(rattle)
rattle()

#####
#Designated proper working environment on my computer. You will want to make sure it is in proper
place for your computer.
#####

setwd("c:/…")
moneyball <- read.csv("moneyball.csv",header=T)

############## Part 1: Data Exploration
##########################################################################
str(moneyball[-1])
summary(moneyball)

# Wins - Use lower bound for lower outliers, upper bound for higher outliers.
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))

################# Batting ####################
moneyball$TEAM_BATTING_1B <- moneyball$TEAM_BATTING_H - moneyball$TEAM_BATTING_HR -
moneyball$TEAM_BATTING_3B -
  moneyball$TEAM_BATTING_2B

# Hits and Doubles
par(mfrow=c(2,3))
hist(moneyball$TEAM_BATTING_H, col = "#A71930", xlab = "Team_Batting_H", main = "Histogram of
Hits")
hist(moneyball$TEAM_BATTING_1B, col = "green", xlab = "Team_Batting_1B", main = "Histogram of
Singles")
hist(moneyball$TEAM_BATTING_2B, col = "#09ADAD", xlab = "Doubles", main = "Histogram of Doubles")
```

```
boxplot(moneyball$TEAM_BATTING_H, col = "#A71930", main = "Boxplot of Hits")
boxplot(moneyball$TEAM_BATTING_1B, col = "green", main = "Boxplot of Singles")
boxplot(moneyball$TEAM_BATTING_2B, col = "#09ADAD", main = "Boxplot of Doubles")
par(mfrow=c(1,1))

# Triples and Home Runs
par(mfrow=c(2,2))
hist(moneyball$TEAM_BATTING_3B, col = "#A71930", xlab = "Triples", main = "Histogram of Triples")
hist(moneyball$TEAM_BATTING_HR, col = "#DBCEAC", xlab = "Home Runs", main = "Histogram of Home Runs")
boxplot(moneyball$TEAM_BATTING_3B, col = "#A71930", main = "Boxplot of Triples")
boxplot(moneyball$TEAM_BATTING_HR, col = "#DBCEAC", main = "Boxplot of Home Runs")
par(mfrow=c(1,1))

# Walks, Strikeouts, HBP
par(mfrow=c(2,3))
hist(moneyball$TEAM_BATTING_BB, col = "#A71930", xlab = "Walks", main = "Histogram of Walks")
hist(moneyball$TEAM_BATTING_SO, col = "#09ADAD", xlab = "Strikeouts", main = "Histogram of Strikeouts")
hist(moneyball$TEAM_BATTING_HBP, col = "#DBCEAC", xlab = "Hit By Pitches", main = "Histogram of HBP")
boxplot(moneyball$TEAM_BATTING_BB, col = "#A71930", main = "Boxplot of Walks")
boxplot(moneyball$TEAM_BATTING_SO, col = "#09ADAD", main = "Boxplot of Strikeouts")
boxplot(moneyball$TEAM_BATTING_HBP, col = "#DBCEAC", main = "Boxplot of HBP")
par(mfrow=c(1,1))

# Stolen Bases and Caught Stealing
par(mfrow=c(2,2))
hist(moneyball$TEAM_BASERUN_SB, col = "#A71930", xlab = "Stolen Bases", main = "Histogram of Steals")
hist(moneyball$TEAM_BASERUN_CS, col = "#DBCEAC", xlab = "Caught Stealing", main = "Histogram of CS")
boxplot(moneyball$TEAM_BASERUN_SB, col = "#A71930", main = "Boxplot of Steals")
boxplot(moneyball$TEAM_BASERUN_CS, col = "#DBCEAC", main = "Boxplot of CS")
par(mfrow=c(1,1))

################ Pitching ############
# Hits and Home Runs
par(mfrow=c(2,4))
hist(moneyball$TEAM_PITCHING_H, col = "#A71930", xlab = "Hits Against", main = "Histogram of Hits Against")
hist(moneyball$TEAM_PITCHING_HR, col = "#09ADAD", xlab = "Home Runs Against", main = "Histograms of HR Against")
boxplot(moneyball$TEAM_PITCHING_H, col = "#A71930", main = "Boxplot of Hits Against")
boxplot(moneyball$TEAM_PITCHING_HR, col = "#09ADAD", main = "Boxplot of HR Against")
par(mfrow=c(1,1))

# Walks and Strikeouts
```

```
par(mfrow=c(2,2))
hist(moneyball$TEAM_PITCHING_BB, col = "#A71930", xlab = "Walks Allowed", main = "Histogram of
Walks Allowed")
hist(moneyball$TEAM_PITCHING_SO, col = "#DBCEAC", xlab = "Strikeouts", main = "Histograms of
Strikeouts")
boxplot(moneyball$TEAM_PITCHING_BB, col = "#A71930", main = "Boxplot of Walks Allowed")
boxplot(moneyball$TEAM_PITCHING_SO, col = "#DBCEAC", main = "Boxplot of Strikeouts")
par(mfrow=c(1,1))

############## Fielding ###########
# Double Plays and Errors
par(mfrow=c(2,2))
hist(moneyball$TEAM_FIELDING_DP, col = "#A71930", xlab = "Double Plays", main = "Histogram of
Double Plays")
hist(moneyball$TEAM_FIELDING_E, col = "#09ADAD", xlab = "Errors Committed", main = "Histogram of
Errors Committed")
boxplot(moneyball$TEAM_FIELDING_DP, col = "#A71930", main = "Boxplot of Double Plays")
boxplot(moneyball$TEAM_FIELDING_E, col = "#09ADAD", main = "Boxplot of Errors Committed")
par(mfrow=c(1,1))


######## Scatterplot Matrix ##########

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)
}

# Batting Stats and Wins
pairs(moneyball[2:8], lower.panel=panel.smooth, upper.panel = panel.cor)

#Baserunning  Stats and Wins
pairs(~ moneyball$TARGET_WINS + moneyball$TEAM_BASERUN_CS + moneyball$TEAM_BASERUN_SB,
lower.panel = panel.smooth, upper.panel = panel.cor)

#Pitcher Stats and Wins
pairs(~ moneyball$TARGET_WINS + moneyball$TEAM_PITCHING_BB + moneyball$TEAM_PITCHING_H +
        moneyball$TEAM_PITCHING_HR + moneyball$TEAM_PITCHING_SO, lower.panel = panel.smooth,
upper.panel = panel.cor)

pairs(moneyball[c(2,16,17)], lower.panel = panel.smooth, upper.panel = panel.cor)
```

```
######################### Part 2: Data Preparation #####################

## Using MICE
pMiss <- function(x)
{
  sum(is.na(x)) / length(x) * 100
}

apply(moneyball, 2, pMiss)

md.pattern(moneyball)
aggr(moneyball[c(8,9,10,11,15,17)], numbers = TRUE, sortVars = TRUE, labels =
names(moneyball[c(8,9,10,11,15,17)]), gap = 0)

moneyball_imput <- mice(moneyball, m=5, maxit=50, meth="rf", seed=500)
summary(moneyball_imput)
densityplot(moneyball_imput)
#stripplot(moneyball_imput)
moneyball2 <- complete(moneyball_imput,1)
summary(moneyball2)

# Create flags for missing variables

moneyball2$TEAM_BATTING_SO_M <- ifelse(is.na(moneyball$TEAM_BATTING_SO), 1, 0)
moneyball2$TEAM_BASERUN_SB_M <- ifelse(is.na(moneyball$TEAM_BASERUN_SB), 1, 0)
moneyball2$TEAM_BASERUN_CS_M <- ifelse(is.na(moneyball$TEAM_BASERUN_CS), 1, 0)
moneyball2$TEAM_BATTING_HBP_M <- ifelse(is.na(moneyball$TEAM_BATTING_HBP), 1, 0)
moneyball2$TEAM_PITCHING_SO_M <- ifelse(is.na(moneyball$TEAM_PITCHING_SO), 1, 0)
moneyball2$TEAM_FIELDING_DP_M <- ifelse(is.na(moneyball$TEAM_FIELDING_DP), 1, 0)

#Straighten Relationships
#### determine the appropriate transformation
par(mfrow = c(2,3))
hist(moneyball2$TEAM_BATTING_1B, col = 'red', main = 'Singles')
hist(log(moneyball2$TEAM_BATTING_1B), col = 'blue', main = 'log(Singles)')
hist(sqrt(moneyball2$TEAM_BATTING_1B), col = 'green', main = 'sqrt(Singles)')

hist(moneyball2$TEAM_BATTING_3B, col = 'red', main = 'Triples')
hist(log(moneyball2$TEAM_BATTING_3B), col = 'blue', main = 'log(Triples)')
hist(sqrt(moneyball2$TEAM_BATTING_3B), col = 'green', main = 'sqrt(Triples)')

hist(moneyball2$TEAM_BASERUN_SB, col = 'red', main = 'Stolen Bases')
hist(log(moneyball2$TEAM_BASERUN_SB), col = 'blue', main = 'log(Stolen Bases)')
hist(sqrt(moneyball2$TEAM_BASERUN_SB), col = 'green', main = 'sqrt(Stolen Bases)')

hist(moneyball2$TEAM_BASERUN_CS, col = 'red', main = 'Caught Stealing')
hist(log(moneyball2$TEAM_BASERUN_CS), col = 'blue', main = 'log(Caught Stealing)')
hist(sqrt(moneyball2$TEAM_BASERUN_CS), col = 'green', main = 'sqrt(Caught Stealing)')
```

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

round(skewness(moneyball2$TEAM_BATTING_1B), 2)
round(skewness(log(moneyball2$TEAM_BATTING_1B)), 2)
round(skewness(sqrt(moneyball2$TEAM_BATTING_1B)), 2)

round(skewness(moneyball2$TEAM_BATTING_3B), 2)
moneyball2$TEAM_BATTING_3B[which(moneyball2$TEAM_BATTING_3B < 1)] <- 1
round(skewness(log(moneyball2$TEAM_BATTING_3B)), 2)
round(skewness(sqrt(moneyball2$TEAM_BATTING_3B)), 2)

round(skewness(moneyball2$TEAM_BASERUN_SB), 2)
moneyball2$TEAM_BASERUN_SB[which(moneyball2$TEAM_BASERUN_SB < 1)] <- 1
round(skewness(log(moneyball2$TEAM_BASERUN_SB)), 2)
round(skewness(sqrt(moneyball2$TEAM_BASERUN_SB)), 2)

round(skewness(moneyball2$TEAM_BASERUN_CS), 2)
moneyball2$TEAM_BASERUN_CS[which(moneyball2$TEAM_BASERUN_CS < 1)] <- 1
round(skewness(log(moneyball2$TEAM_BASERUN_CS)), 2)
round(skewness(sqrt(moneyball2$TEAM_BASERUN_CS)), 2)

## transform the four highly skewed variables using the log transform
moneyball2$TEAM_BATTING_1B <- log(moneyball2$TEAM_BATTING_1B)
moneyball2$TEAM_BATTING_3B <- log(moneyball2$TEAM_BATTING_3B)
moneyball2$TEAM_BASERUN_SB <- log(moneyball2$TEAM_BASERUN_SB)
moneyball2$TEAM_BASERUN_CS <- log(moneyball2$TEAM_BASERUN_CS)

### Address outliers with trimming
par(mfrow = c(1,3))
H_QT95 <- quantile(moneyball2$TEAM_PITCHING_H, probs = 0.95)
moneyball2$TEAM_PITCHING_H[which(moneyball2$TEAM_PITCHING_H > H_QT95)] <- H_QT95
hist(moneyball2$TEAM_PITCHING_H, col = 'Red', main = 'Original')
hist(moneyball2$TEAM_PITCHING_H, col = 'Green', main = '99th Quantile')
hist(moneyball2$TEAM_PITCHING_H, col = 'Blue', main = '95th Quantile')

BB_QT95 <- quantile(moneyball2$TEAM_PITCHING_BB, probs = 0.95)
moneyball2$TEAM_PITCHING_BB[which(moneyball2$TEAM_PITCHING_BB > BB_QT95)] <- BB_QT95
hist(moneyball2$TEAM_PITCHING_BB, col = 'Red', main = 'Original')
hist(moneyball2$TEAM_PITCHING_BB, col = 'Green', main = '99th Quantile')
hist(moneyball2$TEAM_PITCHING_BB, col = 'Blue', main = '95th Quantile')

SO_QT99 <- quantile(moneyball2$TEAM_PITCHING_SO, probs = 0.99)
moneyball2$TEAM_PITCHING_SO[which(moneyball2$TEAM_PITCHING_SO > SO_QT99)] <- SO_QT99
hist(moneyball2$TEAM_PITCHING_SO, col = 'Red', main = 'Original')
hist(moneyball2$TEAM_PITCHING_SO, col = 'Green', main = '99th Quantile')
hist(moneyball2$TEAM_PITCHING_SO, col = 'Blue', main = '95th Quantile')

ER_QT95 <- quantile(moneyball2$TEAM_FIELDING_E, probs = 0.95)
```

```
moneyball2$TEAM_FIELDING_E[which(moneyball2$TEAM_FIELDING_E > ER_QT95)] <- ER_QT95
hist(moneyball2$TEAM_FIELDING_E, col = 'Red', main = 'Original')
hist(moneyball2$TEAM_FIELDING_E, col = 'Green', main = '99th Quantile')
hist(moneyball2$TEAM_FIELDING_E, col = 'Blue', main = '95th Quantile')
par(mfrow = c(1,1))

#Check that na's are gone.
summary(moneyball2)

#***Note at this point you may wish to also check to ensure outliers are imputed but not deleted***

#################### Part 3: Model Creation #################################################

#Function for Mean Square Error Calculation
mse <- function(sm)
  mean((sm$residuals)^2)

# Stepwise Approach

stepwisemodel <- lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_2B +
                    TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
                    TEAM_BATTING_HBP + TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_PITCHING_H +
                    TEAM_PITCHING_BB + TEAM_PITCHING_SO + TEAM_FIELDING_E +
                    TEAM_FIELDING_DP + TEAM_BATTING_SO_M + TEAM_BASERUN_SB_M +
TEAM_BASERUN_CS_M +
                    TEAM_BATTING_HBP_M + TEAM_PITCHING_SO_M + TEAM_FIELDING_DP_M,
                   data = moneyball2)
stepwise <- stepAIC(stepwisemodel, direction = "both")
summary(stepwise)
vif(stepwise)
sqrt(vif(stepwise)) > 2

# All subsets regression
subsets <- regsubsets(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_2B +
                      TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
                      TEAM_BATTING_HBP + TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_PITCHING_H +
                      TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_PITCHING_SO + TEAM_FIELDING_E +
                      TEAM_FIELDING_DP + TEAM_BATTING_SO_M + TEAM_BASERUN_SB_M +
TEAM_BASERUN_CS_M +
                      TEAM_BATTING_HBP_M + TEAM_PITCHING_SO_M + TEAM_FIELDING_DP_M,
                     data = moneyball2, nbest = 2)
plot(subsets, scale="adjr2", cex = 0.4)

subset <- lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B +
             TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_HR +
             TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP +
             TEAM_BASERUN_SB_M, data = moneyball2)
summary(subset)
```

```
vif(subset)

# Random Forest Regression
set.seed(1234)
model.rf <- randomForest(TARGET_WINS ~ ., moneyball2[-c(1,19:24)], ntree=500, importance = TRUE)
model.rf
model.rf$importance
plot(model.rf)
varImpPlot(model.rf)

model.rf1 <- randomForest(TARGET_WINS ~ ., moneyball[c(-1)], ntree=500, importance = TRUE,
na.action = na.roughfix)
model.rf1
varImpPlot(model.rf1)

######## Performance #######
summary(stepwise)$adj.r.squared
summary(subset)$adj.r.squared
model.rf$rsq[500]

mse(stepwise)
mse(subset)
model.rf$mse[500]

#####
#Designated proper working environment on my computer. You will want to make sure it is in proper
place for your computer.
#####

#################### Test Data ##########################
moneyball_test=read.csv("moneyball_test.csv",header=T)
moneyball_test$TEAM_BATTING_1B <- moneyball_test$TEAM_BATTING_H -
  moneyball_test$TEAM_BATTING_HR - moneyball_test$TEAM_BATTING_3B -
  moneyball_test$TEAM_BATTING_2B

# Clean the Data
moneyball_imput_test <- mice(moneyball_test, m=5, maxit=50, meth="rf", seed=500)
densityplot(moneyball_imput_test)
moneyball_T <- complete(moneyball_imput_test,4)

#moneyball_T$TEAM_BATTING_SO_M <- ifelse(is.na(moneyball_test$TEAM_BATTING_SO), 1, 0)
#moneyball_T$TEAM_BASERUN_SB_M <- ifelse(is.na(moneyball_test$TEAM_BASERUN_SB), 1, 0)
#moneyball_T$TEAM_BASERUN_CS_M <- ifelse(is.na(moneyball_test$TEAM_BASERUN_CS), 1, 0)
#moneyball_T$TEAM_BATTING_HBP_M <- ifelse(is.na(moneyball_test$TEAM_BATTING_HBP), 1, 0)
#moneyball_T$TEAM_PITCHING_SO_M <- ifelse(is.na(moneyball_test$TEAM_PITCHING_SO), 1, 0)
#moneyball_T$TEAM_FIELDING_DP_M <- ifelse(is.na(moneyball_test$TEAM_FIELDING_DP), 1, 0)

moneyball_T$TEAM_BATTING_3B[which(moneyball_T$TEAM_BATTING_3B < 1)] <- 1
```

```
moneyball_T$TEAM_BASERUN_SB[which(moneyball_T$TEAM_BASERUN_SB < 1)] <- 1
moneyball_T$TEAM_BASERUN_CS[which(moneyball_T$TEAM_BASERUN_CS < 1)] <- 1
moneyball_T$TEAM_BATTING_1B <- log(moneyball_T$TEAM_BATTING_1B)
moneyball_T$TEAM_BATTING_3B <- log(moneyball_T$TEAM_BATTING_3B)
moneyball_T$TEAM_BASERUN_SB <- log(moneyball_T$TEAM_BASERUN_SB)
moneyball_T$TEAM_BASERUN_CS <- log(moneyball_T$TEAM_BASERUN_CS)

H_QT95 <- quantile(moneyball_T$TEAM_PITCHING_H, probs = 0.95)
moneyball_T$TEAM_PITCHING_H[which(moneyball_T$TEAM_PITCHING_H > H_QT95)] <- H_QT95

BB_QT95 <- quantile(moneyball_T$TEAM_PITCHING_BB, probs = 0.95)
moneyball_T$TEAM_PITCHING_BB[which(moneyball_T$TEAM_PITCHING_BB > BB_QT95)] <- BB_QT95

SO_QT99 <- quantile(moneyball_T$TEAM_PITCHING_SO, probs = 0.99)
moneyball_T$TEAM_PITCHING_SO[which(moneyball_T$TEAM_PITCHING_SO > SO_QT99)] <- SO_QT99

ER_QT95 <- quantile(moneyball_T$TEAM_FIELDING_E, probs = 0.95)
moneyball_T$TEAM_FIELDING_E[which(moneyball_T$TEAM_FIELDING_E > ER_QT95)] <- ER_QT95

# Stand Alone Scoring
moneyball_T$P_TARGET_WINS <- predict(model.rf, moneyball_T)

#subset of data set for the deliverable "Scored data file"
prediction <- moneyball_T[c("INDEX","P_TARGET_WINS")]

#####
#Note, this next function will output an Excel file in your work environment called write.xlsx.
#####

#Prediction File
write.csv(prediction, file = "moneyball_NAME.csv")


#########  BONUS

bonus1 <- lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B +
            TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_HR +
            TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP +
            TEAM_BASERUN_SB_M, data = moneyball2)
summary(bonus1)

bonus2 <- glm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B +
            TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_HR +
            TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP +
            TEAM_BASERUN_SB_M, data = moneyball2)
summary(bonus2)
AIC(bonus1, bonus2)
```