Moneyball OLS Regression Project

Introduction

Building upon the advantages of RStudio capabilities, the report addresses the common business problem in sport management of predicting the number of wins (TARGET_WINS) for a professional baseball team during a regular season. To accomplish the objective, the study evaluates the behavior of relevant performance predictors in multiple linear regression modeling, specifically the Ordinary Least Square method (OLS). Hence, the creation of regression models relies on a given Moneyball training data set containing records of baseball teams and variables reflecting their performance statistics for the time period 1871 to 2006 inclusive and using offensive and defensive strategies as action of preventing an opponent team from scoring.

A systematic analysis starts by defining the business objective from the information revealed on the training data set with exploratory data analysis (EDA) techniques. Following EDA, data preparation and processing for modeling detects and imputes missing values, and also truncates, transforms, and creates variables to fit more closely the underlying assumptions of statistical tests in multiple linear regressions, including to improve normality of the distribution and adjust for outliers.

With a clean Moneyball training data set producing an adequate in-sample of the population, a modeling framework constructed from relevant variables and automated variable selection (AVS) methods, continues with cross-validation comparison techniques to determine the best method based on statistical model performance criteria and simplicity in the number of variables. Finally, with a selected model and an out-of-sample Moneyball test data set cleaned with the same processing applied to the training data, the linear equation from the selected best method predicts the performance of the team for the next season and the scoring program solution.

The OLS regression application process consist of:

Section 1: Data Exploration

Section 2: Data Preparation

Section 3: Model Building

Section 4: Model Selection

Section 5: Model Deployment Code

Section 1: Data Exploration

To maximize insights from the baseball teams' records on performance variables, EDA subsections aim to reveal the underlying structure and properties of the Moneyball training data frame using descriptive statistics and graphical analysis to clarify the completeness, quality, and adequacy of different types of measurements.

Data Collection and Types of Data Measurement Scales: Collected from an observational setting, the Moneyball training data frame structure represents 2,276 observations (rows), each record describing a professional baseball team' performance in a given year from 1871-2006, wide time frame range hinting inconsistencies in the collection of data over the years. The type of measurements for all variables (columns) in the data set consist of quantitative or numerical integer values with statistics adjusted to match the performance of a 162-game season. The columns consist of:

- 1 observation identifier, INDEX.
- 1 continuous response variable, TARGET_WINS.
- 15 predictors or descriptive explanatory variables.

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

Response and Explanatory Variables: The examination of the Moneyball training data set reveals a continuous response variable, TARGET_WINS used in the regression model building process to establish whether there is functional relationship linked to one or several of the 15 potential predictor variables available in the training data set. The predictors, all quantitative measures, define the performance of a baseball team based on offensive and defensive strategies further subdivided on:

- 2 offensive strategies, Batting with 7 variables and Baserun with 2 variables.
- 2 defensive strategies, Fielding with 2 variables and Pitching with 4 variables.

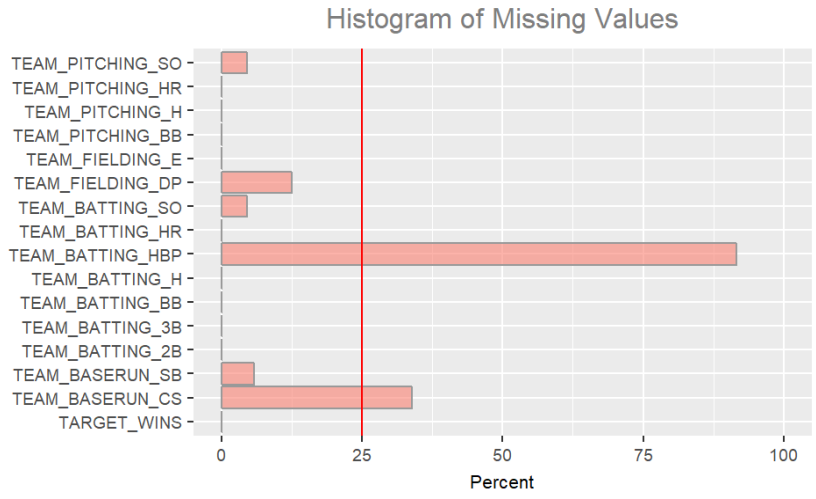
In addition, a color-coded data dictionary presents the name of the variables, definitions, individual theoretical effect on a game (positive or negative impact) for later evaluation of the accuracy in the correlations with predictors and the beta coefficients signs in the results from regression modeling equations, and finally the type of game strategies (offensive or defensive).

Data Dictionary

VARIABLE.NAME	DEFINITION	THEORETICAL.EFFECT	STRATEGY
INDEX	Identification Variable (do not use)	None	NA
TARGET_WINS	NA	NA	NA
TEAM_BATTING_H	**Base Hits by batters (1B,2B,3B,HR)**	**Positive Impact on Wins**	**Offensive strategy**
TEAM_BATTING_2B	**Doubles by batters (2B)**	**Positive Impact on Wins**	**Offensive strategy**
TEAM_BATTING_3B	**Triples by batters (3B)**	**Positive Impact on Wins**	**Offensive strategy**
TEAM_BATTING_HR	**Homeruns by batters (4B)**	**Positive Impact on Wins**	**Offensive strategy**
TEAM_BATTING_BB	**Walks by batters**	**Positive Impact on Wins**	**Offensive strategy**
TEAM_BATTING_HBP	**Batters hit by pitch (get a free base)**	**Positive Impact on Wins**	**Offensive strategy**
TEAM_BATTING_SO	**Strikeouts by batters**	**Negative Impact on Wins**	**Offensive strategy**
TEAM_BASERUN_SB	**Stolen bases**	**Positive Impact on Wins**	**Offensive strategy**
TEAM_BASERUN_CS	**Caught stealing**	**Negative Impact on Wins**	**Offensive strategy**
TEAM_FIELDING_E	**Errors**	**Negative Impact on Wins**	**Defensive strategy**
TEAM_FIELDING_DP	**Double Plays**	**Positive Impact on Wins**	**Defensive strategy**
TEAM_PITCHING_BB	**Walks allowed**	**Negative Impact on Wins**	**Defensive strategy**
TEAM_PITCHING_H	**Hits allowed**	**Negative Impact on Wins**	**Defensive strategy**
TEAM_PITCHING_HR	**Homeruns allowed**	**Negative Impact on Wins**	**Defensive strategy**
TEAM_PITCHING_SO	**Strikeouts by pitchers**	**Positive Impact on Wins**	**Defensive strategy**

Missing Values: A total of 6 variables in the training data set expose 3,478 missing values (NAs) or incomplete records in the following proportions by group of activities (Batting, Baserun, Fielding, and Pitching) and by specific variables:

- 63% in 2 Batting variables: TEAM_BATTING_HBP (2085 values ~ 92%) and TEAM_BATTING_SO (102 values ~ 4%)
- 26% in 2 Baserun variables: TEAM_BASERUN_CS (772 values ~ 34%) and TEAM_BASERUN_SB (131 values ~ 6%)
- 8% in 1 Fielding variable: TEAM_FIELDING_DP (286 values ~ 13%)
- 3% in 1 Pitching variable: TEAM_PITCHING_SO (102 values ~ 4%)



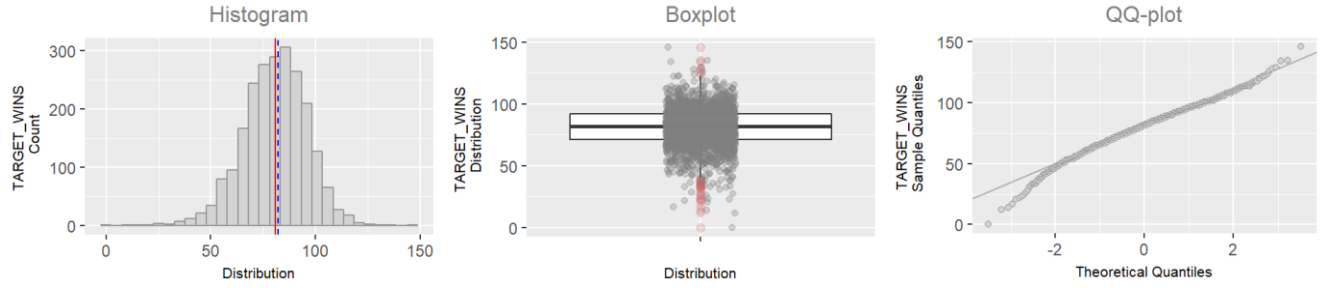
An important consideration before performing multiple regression analysis relates to properly handling those missing values in the predictors allowing to draw accurate inference about the data, and therefore, pre-processing by imputation methods aims to replace NAs with estimated values prior to modeling. Noteworthy, two variables display significant percentage of missing values, TEAM_BATTING_HBP (92%) and TEAM_BASERUN_CS (34%) imposing a counterproductive biased effect if used in the modeling process. At this stage, insufficient evidence impedes dropping any other variable with missing values.

Descriptive and Graphic Statistics: Inspecting the data quality and integrity of the Moneyball data includes analysis of measures of central tendency: minimum and maximum values, 1st and 3rd quartiles (25th and 75th quartiles respectively), mean, median, sum of all values, lower value (LCL) and upper value (UCL) interval limits estimate for the mean based on a t-distribution, variance, standard deviation (Stdev), skewness, and kurtosis. These summary statistics accompanied by statistical graphical techniques assess alignment to common data assumptions, including normality of the distribution, linearity in the correlation between the dependent and independent variables, and detect possible anomalies such as missing data and the presence of outliers.

- ***On the response variable TARGET_WINS,*** the univariate summary statistics analysis indicates data clear of NAs without negative numbers, and a range from zero or no wins to an unlikely higher maximum number of wins of 146, considering normal between 30 to 120 wins and a perfect 162-0 if the team wins all games in a regular season. The histogram shows a fairly normal distribution of the response variable data showing the mean draw with a red line at 80.79 and the median with a dashed blue line at 82 team wins close to each other. The combination of three graphic statistics evaluates the data by using a histogram plotting an approximately normal distribution slightly left skewed (-0.4), a boxplot rendering the bulk of values against the median and quartiles, and a QQ-plot denoting two sets of quantiles against one another inferring about the underlying broad linearity of the distribution with few outliers. From the histogram and QQ-plot results, insignificant skewness value does not call for transformation.

Descriptive and Graphic Statistics of TARGET_WINS

	nobs	NAs	Minimum	Maximum	Quartile	Quartile	Mean	Median	Sum	SE Mean	LCL Mean	UCL Mean	Variance	Stdev	Skewness	Kurtosis
TARGET_WINS	2276	0	0	146	71	92	80.79	82	183880	0.33	80.14	81.44	248.13	15.75	-0.4	1.03

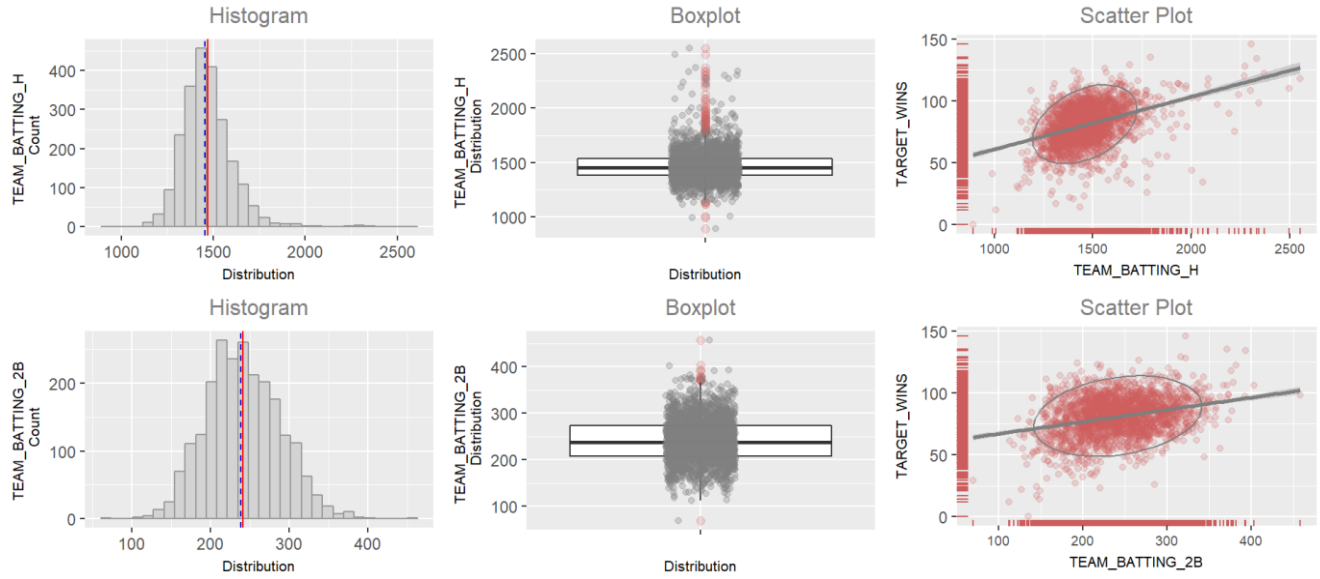


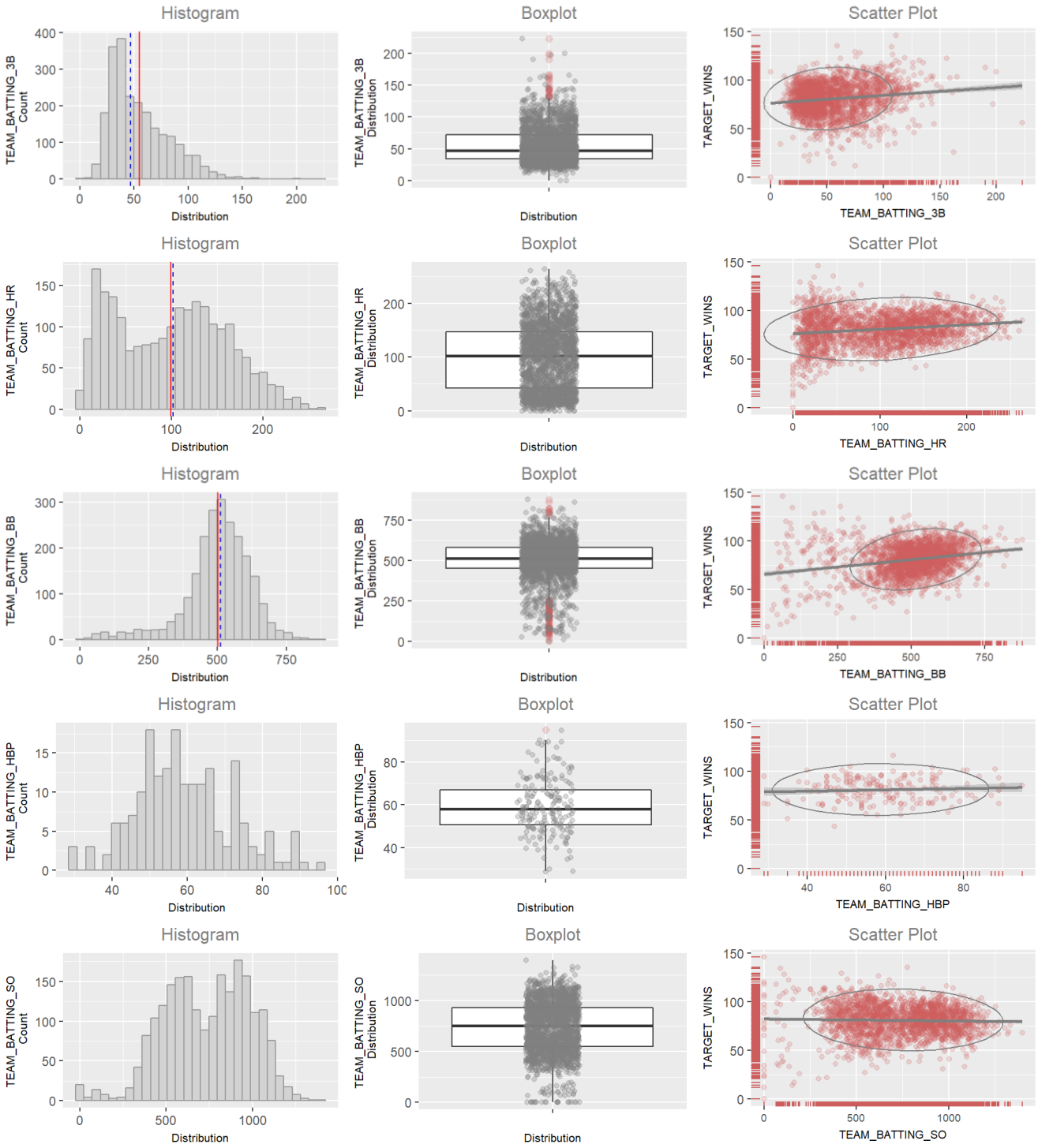
- ***On the descriptive and graphic statistics for the seven Batting variables (offensive strategy),*** the figures show no negative numbers, however with minimum values of zeros, an unlikely scenario in reality. Regarding missing values, the high number in TEAM_BATTING_HBP, makes this variable a candidate to be eliminated from the linear regression modeling, while the lower number in TEAM_BATTING_SO enforces imputation.

  The statistical graphs evaluate the data by each Batting variable per row using histograms, boxplots, and scatterplots for bivariate data analysis of the correlation with the response variable TARGET_WINS. The variable TEAM_BATTING_2B shows a broadly normal bell-shaped curve distribution, while TEAM_BATTING_HR records two abnormal bell-shape curves in the distribution. Also, TEAM_BATTING_H and TEAM_BATTING_3B confirm positive skewed distribution with many outliers, and TEAM_BATTING_BB with a negative skewed distribution. Visually, the majority of batting variables indicate a positive linear relationship with TARGET_WINS from the scatterplots.

Descriptive and Graphic Statistics of Batting Variables

	nobs	NAs	Minimum	Maximum	Quartile	Quartile	Mean	Median	Sum	SE Mean	LCL Mean	UCL Mean	Variance	Stdev	Skewness	Kurtosis
TEAM_BATTING_H	2276	0	891	2554	1383.0	1537.25	1469.27	1454	3344058	3.03	1463.33	1475.21	20906.61	144.59	1.57	7.28
TEAM_BATTING_2B	2276	0	69	458	208.0	273.00	241.25	238	549078	0.98	239.32	243.17	2190.37	46.80	0.22	0.01
TEAM_BATTING_3B	2276	0	0	223	34.0	72.00	55.25	47	125749	0.59	54.10	56.40	780.56	27.94	1.11	1.50
TEAM_BATTING_HR	2276	0	0	264	42.0	147.00	99.61	102	226717	1.27	97.12	102.10	3665.92	60.55	0.19	-0.96
TEAM_BATTING_BB	2276	0	0	878	451.0	580.00	501.56	512	1141548	2.57	496.52	506.60	15048.14	122.67	-1.03	2.18
TEAM_BATTING_HBP	2276	2085	29	95	50.5	67.00	59.36	58	11337	0.94	57.51	61.21	168.15	12.97	0.32	-0.11
TEAM_BATTING_SO	2276	102	0	1399	548.0	930.00	735.61	750	1599206	5.33	725.15	746.06	61765.38	248.53	-0.30	-0.32

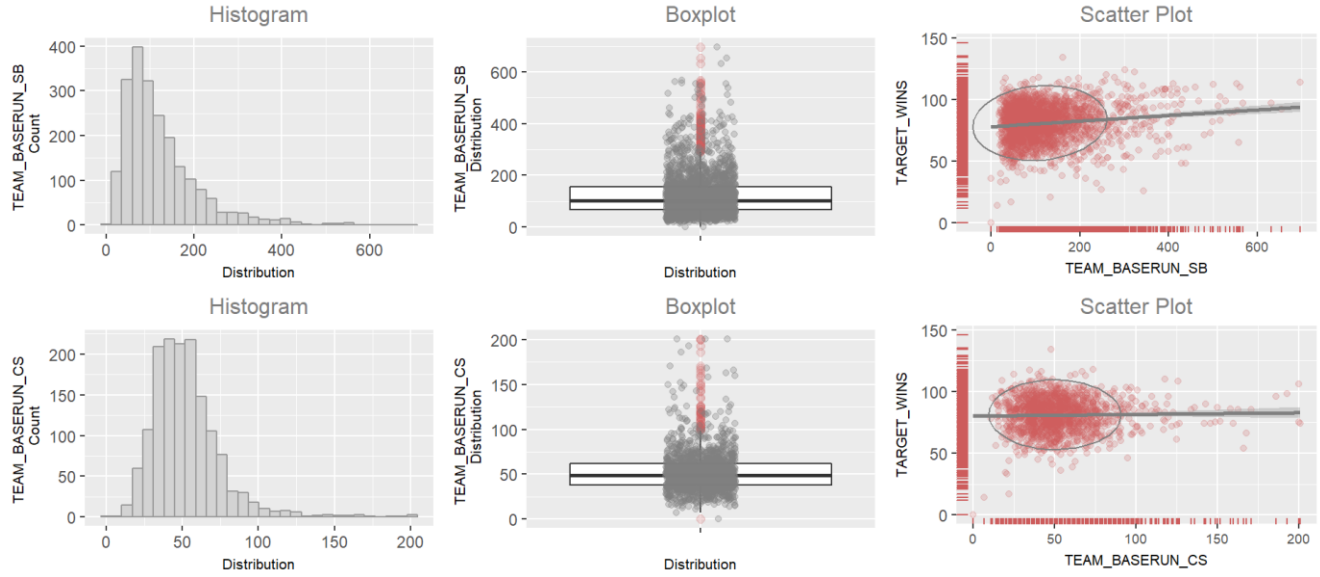




- ***On the descriptive and graphic statistics for the two Baserun variables (offensive strategy),*** main concerns relate to the unlikely scenario of minimum values of zeros, fixing missing data for both variables, and determining if common right skewness remains to be address after imputation pre-process and before modeling. On the other hand, the relationship between both variables separately with TARGET_WINS seem constant up to certain number of values. The number of missing values in TEAM_BASERUN_CS is fairly large for imputation methods to succeed; therefore, variable deletion will be considered.

Descriptive and Graphic Statistics of Baserun Variables

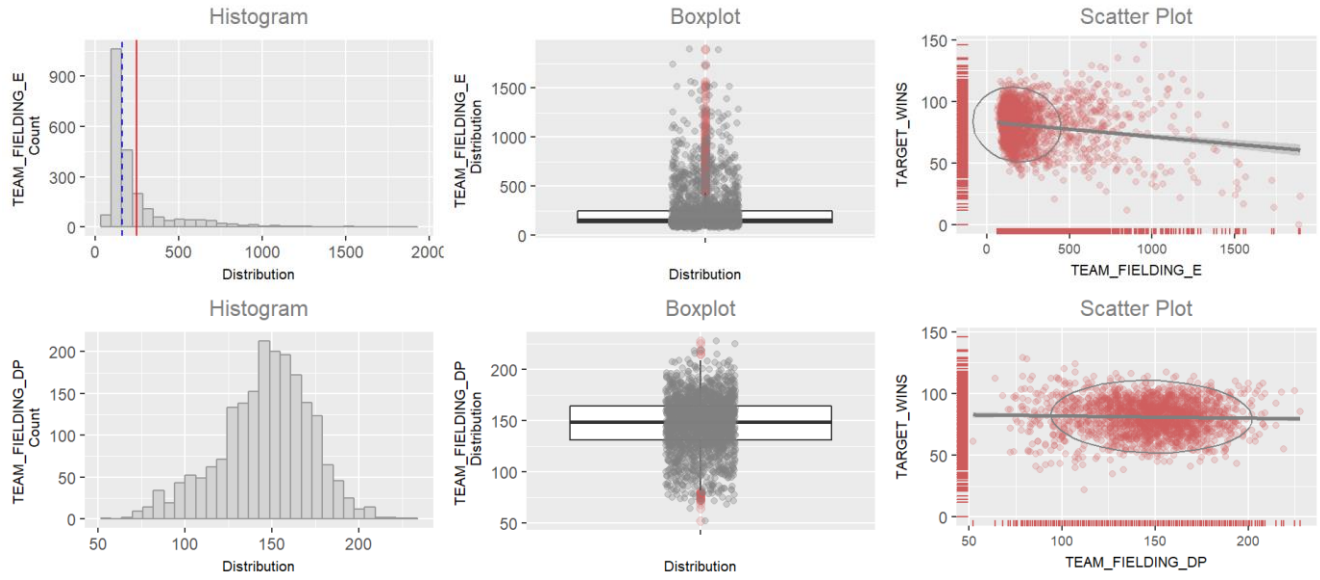
	nobs	NAs	Minimum	Maximum	Quartile	Quartile	Mean	Median	Sum	SE Mean	LCL Mean	UCL Mean	Variance	Stdev	Skewness	Kurtosis
TEAM_BASERUN_SB	2276	131	0	697	66	156	124.76	101	267614	1.90	121.04	128.48	7707.29	87.79	1.97	5.49
TEAM_BASERUN_CS	2276	772	0	201	38	62	52.80	49	79417	0.59	51.64	53.96	526.99	22.96	1.98	7.62



- ***On the descriptive and graphic statistics for the two Fielding variables (defensive strategy),*** the variable TEAM_FIELDING_E presents a marked right skewness for unrealistic values above 500 errors in the upper quartile of the boxplot, whereas the variable for double plays, TEAM_FIELDING_DP shows left skewness and missing values that can distort the interpretation of the model if used.

Descriptive and Graphic Statistics of Fielding Variables

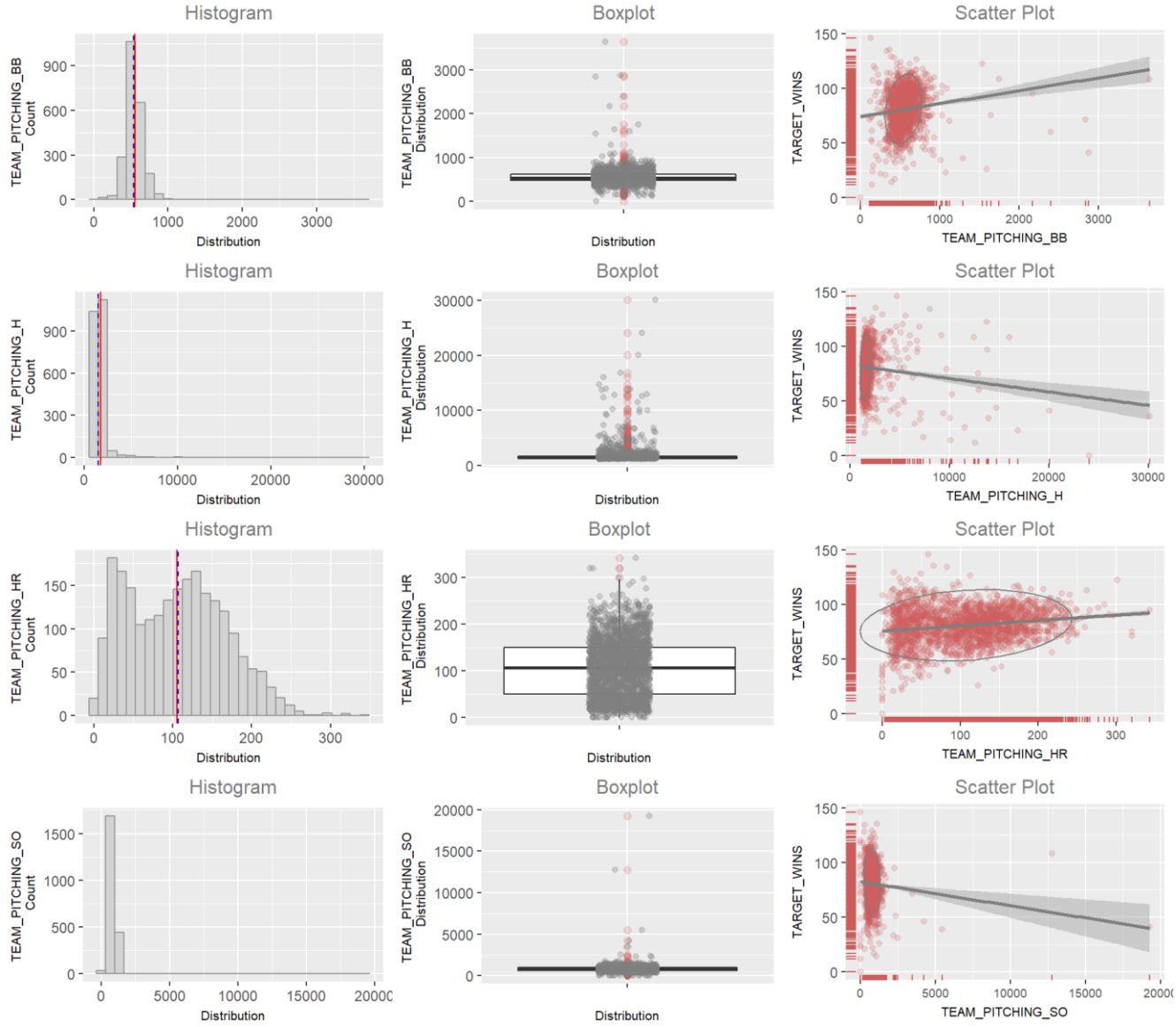
	nobs	NAs	Minimum	Maximum	Quartile	Quartile	Mean	Median	Sum	SE Mean	LCL Mean	UCL Mean	Variance	Stdev	Skewness	Kurtosis
TEAM_FIELDING_E	2276	0	65	1898	127	249.25	246.48	159	560990	4.77	237.12	255.84	51879.62	227.77	2.99	10.97
TEAM_FIELDING_DP	2276	286	52	228	131	164.00	146.39	149	291312	0.59	145.23	147.54	687.82	26.23	-0.39	0.18



- ***On the descriptive and graphic statistics of the four Pitching variables (defensive strategy),*** three of them, namely TEAM_PITCHING_BB, TEAM_PITCHING_H, and TEAM_PITCHING_SO stand out for the unrealistic maximum values and pronounced right skewed distribution plenty of influential points, while TEAM_PITCHING_HR presents two distinctive bell-shaped curves distributions. Atypical values of zero conform some data point of TEAM_PITCHING_BB, TEAM_PITCHING_HR, and TEAM_PITCHING_SO.

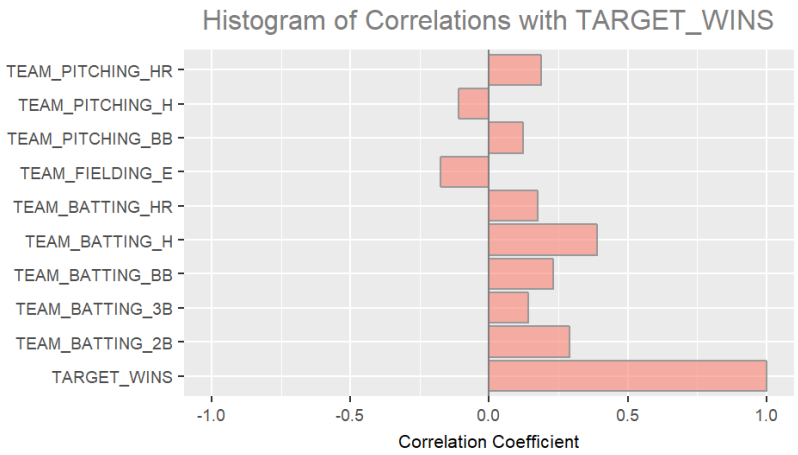
Descriptive and Graphic Statistics of Pitching Variables

	nobs	NAs	Minimum	Maximum	Quartile	Quartile	Mean	Median	Sum	SE Mean	LCL Mean	UCL Mean	Variance	Stdev	Skewness	Kurtosis
TEAM_PITCHING_BB	2276	0	0	3645	476	611.0	553.01	536.5	1258646	3.49	546.17	559.85	27674.77	166.36	6.74	96.97
TEAM_PITCHING_H	2276	0	1137	30132	1419	1682.5	1779.21	1518.0	4049483	29.49	1721.38	1837.04	1979207.03	1406.84	10.33	141.84
TEAM_PITCHING_HR	2276	0	0	343	50	150.0	105.70	107.0	240570	1.28	103.18	108.22	3757.54	61.30	0.29	-0.60
TEAM_PITCHING_SO	2276	102	0	19278	615	968.0	817.73	813.5	1777746	11.86	794.47	840.99	305903.05	553.09	22.17	671.19



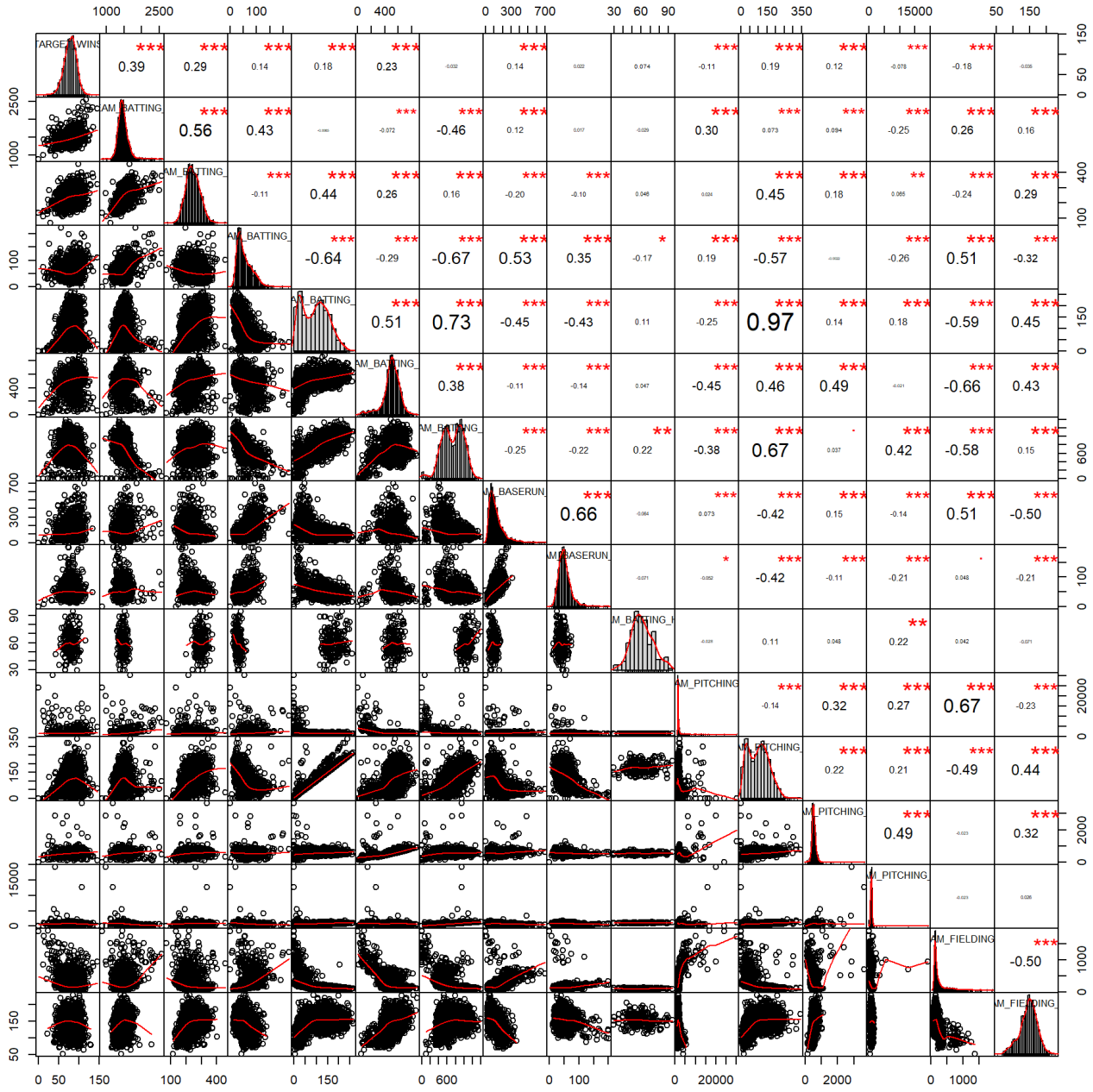
Correlation with the Predicted Variable and Multicollinearity: The following graphs aim to determine the strength and direction of explanatory variables in the Moneyball data linearly correlated with the target variable, but also the possible correlations between predictors.

Visually, the strongest positive correlation coefficient in the histogram points at base hits by batters in TEAM_BATTING_H, as expected, since hits should increase the likelihood of team wins. Also, errors in TEAM_FIELDING_E decreasing the chance of wins verifies the impact as the highest negative correlation coefficient.



Overall, the magnitude of correlations in absolute values with TARGET_WINS varies from medium to low, meaning not highly correlated. The high coefficient refers to TEAM_BATTING_H (0.39) not significant enough for consideration on simple linear regression. However, noticeable inconsistencies surface in the pair-wise scatterplots related to the actual value, positive or negative after comparing the expected impact of predictors in TARGET_WINS. The correlation coefficients of the relationships between the target variable and the predictors TEAM_BASERUN_CS (0.02), TEAM_PITCHING_HR (0.19), and TEAM_PITCHING_BB (0.12) contradict their expected negative impact on team wins as previously explained in the Data Dictionary, while TEAM_PITCHING_SO (-0.08) and TEAM_FIELDING_DP (-0.03) conflict with the intuitive positive impact on wins. Variables with inconsistences present interpretability concerns, if used alone in regression modeling.

Pair-wise Scatterplot of Correlations between Moneyball Variables

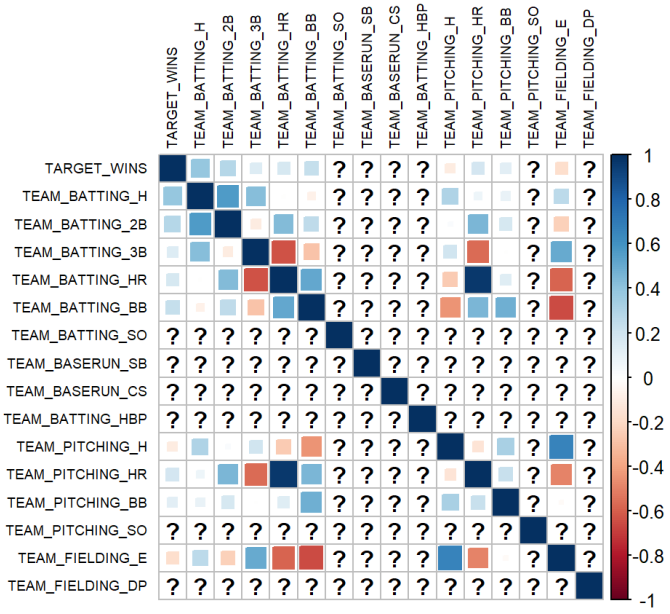


While the scatterplot/histogram matrix includes coefficient estimates for variables with missing values, a re-run deems appropriate after cleaning for non-data with an enhanced imputation method. Meanwhile, a mosaic heatmap matrix below aids on multicollinearity assessment from all pair-wise combinations displaying the behavior of predictor variables in the data frame. The heatmap reveals non-significant correlations with TARGET_WINS, however many correlations among explanatory variables with noticeable high positive (color blue) and high negative (color red) linear relationships as follows:

- Among Batting variables.
- Mainly negative correlations between Batting variables and homeruns allowed (TEAM_PITCHING_HR) and walks allowed (TEAM_PITCHING_BB), whereas primarily positive with errors (TEAM_FIELDING_E).
- Between errors (TEAM_FIELDING_E) and pitching variables with negative impact on hits allowed (TEAM_PITCHING_H) and positive with homeruns allowed (TEAM_PITCHING_HR).

With the presence of multicollinearity, further exploration includes assessing the severity with the variance inflation factor (VIF) and eliminating the influence of the issue during regression modeling.

Heatmap Matrix for Moneyball Variables

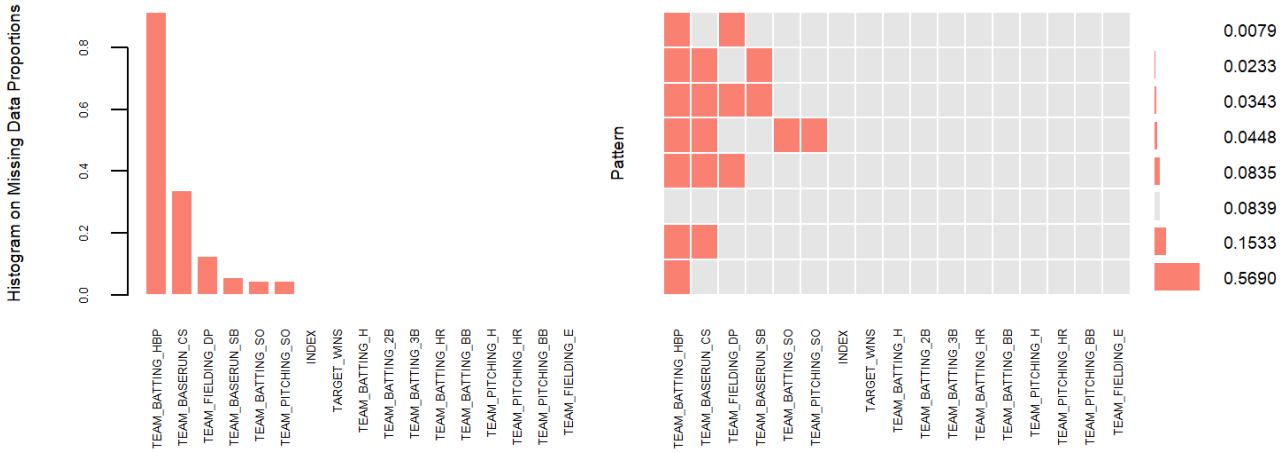


Section 2: Data Preparation

After initial diagnostic background in Section 1, issues with missing-data, unrealistic lower and higher values, and atypical data distribution patterns in certain variables arose as serious constraints for a complete-case analysis to define an appropriate sample for multiple regression analysis. The following set of variable transformations, but also additions, and eliminations of variables treat the negative impact of outliers for a data reasonably normally distributed.

BONUS* - *Missing-data Imputation with Random Forest/Decision Trees: Regarding completeness of the Moneyball training data, as exposed in the heatmap, over 91% of the observations represent complete data (35,214 values in gray color), while less than 9% depict missing data (3,478 values in red color).

Plot of Patterns of Missing Values

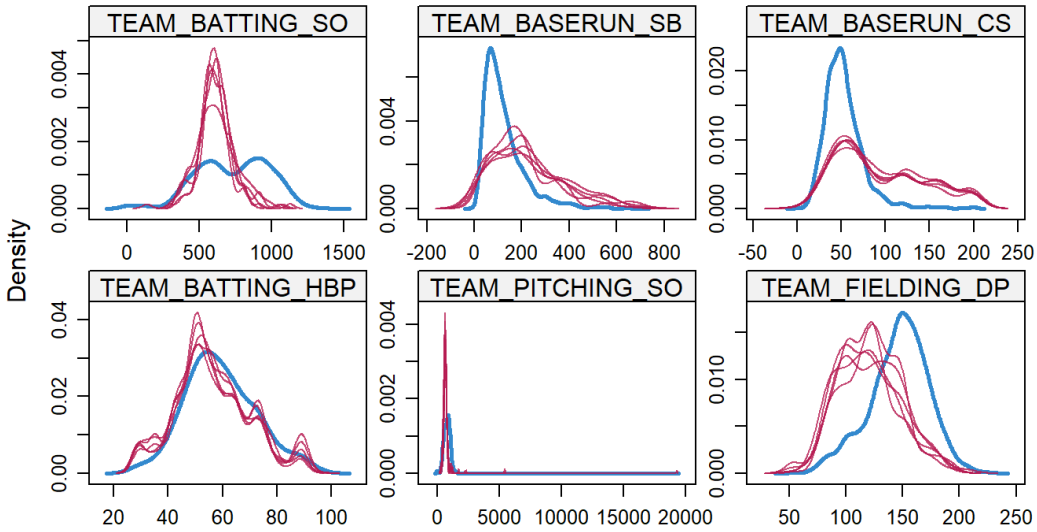


The six variables displaying missing data require pre-processing by fixing NAs with imputation methods before proceeding with transformation that will later minimize the effect of unrealistic values and outliers. The selection of the R-package "mice" (Generates Multivariate Imputations by Chained Equations) for multiple imputation relies in its logarithm power and flexibility to use information from other variables in the data set to predict and impute the missing values. In order to perform imputations correctly, the comparison of two methods built in the "mice" package allows for selection based on their results:

- Method "rf" for Random Forest imputations enhances predictive accuracy by generating a significant number of defaulted bootstrapped trees.
- Method "cart" for regression trees works under the assumption that the missingness is random, unlikely in the case of the Moneyball data set.

The selection-criteria of the Random Forest method "rf" leans on the stability showcased from the more normal distribution after addressing missing values (in color red) compared to the density-curve with missing values (in color blue). The graphs show 5 multiple imputations (m=5) and 5 iterations for each imputation (maxit=5) for all six variables. In addition, the non-predictive value of the variable INDEX, as only an identifier enforces its removal.

Imputation of the Estimated Density and Comparison

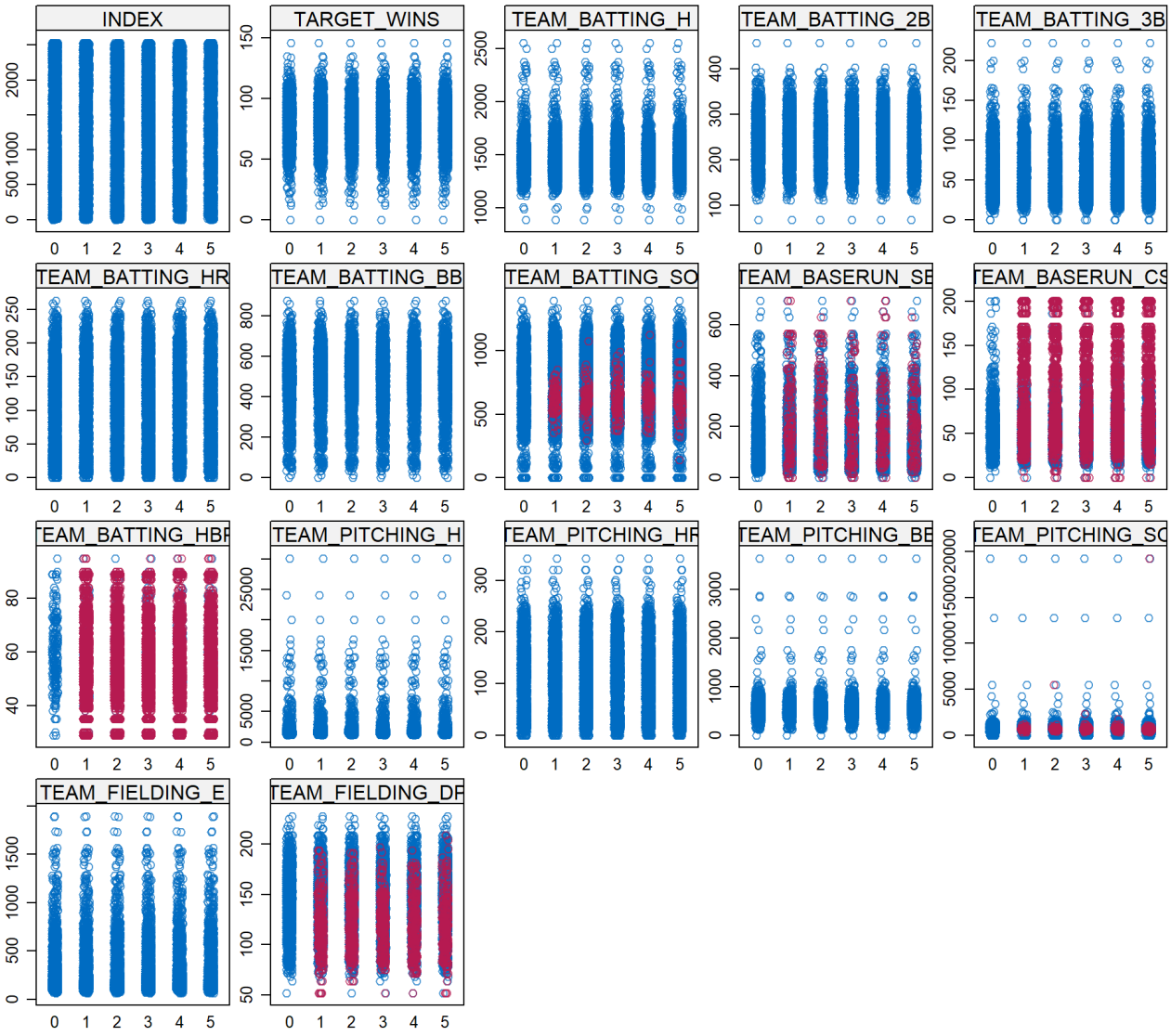


The addition of imputed variables free of missing values to the Moneyball data will be used for predictive modeling, as well as a set of flag variables to determine if a variable that originally presented missing values becomes a predictor.

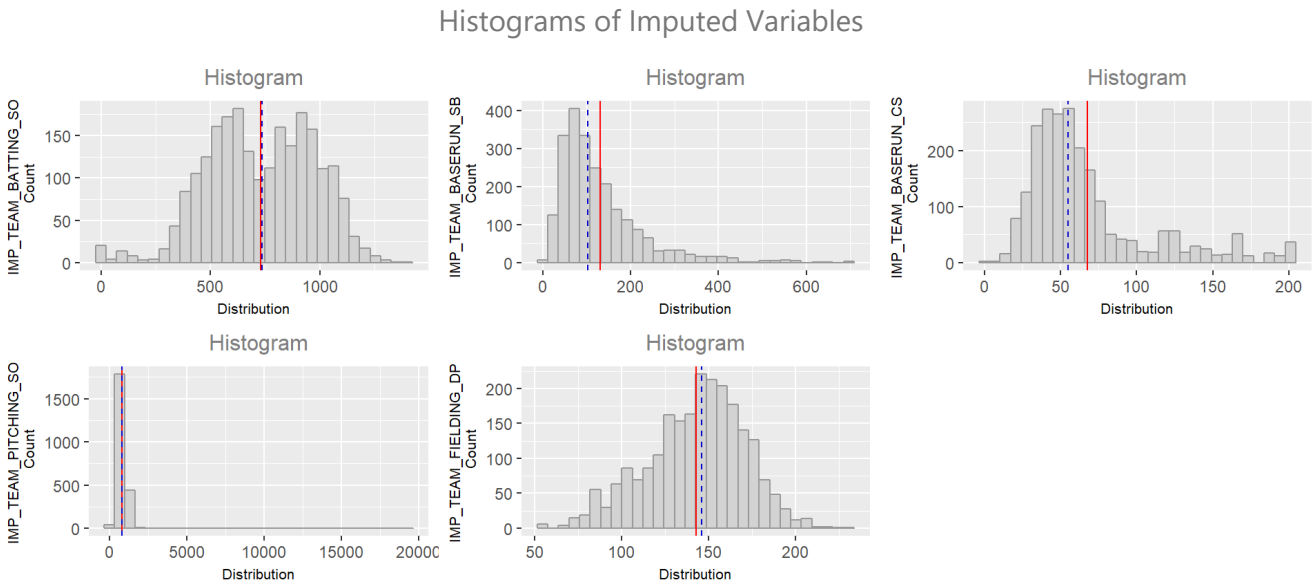
- IMP_X: New Input Variable with all missing values "fixed" added to the Moneyball data set.
- M_X: Flag variable M_X=0 means variable is original 1 means variable imputed.

Variable Elimination: As previously stated, the variable TEAM_BATTING_HBP present 92% of missingness, data gap expected to affect significantly the accuracy of the prediction. Therefore, this variable will be dropped from regression modeling, while TEAM_BASERUN_CS with 34% of missing values and all others remain in the data set for regression analysis.

Comparison between the Original Data and Imputed Data



The variables in the Moneyball training data close to normal distribution remain in the original form for regression modeling, while re-expressing variables with anomalies aims to better interpretation of the models in compliance with the normality and homoscedasticity assumption by optimizing the data from asymmetrical distributions. Moreover, additional pre-processing of the data imposes minimizing the effect of outliers after results from data exploration in section 1, together with the imputed variables displayed in the histograms below.



Evaluating the assumption for normality distribution from all variable histograms, some predictors favor from truncation with a quartile calculation for unlike lower or higher values, and others require square-root and log transformations to normalize moderate or marked distributions adjusting for outliers.

Variable Truncation: Variables with unfeasible zeros, lower values, or higher values undergo trimming to soften the distribution of variables exceeding certain intuitive limit. The process accounts for the effect of multiple extreme and unrealistic values outside bounds in the data set.

Trimming to 1st or 5th percentiles on variables with zeros or unlikely lower values:

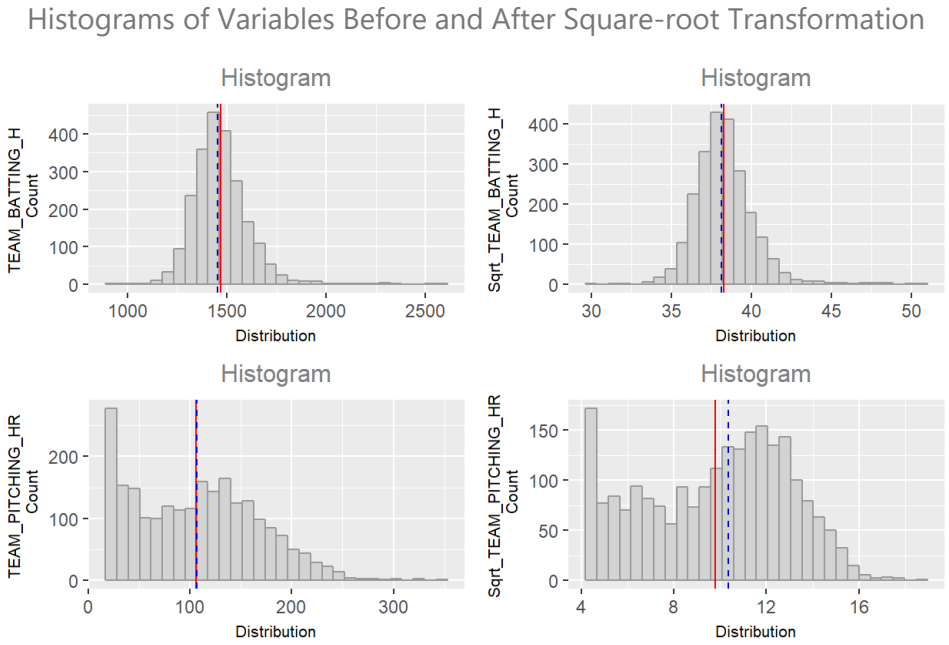
- TEAM_BATTING_3B (5th percentile)
- TEAM_BATTING_HR (5th percentile)
- TEAM_BATTING_BB (5th percentile)
- IMP_TEAM_BATTING_SO (5th percentile)
- IMP_TEAM_BASERUN_SB (5th percentile)
- IMP_TEAM_BASERUN_CS (5th percentile)
- TEAM_PITCHING_BB (1st percentile)
- TEAM_PITCHING_HR (5th percentile)
- IMP_TEAM_PITCHING_SO (1st percentile)

Trimming to a bound, 95th, or 99th percentiles on variables with unrealistic higher values after the limit indicated:

- TEAM_BATTING_3B with high values above 150 (99th percentile)
- IMP_TEAM_BASERUN_SB with high values above 450 (99th percentile)
- IMP_TEAM_BASERUN_CS with high values above 150 (99th percentile)
- TEAM_FIELDING_E with high values above 500
- TEAM_PITCHING_BB with high values above 1,100 (99th percentile)
- TEAM_PITCHING_H with high values above 3,000
- IMP_TEAM_PITCHING_SO with high values after 2,500 (99th percentile)

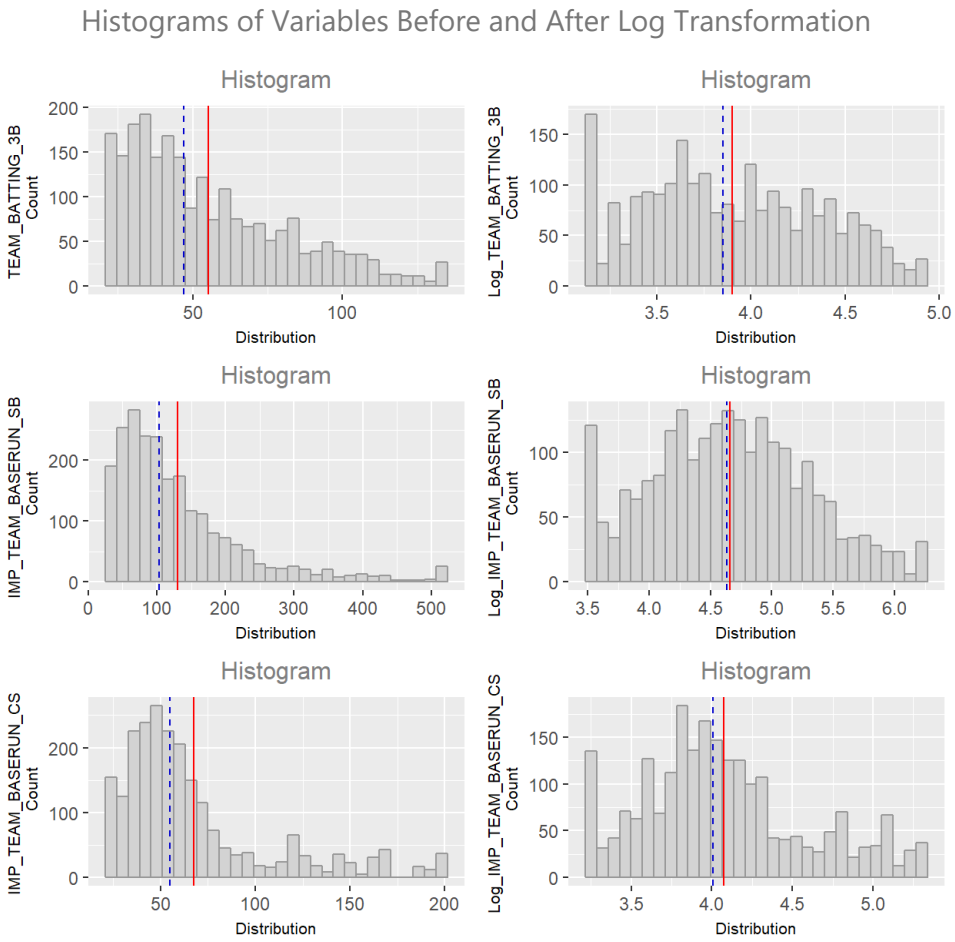
Square-root transformation: The mathematical transformation applied to variables with moderate positive skewness help the distribution to equate group variances.

- TEAM_BATTING_H with high values after 2,000
- TEAM_PITCHING_HR with high values after 250



Log Transformation: Re-expressing variables with marked positive skewness covers the required data preparation to compress the upper end more than the lower end in positive skewed distributions, constraining the effect of extreme outlier values and normalizing distributions before initiating modeling. After determining less significant skewness results from log transformation versus square-root transformations, the transformed variables include:

- TEAM_BATTING_3B with high values after 175
- IMP_TEAM_BASERUN_SB with high values after 400
- IMP_TEAM_BASERUN_CS with high values after 120

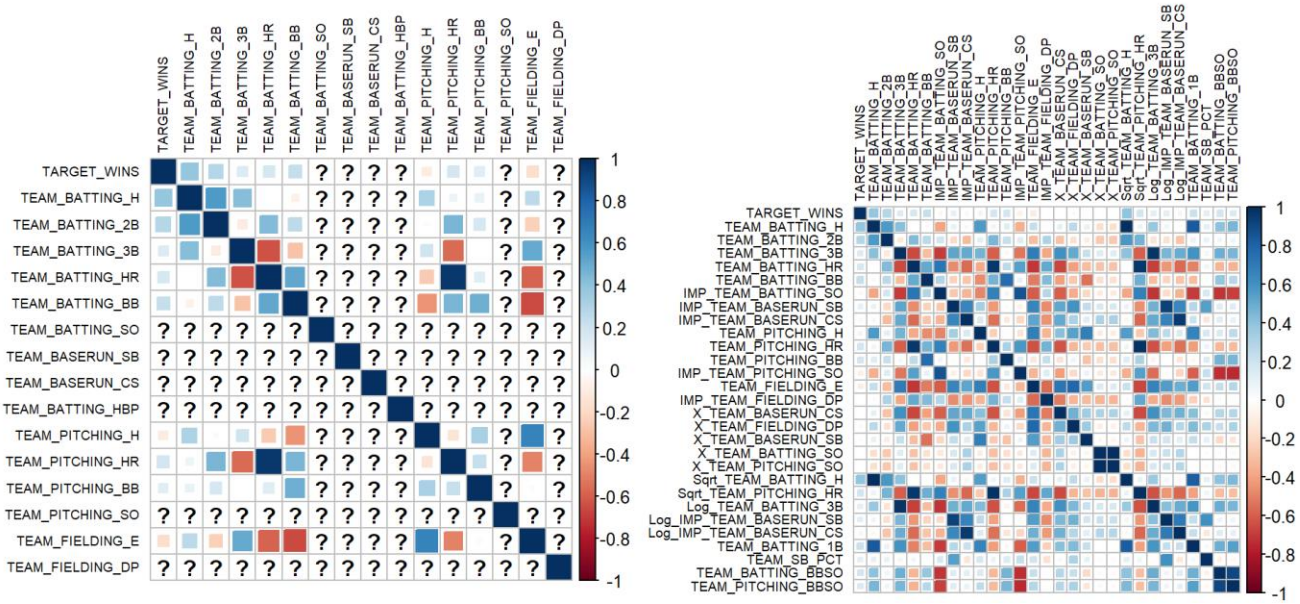


Variable Addition: The combination of variables creates four new variables added to the Moneyball data set to complement the information provided for regression modeling. The new variables consist of:

- Singles by batters (TEAM_BATTING_1B): Complete the set of number of batting by subtracting homeruns by batters (4B), triples by batters (3B), and doubles by batters (2B) from the overall base hits by batters (1B,2B,3B,HR).
- Ratio of stealing bases (TEAM_SB_PTC): Overall review of the effect of TEAM_BASERUN_SB versus TEAM_BASERUN_CS in stealing bases.
- Ratio of walks versus strikeouts by batters (TEAM_BATTING_BBSO): The variable accounts for the positive effect of walks by batters in contrast with the negative impact of strikeouts by batters.
- Ratio of walks allowed versus strikeouts by pitchers (TEAM_PITCHING_BBSO): The variable accounts for the negative effect of walks allowed in contrast with the positive impact of strikeouts by pitchers.

Looking at possible evidence of collinearity with TARGET_WINS and clear multicollinearity in the regressors, the following mosaic matrices compare the initial heatmap of the raw data with the cleaned data set including the effect of new variables.

Heatmap Matrices for Moneyball Variables Before and After Transformations



Section 3: Model Building

After applying multiple variable transformations or re-expression, a refined set of predictors in the Moneyball training data set compensate for adverse effects in predictive analytic assumptions, namely normality and outlier's treatment. The pool of relevant explanatory candidates represents a better sample population for a complete-case analysis of typical baseball seasons. Now, the data set consists of 1 continuous response variable, TARGET_WINS and 23 continuous influential variables to develop several multiple linear regression models inferring on team wins.

By performing multiple OLS regression analysis, the resulting models enable the identification and characterization of relationships among baseball strategies described in the variables information to predict the number of wins that a baseball team will have in a regular season. The diagnostic of models for regression and further selection relies on checking model fit, both by formal statistical tests and graphical methods to verify compliance with linear regression assumptions, and by the model simplicity and performance criteria.

Model building processes include two Automated Variable Selection (AVS) methods (forward and stepwise), an additional variable selection, Principal Component Analysis (PCA) with MLR as appropriate fit for the perceived multicollinearity in the Moneyball training data set, and finally, Random Forest regression.

Forward AVS Model: The model starts with no predictors and just the intercept testing individual addition of the 23 variables from those that reveal a more statistically significant or smallest p-value up to the point that no other variable adds value to the model. Using Akaike information criterion (AIC) criteria as the function of sum of squares of errors and the model size, the measure balances the conflicting demands of accuracy (fit) and simplicity (small number of variables) penalizing the model for large number of variables. Therefore, the optimization criteria for the forward selection method adds predictor variables one-by-one building upon the maximum reduction in AIC criteria in each variable selection by determining that the regression coefficient of each variable is significantly different from zero, retaining the variable in the equation, and continuing setting out to ascertain the next variable.

```
##
## Call:
## lm(formula = TARGET_WINS ~ Sqrt_TEAM_BATTING_H + TEAM_BATTING_2B +
##     Log_TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB +
##     IMP_TEAM_BATTING_SO + Log_IMP_TEAM_BASERUN_SB + Log_IMP_TEAM_BASERUN_CS +
##     TEAM_FIELDING_E + IMP_TEAM_FIELDING_DP + TEAM_PITCHING_BB +
##     TEAM_PITCHING_H + Sqrt_TEAM_PITCHING_HR + IMP_TEAM_PITCHING_SO +
##     X_TEAM_BASERUN_CS + X_TEAM_FIELDING_DP + X_TEAM_BASERUN_SB +
##     X_TEAM_BATTING_SO + TEAM_SB_PCT + TEAM_BATTING_BBSO + TEAM_PITCHING_BBSO,
##     data = moneyball1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.186  -7.642   0.133   7.760  63.183
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.228e+02  1.420e+01  -8.649  < 2e-16 ***
## Sqrt_TEAM_BATTING_H       4.401e+00  3.221e-01  13.662  < 2e-16 ***
## TEAM_BATTING_2B          -2.902e-02  8.713e-03  -3.331  0.00088 ***
## Log_TEAM_BATTING_3B       6.848e+00  9.903e-01   6.916 6.04e-12 ***
## TEAM_BATTING_HR           1.379e-02  2.155e-02   0.640  0.52223
## TEAM_BATTING_BB           6.195e-02  1.115e-02   5.554 3.12e-08 ***
## IMP_TEAM_BATTING_SO      -2.216e-02  7.276e-03  -3.045  0.00235 **
## Log_IMP_TEAM_BASERUN_SB   1.158e+00  4.603e+00   0.252  0.80138
## Log_IMP_TEAM_BASERUN_CS   6.983e+00  4.476e+00   1.560  0.11891
## TEAM_FIELDING_E          -9.392e-02  6.708e-03 -14.001  < 2e-16 ***
## IMP_TEAM_FIELDING_DP     -1.007e-01  1.268e-02  -7.944 3.07e-15 ***
## TEAM_PITCHING_BB          5.347e-03  9.042e-03   0.591  0.55439
## TEAM_PITCHING_H          -6.161e-03  2.254e-03  -2.733  0.00633 **
## Sqrt_TEAM_PITCHING_HR     9.220e-01  4.026e-01   2.290  0.02212 *
## IMP_TEAM_PITCHING_SO     -6.585e-03  5.975e-03  -1.102  0.27058
## X_TEAM_BASERUN_CS         4.664e+00  9.014e-01   5.174 2.49e-07 ***
## X_TEAM_FIELDING_DP        7.845e+00  1.584e+00   4.954 7.81e-07 ***
## X_TEAM_BASERUN_SB         2.683e+01  2.122e+00  12.645  < 2e-16 ***
## X_TEAM_BATTING_SO         7.440e+00  1.494e+00   4.981 6.80e-07 ***
## TEAM_SB_PCT               2.120e+01  2.130e+01   0.996  0.31957
## TEAM_BATTING_BBSO        -1.534e+01  4.827e+00  -3.178  0.00151 **
## TEAM_PITCHING_BBSO       -4.561e+00  3.600e+00  -1.267  0.20538
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.21 on 2254 degrees of freedom
## Multiple R-squared:  0.4047, Adjusted R-squared:  0.3991
## F-statistic: 72.96 on 21 and 2254 DF,  p-value: < 2.2e-16
```

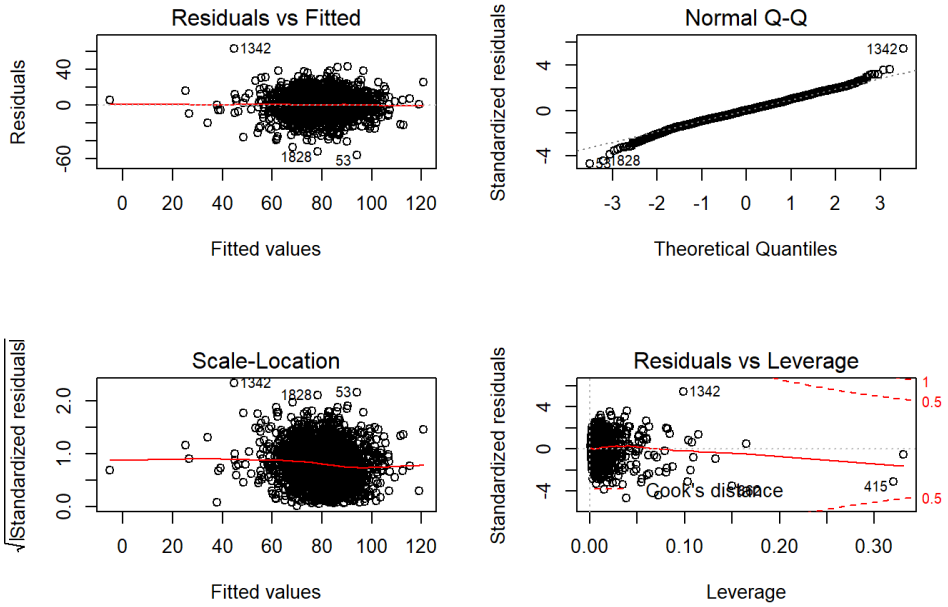
The refined forward selection model dismisses X_TEAM_PITCHING_SO and TEAM_BATTING_1B due to null result in their coefficient estimates and reduces the size of the equation from 23 to 21 variables a provides a high Adjusted R-squared accounting for 39.91% of the variation in TARGET_WINS based on the variables in the model. Also, the model shows serious discrepancies with the signs results in the parameters based on the variables expected impact on TARGET_WINS. Regarding negative results for variables with expected positive impact include doubles by batters (TEAM_BATTING_2B), double plays (IMP_TEAM_FIELDING_DP), strikeouts by pitchers (IMP_TEAM_PITCHING_SO). In contrast, positive regressors coefficients with anticipated negative results comprise caught stealing (Log_IMP_TEAM_BASERUN_CS), walks allowed (TEAM_PITCHING_BB), and homeruns allowed (Sqrt_TEAM_PITCHING_HR).

The model offers significance for 14 individual variables with lower p-values than the t-values, and overall the model also reflects a very small 2.2e-16 p-value compared with the F-statistics, rejecting the null hypothesis (NH).

Noticeable, many variables include VIF output over the acceptable cut-off of 10 unveiling serious collinearity concerns hinting to explore methods to soften the phenomenon.

```
##      Sqrt_TEAM_BATTING_H          TEAM_BATTING_2B      Log_TEAM_BATTING_3B
##                 5.329449                 2.537175                 3.306634
##          TEAM_BATTING_HR          TEAM_BATTING_BB    IMP_TEAM_BATTING_SO
##                25.631846                22.654794                42.327458
## Log_IMP_TEAM_BASERUN_SB Log_IMP_TEAM_BASERUN_CS       TEAM_FIELDING_E
##               132.640525                80.318598                11.278979
##     IMP_TEAM_FIELDING_DP        TEAM_PITCHING_BB       TEAM_PITCHING_H
##                 1.924050                16.827916                10.819053
##    Sqrt_TEAM_PITCHING_HR    IMP_TEAM_PITCHING_SO       X_TEAM_BASERUN_CS
##                24.754735                30.999346                 2.779869
##       X_TEAM_FIELDING_DP       X_TEAM_BASERUN_SB       X_TEAM_BATTING_SO
##                 4.205748                 3.728182                 1.458016
##              TEAM_SB_PCT       TEAM_BATTING_BBSO      TEAM_PITCHING_BBSO
##                65.756137                25.994187                18.860026
```

The goodness-of-fit plot for this model show random distribution of residuals that can be accommodated inside a double bow, then such pattern indicates heteroscedasticity, meaning that the variance of errors is not constant. This shape results from questionable outliers and highly influential points observed in the Q-Q plot and confirmed with the Cook's distance in the residual vs. leverage plot deviate the full compliance with linearity assumption.



Stepwise AVS Model: As a variation of the previous method, the selection process fine-tunes the combination of backward elimination and forward selection by selecting variables based on the assessment of the F statistics. This method is usually preferred as involves analysis at every step to determine the contribution of the predictor variable entered previously in the equation.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + Sqrt_TEAM_BATTING_H +
##     Log_TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB +
##     IMP_TEAM_BATTING_SO + Log_IMP_TEAM_BASERUN_CS + TEAM_FIELDING_E +
##     IMP_TEAM_FIELDING_DP + TEAM_PITCHING_H + Sqrt_TEAM_PITCHING_HR +
##     X_TEAM_BASERUN_CS + X_TEAM_FIELDING_DP + X_TEAM_BASERUN_SB +
##     X_TEAM_BATTING_SO + TEAM_SB_PCT + TEAM_BATTING_BBSO, data = moneyball1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.959  -7.642   0.106   7.747  64.431
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -87.912914  10.760630  -8.170 5.08e-16 ***
## TEAM_BATTING_1B           0.029493   0.008242   3.578 0.000353 ***
## Sqrt_TEAM_BATTING_H       2.196206   0.551798   3.980 7.11e-05 ***
## Log_TEAM_BATTING_3B       8.474253   1.068123   7.934 3.32e-15 ***
## TEAM_BATTING_HR           0.048625   0.022297   2.181 0.029306 *
## TEAM_BATTING_BB           0.066916   0.006370  10.505  < 2e-16 ***
## IMP_TEAM_BATTING_SO      -0.028665   0.004160  -6.891 7.16e-12 ***
## Log_IMP_TEAM_BASERUN_CS   8.202228   0.687701  11.927  < 2e-16 ***
## TEAM_FIELDING_E          -0.093533   0.006597 -14.177  < 2e-16 ***
## IMP_TEAM_FIELDING_DP     -0.101875   0.012574  -8.102 8.74e-16 ***
## TEAM_PITCHING_H          -0.007615   0.001738  -4.381 1.24e-05 ***
## Sqrt_TEAM_PITCHING_HR     0.870146   0.375046   2.320 0.020424 *
## X_TEAM_BASERUN_CS         4.766293   0.894019   5.331 1.07e-07 ***
## X_TEAM_FIELDING_DP        7.717570   1.522148   5.070 4.30e-07 ***
## X_TEAM_BASERUN_SB        26.942156   1.960184  13.745  < 2e-16 ***
## X_TEAM_BATTING_SO         7.104759   1.490178   4.768 1.98e-06 ***
## TEAM_SB_PCT              27.121866   3.145770   8.622  < 2e-16 ***
## TEAM_BATTING_BBSO       -19.129530   3.505696  -5.457 5.38e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.2 on 2258 degrees of freedom
## Multiple R-squared:  0.4044, Adjusted R-squared:    0.4
## F-statistic:  90.2 on 17 and 2258 DF,  p-value: < 2.2e-16
```

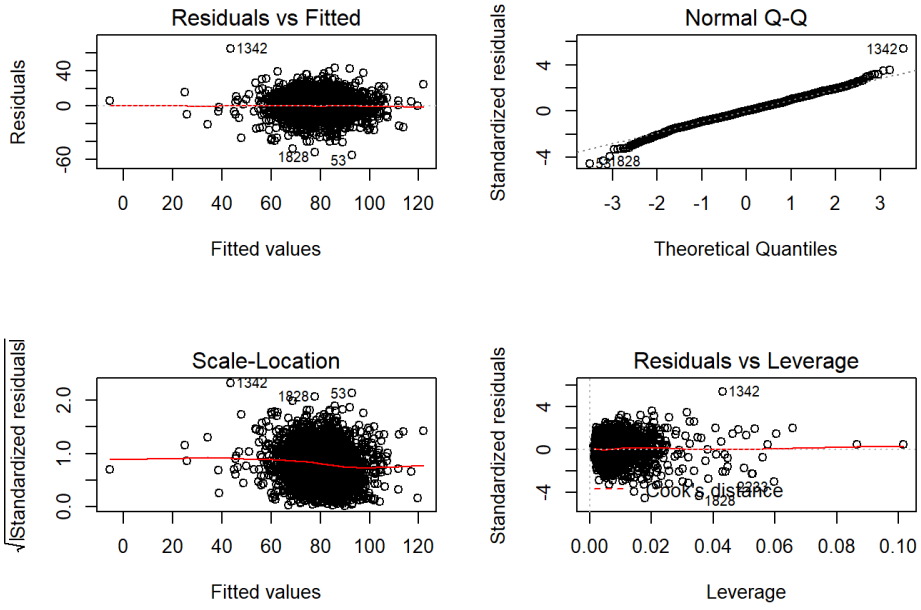
The model reduces the equation from 23 to 17 variables, a simplicity enhancement versus the forward model. In addition, the model produces a better Adjusted R-squared accounting for 40.00% of the variation in TARGET_WINS based on lesser variables in the model. An additional advantage of this model gathers fewer inconsistencies with regressor coefficient signs from 6 in the forward model to only 3 in the stepwise model. The inconsistencies stand for caught stealing (Log_IMP_TEAM_BASERUN_CS) and homeruns allowed (Sqrt_TEAM_PITCHING_HR), both variables that should negatively influence team wins while double plays (IMP_TEAM_FIELDING_DP) must favor the number of wins.

The model offers significance for all 17 individual variables with lower p-values than the t-values, but also for the overall model reflecting a very small 2.2e-16 p-value compared with the F-statistics, rejecting the null hypothesis (NH).

Compared to the forward model, the VIF output disclose multicollinearity in 7 variables compared to 13 in the previous model and in lesser magnitude, therefore a representation of improved model.

```
##          TEAM_BATTING_1B      Sqrt_TEAM_BATTING_H      Log_TEAM_BATTING_3B
##                17.439220                15.658066                 3.852384
##          TEAM_BATTING_HR          TEAM_BATTING_BB     IMP_TEAM_BATTING_SO
##                27.469359                 7.398495                13.855342
## Log_IMP_TEAM_BASERUN_CS         TEAM_FIELDING_E    IMP_TEAM_FIELDING_DP
##                 1.898550                10.925044                 1.894182
##          TEAM_PITCHING_H    Sqrt_TEAM_PITCHING_HR       X_TEAM_BASERUN_CS
##                 6.441831                21.508308                 2.738602
##       X_TEAM_FIELDING_DP       X_TEAM_BASERUN_SB       X_TEAM_BATTING_SO
##                 3.891374                 3.186120                 1.453134
##             TEAM_SB_PCT       TEAM_BATTING_BBSO
##                 1.436834                13.732781
```

As seen in the forward model, the goodness-of-fit plots for the stepwise model produce similar double bow distribution in the variances of the residuals and heteroscedasticity and reasonably normality assumption based on the Q-Q plot.



The downside of using AVS methods resides in the tendency to amplify the statistical significance of the variables that remain in the model, while variables dropped can still be linearly correlated with the response. Therefore, an additional method built, Principal Component Analysis (PCA) address dimension-reduction of the large set of correlated variables into a smaller number of uncorrelated variables named principal components.

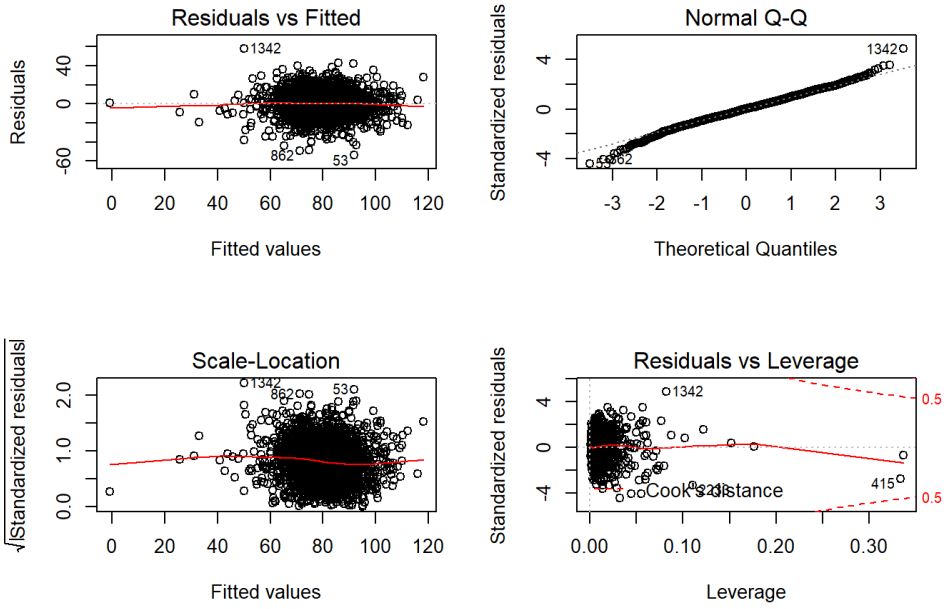
Selected Variables Model: The selection of 19 variables for this model resides in the result offered by both, the forward and stepwise methods and proceed to dismiss those variables implying counterintuitive signs in the parameters estimates, namely caught stealing (LOG_IMP_TEAM_BASERUN_CS), homeruns allowed (Sqrt_TEAM_PITCHING_HR) and double Plays (IMP_TEAM_FIELDING_DP), and well as the flag X_TEAM_PITCHING_SO warning. However, the model produces new errors in coefficient signs, walks allowed (TEAM_PITCHING_BB) should be negative and strikeouts by pitchers (IMP_TEAM_PITCHING_SO) should be positive impact in wins.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + Sqrt_TEAM_BATTING_H +
##     TEAM_BATTING_2B + Log_TEAM_BATTING_3B + TEAM_BATTING_HR +
##     TEAM_BATTING_BB + IMP_TEAM_BATTING_SO + Log_IMP_TEAM_BASERUN_SB +
##     TEAM_FIELDING_E + TEAM_PITCHING_BB + TEAM_PITCHING_H + IMP_TEAM_PITCHING_SO +
##     X_TEAM_BASERUN_CS + X_TEAM_FIELDING_DP + X_TEAM_BASERUN_SB +
##     X_TEAM_BATTING_SO + TEAM_SB_PCT + TEAM_BATTING_BBSO + TEAM_PITCHING_BBSO,
##     data = moneyball1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.817  -7.643   0.323   7.868  57.947
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -60.102519  42.978890  -1.398  0.16212
## TEAM_BATTING_1B            0.043886   0.031419   1.397  0.16261
## Sqrt_TEAM_BATTING_H        0.804129   2.466197   0.326  0.74441
## TEAM_BATTING_2B            0.011223   0.032771   0.342  0.73203
## Log_TEAM_BATTING_3B        9.381761   1.971166   4.759 2.06e-06 ***
## TEAM_BATTING_HR            0.100671   0.034803   2.893  0.00386 **
## TEAM_BATTING_BB            0.056997   0.011139   5.117 3.36e-07 ***
## IMP_TEAM_BATTING_SO       -0.016638   0.007323  -2.272  0.02319 *
## Log_IMP_TEAM_BASERUN_SB    9.094042   0.723120  12.576  < 2e-16 ***
## TEAM_FIELDING_E           -0.089739   0.006607 -13.582  < 2e-16 ***
## TEAM_PITCHING_BB           0.008581   0.008917   0.962  0.33599
## TEAM_PITCHING_H           -0.002655   0.002244  -1.183  0.23683
## IMP_TEAM_PITCHING_SO      -0.010094   0.006055  -1.667  0.09565 .
## X_TEAM_BASERUN_CS          5.477255   0.906570   6.042 1.78e-09 ***
## X_TEAM_FIELDING_DP         7.162560   1.577282   4.541 5.89e-06 ***
## X_TEAM_BASERUN_SB         24.941464   2.132748  11.695  < 2e-16 ***
## X_TEAM_BATTING_SO          8.920760   1.485764   6.004 2.24e-09 ***
## TEAM_SB_PCT              -11.710659   3.951943  -2.963  0.00308 **
## TEAM_BATTING_BBSO        -14.136845   4.916830  -2.875  0.00408 **
## TEAM_PITCHING_BBSO        -6.600640   3.629096  -1.819  0.06907 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.38 on 2256 degrees of freedom
## Multiple R-squared:  0.3871, Adjusted R-squared:  0.382
## F-statistic:    75 on 19 and 2256 DF,  p-value: < 2.2e-16
```

The model produces a lower proportional variation of 38.20% in TARGET_WINS based on the 19 selected variables, with a complex equation of 19 variables, from which 14 are significant based on their p-values. Compared to the previous models, 13 of the VIF values alert of marked multicollinearity.

```
##          TEAM_BATTING_1B      Sqrt_TEAM_BATTING_H          TEAM_BATTING_2B
##                246.018391                303.661632                34.896182
##      Log_TEAM_BATTING_3B          TEAM_BATTING_HR          TEAM_BATTING_BB
##                 12.737663                 64.973070                21.962706
##      IMP_TEAM_BATTING_SO Log_IMP_TEAM_BASERUN_SB          TEAM_FIELDING_E
##                 41.690789                  3.183052                10.638940
##         TEAM_PITCHING_BB          TEAM_PITCHING_H     IMP_TEAM_PITCHING_SO
##                 15.911069                 10.418832                30.950254
##        X_TEAM_BASERUN_CS       X_TEAM_FIELDING_DP        X_TEAM_BASERUN_SB
##                  2.733974                  4.056619                 3.661881
##        X_TEAM_BATTING_SO             TEAM_SB_PCT       TEAM_BATTING_BBSO
##                  1.402446                  2.201561                26.226315
##       TEAM_PITCHING_BBSO
##                 18.630767
```

The model does not represent an improved version of the forward and stepwise model in simplicity or goodness-of-fit plot results.



BONUS* - *PCA Model: The unsupervised learning algorithm determines the common components within the multicollinearity of predictors to be used in building the MLR model. Including the first 13 out of 30 principal components (PCA scores), as predictor variables accounts for over 95% of the variation in the Moneyball training data set used for inference on the response variable TARGET_WINS.

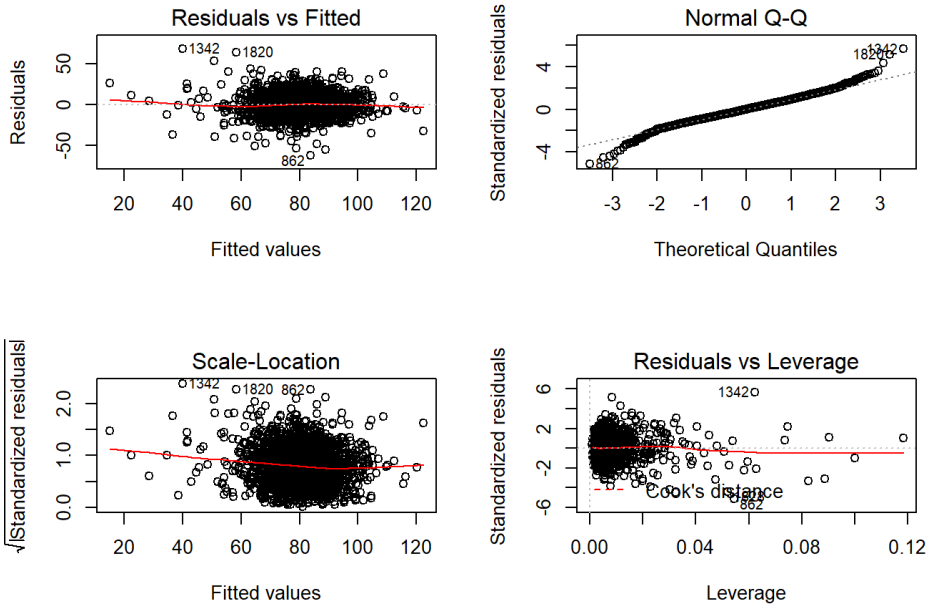
The model builds upon the advantages of PCA and linear regression methods to address collinearity as a multidimensional data property concern in the Moneyball sample with a data set intrinsically autocorrelated. As a data dimension reduction and variability approximation technique, PCA exploration reveals those covariance responsible for the most significant variations in the Moneyball training data set in contrast with the effect of automated variable selection methods. Below, the computations and score of the principal components on the whole data set determine the 13 relevant PCA scores.

```
##         cumulative percentage of variance
## comp 1                            34.58197
## comp 2                            50.65279
## comp 3                            60.89373
## comp 4                            69.17700
## comp 5                            75.18752
## comp 6                            79.51773
## comp 7                            82.97161
## comp 8                            85.96656
## comp 9                            88.43875
## comp 10                           90.44306
## comp 11                           92.41962
## comp 12                           94.11560
## comp 13                           95.57316
## comp 14                           96.70953
## comp 15                           97.77816
## comp 16                           98.42060
## comp 17                           98.82486
## comp 18                           99.07424
## comp 19                           99.29658
## comp 20                           99.49966
## comp 21                           99.66127
## comp 22                           99.79462
## comp 23                           99.86807
## comp 24                           99.93485
## comp 25                           99.96321
## comp 26                           99.98694
## comp 27                           99.99576
## comp 28                          100.00000
## comp 29                          100.00000
## comp 30                          100.00000
```

```
##
## Call:
## lm(formula = moneyball1$TARGET_WINS ~ pc$x[, 1] + pc$x[, 2] +
##     pc$x[, 3] + pc$x[, 4] + pc$x[, 5] + pc$x[, 6] + pc$x[, 7] +
##     pc$x[, 8] + pc$x[, 9] + pc$x[, 10] + pc$x[, 11] + pc$x[,
##     12] + pc$x[, 13])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.844  -7.950  -0.092   7.645  67.898
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 80.7908612  0.2594105 311.440  < 2e-16 ***
## pc$x[, 1]   -0.0005067  0.0003520  -1.440  0.15013
## pc$x[, 2]   -0.0031869  0.0005807  -5.488 4.53e-08 ***
## pc$x[, 3]    0.0007896  0.0009058   0.872  0.38344
```

```
## pc$x[, 4]   -0.0376060  0.0015362 -24.480  < 2e-16 ***
## pc$x[, 5]    0.0061415  0.0020059   3.062  0.00223 **
## pc$x[, 6]   -0.0339990  0.0025666 -13.247  < 2e-16 ***
## pc$x[, 7]    0.0310941  0.0045769   6.794 1.39e-11 ***
## pc$x[, 8]   -0.0699398  0.0050882 -13.745  < 2e-16 ***
## pc$x[, 9]    0.0394211  0.0059056   6.675 3.10e-11 ***
## pc$x[, 10]  -0.0289216  0.0071580  -4.040 5.51e-05 ***
## pc$x[, 11]  -0.0865691  0.0087642  -9.878  < 2e-16 ***
## pc$x[, 12]   0.0780328  0.0096680   8.071 1.12e-15 ***
## pc$x[, 13]  -0.1488085  0.0122373 -12.160  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.38 on 2262 degrees of freedom
## Multiple R-squared:  0.3863, Adjusted R-squared:  0.3827
## F-statistic: 109.5 on 13 and 2262 DF,  p-value: < 2.2e-16
```

While the performance of PCA in MLR determines less variables in the construction of the model, the 13 PC with 11 of them being significant for inference, on the other hand, 38.27% of the variation is justified by the model, lower than the previous models.



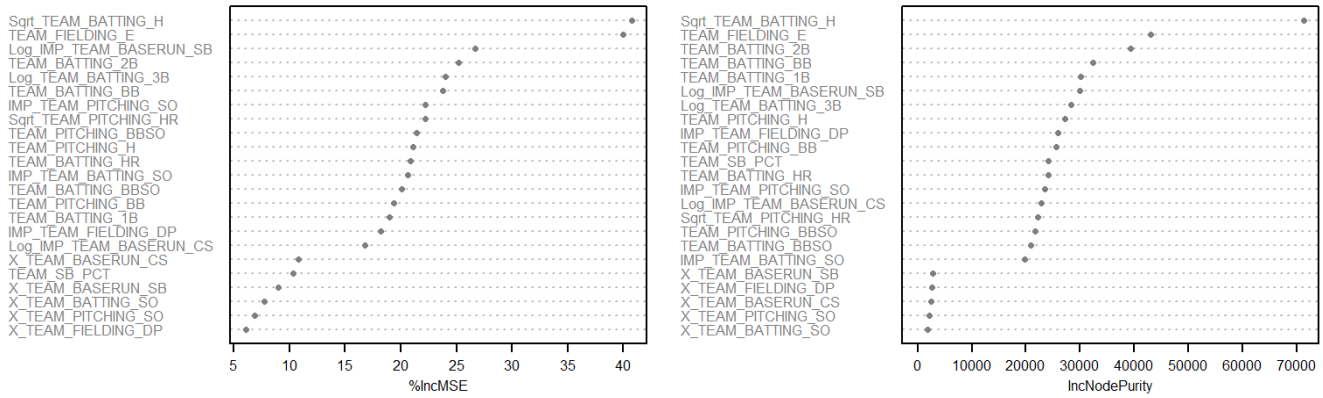
The patterns of residuals shown in the plot maintain the same characteristics seeing in the previous three models with latent heteroscedasticity and close to linearity of the residuals.

BONUS* - *Random Forest Regression Model: The advantage of this supervised learning algorithm relies on its simplicity for regression modeling tasks by building and growing multiple decision trees, ntree = 500, high number used in the regression to minimize the error and merging them together to get a more accurate and stable prediction. Evidently, the 41.26% variation on TARGET_WINS using this model leads to the highest percentage, although using all 23 predictors. The visualizations show the scaled importance of predictor and the error rate produced by the application.

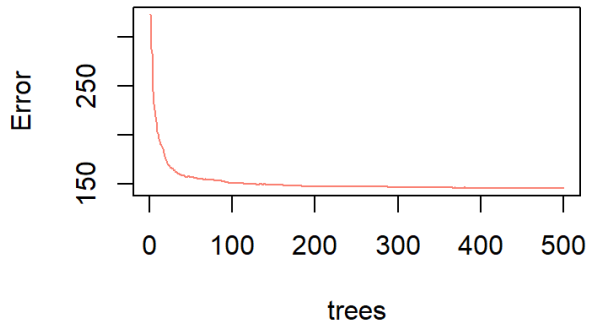
```
##
## Call:
##  randomForest(formula = TARGET_WINS ~ TEAM_BATTING_1B + Sqrt_TEAM_BATTING_H +      TEAM
_BATTING_2B + Log_TEAM_BATTING_3B + TEAM_BATTING_HR +      TEAM_BATTING_BB + IMP_TEAM_BATT
ING_SO + Log_IMP_TEAM_BASERUN_SB +      Log_IMP_TEAM_BASERUN_CS + TEAM_FIELDING_E + IMP_TE
AM_FIELDING_DP +      TEAM_PITCHING_BB + TEAM_PITCHING_H + Sqrt_TEAM_PITCHING_HR +      IM
P_TEAM_PITCHING_SO + X_TEAM_BASERUN_CS + X_TEAM_FIELDING_DP +      X_TEAM_BASERUN_SB + X_T
EAM_BATTING_SO + X_TEAM_PITCHING_SO +      TEAM_SB_PCT + TEAM_BATTING_BBSO + TEAM_PITCHING
_BBSO, data = moneyball1,      ntree = 500, importance = TRUE)
##               Type of random forest: regression
##                     Number of trees: 500
## No. of variables tried at each split: 7
##
##           Mean of squared residuals: 145.6832
##                     % Var explained: 41.26
##                         %IncMSE IncNodePurity
## TEAM_BATTING_1B         21.378876     30091.787
## Sqrt_TEAM_BATTING_H     79.481660     71194.797
## TEAM_BATTING_2B         23.627143     39320.652
## Log_TEAM_BATTING_3B     27.281200     28274.806
## TEAM_BATTING_HR         22.480068     24054.397
## TEAM_BATTING_BB         26.574914     32357.193
## IMP_TEAM_BATTING_SO     16.855538     19786.111
## Log_IMP_TEAM_BASERUN_SB 22.578469     29908.876
## Log_IMP_TEAM_BASERUN_CS 10.058480     22795.029
## TEAM_FIELDING_E         57.408390     43046.800
## IMP_TEAM_FIELDING_DP    10.481811     25873.148
## TEAM_PITCHING_BB        17.373304     25513.153
## TEAM_PITCHING_H         18.628819     27127.492
## Sqrt_TEAM_PITCHING_HR   21.082693     22207.254
## IMP_TEAM_PITCHING_SO    15.997485     23403.932
## X_TEAM_BASERUN_CS        4.173243      2558.186
## X_TEAM_FIELDING_DP       2.071664      2718.996
## X_TEAM_BASERUN_SB        3.763207      2737.176
## X_TEAM_BATTING_SO        1.136839      1866.166
## X_TEAM_PITCHING_SO       1.389456      2125.567
## TEAM_SB_PCT              4.492675     24156.959
## TEAM_BATTING_BBSO       16.530895     20866.079
## TEAM_PITCHING_BBSO      16.238933     21693.334
```

As in a real baseball game, the relevant variables from the model stand out as the combination of base hits by batters (Sqrt_TEAM_BATTING_H) and the errors (TEAM_FIELDING E), while the flag variables indicating imputation could be drop due to the lower significance.

Importance of Variables from Random Forest Regression



Error Rate of Random Forest Regression



Section 4: Model Selection

The following quality model estimators present calculations to aid interpretation and trade-off criteria between simplicity and accuracy for the selection of the best regression model among the five developed.

Heatmap of Criteria for Model Selection

Criteria	Forward AVS Model	Stepwise AVS Model	Selected Variables Model	PCA Model	Random Forest Regression Model
Number of variables	21	17	19	13	23
Adjusted R-Squared (aR2)	39.91%	40.00%	38.20%	38.27%	41.26%
Akaike Information Criteria (AIC)	17873.37	17866.2	17935.49	17926.63	-
Bayesian Information Criterion (BIC)	18005.16	17975.07	18055.82	18012.58	-
Mean Squared Error (MSE)	147.6559	147.7098	152.0085	152.2186	-
Mean Absolute Error (MAE)	9.463907	9.464	9.629421	9.480054	3.821195

Adjusted R-squared: As a bounded scaled measure 0 to 1 of the variability in the response variable based on predictors conforming each model, this computation serves as cross-comparator of goodness of fit between the five regression models adjusting for unequal number of predictor variables. In that order, the Random Forest regression gives the highest variation on TARGET_WINS based on the predictors in the model (41.26%), however it includes all variables in the model with two as the most significant. Therefore, the best balance

between higher precision or variation explained and a reduced model in number of variable relies on the Stepwise model.

Akaike Information Criterion (AIC): While comparing the five models, the smaller Akaike information criterion estimator value resulted from the Stepwise AVS model, meaning that variables in the model better predict the response variable TARGET_WINS. The measure based on in-sample fit to estimate the likelihood of a model to predict future values selects the best model by balancing conflicting demands of accuracy (fit) and simplicity (small number of variables), while penalizing other models due to their additional variable selection.

Bayesian Information Criterion (BIC): As an index used for choosing between competing models based of the smaller values, the BIC as well as the AIC measure, both penalized likelihood criteria by choosing the best predictor subsets and determining a penalty for increasing the number of parameters. Their practical difference relates to the size of the penalty, where BIC penalizes model complexity of the Stepwise AVS model less heavily.

Mean Squared Error (MSE): The expected goal for the five models, stand for reducing the MSE for the model to be less bias (more accurate in prediction) and for the variance of errors to be smaller (more precision) from a formula composed of the two positive values. This way, the smaller MSE obtained from the AVS models (Forward and Stepwise) indicates that they provide more accuracy since includes more explanatory variables in the equation.

Mean Absolute Error (MAE): As an average of the absolute errors, MAE measures how close inferences of the TARGET_WINS are to the team wins, being in the same scale of data set being measured. Then, MAE determines better the bias on the model. In that sense, the Random Forest Regression model followed by both AVS models prove to be less bias or more accurate with the disadvantage that leads to a loss of precision for estimation or prediction of the TARGET_WINS.

From the assessment of different indicators, the best performance to explain the variation on TARGET_WINS favor the Stepwise AVS model after consideration of trade-off between variation of the response variable supported by the model with the reduced number of variables, as well as lower results of AIC, BIC, and MSE indicators, negatively-oriented scores.

Section 5: Model Deployment Code

A stand-alone R data step scores new data using the Moneyball test data set of 259 observations, the identifier INDEX, and the same explanatory variables to predict new results. The first steps in the code incorporate the necessary cleaning of the testing data set, as performed for the training data set, including fixing missing values, variable transformations, and variable elimination and additions. Finally, the model deployment code runs the scores of the new data with the Stepwise champion model using the selected regression equation formula to predict the number of wins. The variable with the predicted number of team wins, P_TARGET_WINS assesses the accuracy of the scoring program.

The scored data file in CSV format produces 259 target scores, with a mean of 79.44 wins, close to the median of 80.41, and logical positive results from the minimum number of 23.81 wins (non-zeros and non-negative values) and the maximum value of 116.04 wins, realistic considering that 162 wins is the highest bound, but unlikely.

```
################################### Section 5: Model Deployment Code

# Setwd("D:/RFile/")
Setwd("D:/RFile/")

# Read the Moneyball test data set
moneyball_test=read.csv("moneyball_test.csv",header=T)

# Initiate missing-data imputations
library(mice)
init = mice(moneyball_test, maxit=0)
meth = init$method
predM = init$predictorMatrix

# Removal of INDEX as predictor
predM[, c("INDEX")]=0

# Missing-data imputation
method1_test <- mice(moneyball_test, m=5, maxit=5, meth="rf", predictorMatrix=predM, seed=500)

# Completed data matrix after imputations
moneyball_test1 <- complete(method1_test,1)

# Flag variables with imputation as IMP_X:
names(moneyball_test1)[names(moneyball_test1)=="TEAM_BATTING_HBP"] <- "IMP_TEAM_BATTING_HBP"
names(moneyball_test1)[names(moneyball_test1)=="TEAM_BASERUN_CS"] <- "IMP_TEAM_BASERUN_CS"
names(moneyball_test1)[names(moneyball_test1)=="TEAM_FIELDING_DP"] <- "IMP_TEAM_FIELDING_DP"
names(moneyball_test1)[names(moneyball_test1)=="TEAM_BASERUN_SB"] <- "IMP_TEAM_BASERUN_SB"
names(moneyball_test1)[names(moneyball_test1)=="TEAM_BATTING_SO"] <- "IMP_TEAM_BATTING_SO"
names(moneyball_test1)[names(moneyball_test1)=="TEAM_PITCHING_SO"] <- "IMP_TEAM_PITCHING_SO"

# Flag variables with missing values M_X:
moneyball_test1$X_TEAM_BATTING_HBP <- ifelse(is.na(moneyball_test$TEAM_BATTING_HBP), 1, 0)
moneyball_test1$X_TEAM_BASERUN_CS <- ifelse(is.na(moneyball_test$TEAM_BASERUN_CS), 1, 0)
moneyball_test1$X_TEAM_FIELDING_DP <- ifelse(is.na(moneyball_test$TEAM_FIELDING_DP), 1, 0)
moneyball_test1$X_TEAM_BASERUN_SB <- ifelse(is.na(moneyball_test$TEAM_BASERUN_SB), 1, 0)
moneyball_test1$X_TEAM_BATTING_SO <- ifelse(is.na(moneyball_test$TEAM_BATTING_SO), 1, 0)
moneyball_test1$X_TEAM_PITCHING_SO <- ifelse(is.na(moneyball_test$TEAM_PITCHING_SO), 1, 0)

# Variable elimination
moneyball_test1 <- subset(moneyball_test1, select = -IMP_TEAM_BATTING_HBP)
moneyball_test1 <- subset(moneyball_test1, select = -X_TEAM_BATTING_HBP)

# Trimming to 1st or 5th percentiles on variables with zeros or unlikely lower values
moneyball_test1$TEAM_BATTING_3B[(moneyball_test1$TEAM_BATTING_3B < quantile(moneyball_test1$TEAM_BATTING_3B,
0.05))] =
quantile(moneyball_test1$TEAM_BATTING_3B, 0.05)
moneyball_test1$TEAM_BATTING_HR[(moneyball_test1$TEAM_BATTING_HR < quantile(moneyball_test1$TEAM_BATTING_HR,
0.05))] =
quantile(moneyball_test1$TEAM_BATTING_HR, 0.05)
moneyball_test1$TEAM_BATTING_BB[(moneyball_test1$TEAM_BATTING_BB < quantile(moneyball_test1$TEAM_BATTING_BB,
0.05))] =
quantile(moneyball_test1$TEAM_BATTING_BB, 0.05)
moneyball_test1$IMP_TEAM_BATTING_SO[(moneyball_test1$IMP_TEAM_BATTING_SO <
quantile(moneyball_test1$IMP_TEAM_BATTING_SO,
0.05))] = quantile(moneyball_test1$IMP_TEAM_BATTING_SO, 0.05)
moneyball_test1$IMP_TEAM_BASERUN_SB[(moneyball_test1$IMP_TEAM_BASERUN_SB <
quantile(moneyball_test1$IMP_TEAM_BASERUN_SB,
0.05))] = quantile(moneyball_test1$IMP_TEAM_BASERUN_SB, 0.05)
moneyball_test1$IMP_TEAM_BASERUN_CS[(moneyball_test1$IMP_TEAM_BASERUN_CS <
quantile(moneyball_test1$IMP_TEAM_BASERUN_CS,
0.05))] = quantile(moneyball_test1$IMP_TEAM_BASERUN_CS, 0.05)
moneyball_test1$TEAM_PITCHING_BB[(moneyball_test1$TEAM_PITCHING_BB <
quantile(moneyball_test1$TEAM_PITCHING_BB, 0.01))] =
quantile(moneyball_test1$TEAM_PITCHING_BB, 0.01)
moneyball_test1$TEAM_PITCHING_HR[(moneyball_test1$TEAM_PITCHING_HR <
quantile(moneyball_test1$TEAM_PITCHING_HR, 0.05))] =
quantile(moneyball_test1$TEAM_PITCHING_HR, 0.05)
moneyball_test1$IMP_TEAM_PITCHING_SO[(moneyball_test1$IMP_TEAM_PITCHING_SO <
quantile(moneyball_test1$IMP_TEAM_PITCHING_SO,
0.01))] = quantile(moneyball_test1$IMP_TEAM_PITCHING_SO, 0.01)

# Trimming to 95th or 99th percentiles on variables with unrealistic higher values after the limit indicated
moneyball_test1$TEAM_BATTING_3B[(moneyball_test1$TEAM_BATTING_3B > quantile(moneyball_test1$TEAM_BATTING_3B,
0.99))] = quantile(moneyball_test1$TEAM_BATTING_3B, 0.99)
moneyball_test1$IMP_TEAM_BASERUN_SB[(moneyball_test1$IMP_TEAM_BASERUN_SB >
quantile(moneyball_test1$IMP_TEAM_BASERUN_SB, 0.99))] = quantile(moneyball_test1$IMP_TEAM_BASERUN_SB, 0.99)
moneyball_test1$IMP_TEAM_BASERUN_CS[(moneyball_test1$IMP_TEAM_BASERUN_CS >
quantile(moneyball_test1$IMP_TEAM_BASERUN_CS, 0.99))] = quantile(moneyball_test1$IMP_TEAM_BASERUN_CS, 0.99)
moneyball_test1$TEAM_FIELDING_E[(moneyball_test1$TEAM_FIELDING_E > 500)] = 500
```

```
moneyball_test1$TEAM_PITCHING_BB[(moneyball_test1$TEAM_PITCHING_BB >
quantile(moneyball_test1$TEAM_PITCHING_BB, 0.99))] =
quantile(moneyball_test1$TEAM_PITCHING_BB, 0.99)
moneyball_test1$TEAM_PITCHING_H[(moneyball_test1$TEAM_PITCHING_H > 5000)] = 5000
moneyball_test1$IMP_TEAM_PITCHING_SO[(moneyball_test1$IMP_TEAM_PITCHING_SO >
quantile(moneyball_test1$IMP_TEAM_PITCHING_SO,
0.99))] = quantile(moneyball_test1$IMP_TEAM_PITCHING_SO, 0.99)

# Square-root transformation
moneyball_test1$Sqrt_TEAM_BATTING_H <- sqrt(moneyball_test1$TEAM_BATTING_H)
moneyball_test1$Sqrt_TEAM_PITCHING_HR <- sqrt(moneyball_test1$TEAM_PITCHING_HR)

# Log transformation
moneyball_test1$Log_TEAM_BATTING_3B <- log(moneyball_test1$TEAM_BATTING_3B)
moneyball_test1$Log_IMP_TEAM_BASERUN_SB <- log(moneyball_test1$IMP_TEAM_BASERUN_SB)
moneyball_test1$Log_IMP_TEAM_BASERUN_CS <- log(moneyball_test1$IMP_TEAM_BASERUN_CS)

# Variable addition
# Singles by batters (TEAM_BATTING_1B)
moneyball_test1$TEAM_BATTING_1B <- moneyball_test1$TEAM_BATTING_H - moneyball_test1$TEAM_BATTING_HR -
moneyball_test1$TEAM_BATTING_3B - moneyball_test1$TEAM_BATTING_2B

# Ratio of stealing bases (TEAM_SB_PTC)
moneyball_test1$TEAM_SB_PCT =
moneyball_test1$IMP_TEAM_BASERUN_SB/(1.0*moneyball_test1$IMP_TEAM_BASERUN_SB+moneyball_test1$IMP_TEAM_BASERU
N_CS)
moneyball_test1$TEAM_SB_PCT[is.na(moneyball_test1$TEAM_SB_PCT)] = mean(moneyball_test1$TEAM_SB_PCT, na.rm =
TRUE)

# Ratio of walks versus strikeouts by batters (TEAM_BATTING_BBSO)
moneyball_test1$TEAM_BATTING_BBSO <- moneyball_test1$TEAM_BATTING_BB/moneyball_test1$IMP_TEAM_BATTING_SO

# Ratio of walks allowed versus strikeouts by pitchers (TEAM_PITCHING_BBSO)
moneyball_test1$TEAM_PITCHING_BBSO <- moneyball_test1$TEAM_PITCHING_BB/moneyball_test1$IMP_TEAM_PITCHING_SO

# Check completeness of the data
summary(moneyball_test1)

# Stand Alone Scoring
moneyball_test1$P_TARGET_WINS <- -87.912914 +
0.029493 * moneyball_test1$TEAM_BATTING_1B +
2.196206 * moneyball_test1$Sqrt_TEAM_BATTING_H +
8.474253 * moneyball_test1$Log_TEAM_BATTING_3B +
0.048625 * moneyball_test1$TEAM_BATTING_HR +
0.066916 * moneyball_test1$TEAM_BATTING_BB -
0.028665 * moneyball_test1$IMP_TEAM_BATTING_SO +
8.202228 * moneyball_test1$Log_IMP_TEAM_BASERUN_CS -
0.093533 * moneyball_test1$TEAM_FIELDING_E -
0.101875 * moneyball_test1$IMP_TEAM_FIELDING_DP -
0.007615 * moneyball_test1$TEAM_PITCHING_H +
0.870146 * moneyball_test1$Sqrt_TEAM_PITCHING_HR +
4.766293 * moneyball_test1$X_TEAM_BASERUN_CS +
7.717570 * moneyball_test1$X_TEAM_FIELDING_DP +
26.942156 * moneyball_test1$X_TEAM_BASERUN_SB +
7.104759 * moneyball_test1$X_TEAM_BATTING_SO +
27.121866 * moneyball_test1$TEAM_SB_PCT -
19.129530 * moneyball_test1$TEAM_BATTING_BBSO

# Subset of data set for the deliverable "Scored data file"
prediction <- moneyball_test1[c("INDEX","P_TARGET_WINS")]

# Prediction CSV file
write.csv(prediction, file = "Name_write.csv", row.names = FALSE)
```

Summary/Conclusions

The objective of the report includes the analysis of information contained in a provided Moneyball data set to select the best model for predicting the number of wins (TARGET_WINS) for a professional baseball team during a season. The report gathers a holistic view of the preparation, construction, evaluation, and deployment of multiple linear regression process, starting with EDA of relevant predictor variables in the Moneyball training data set, clean-up of missing values, and required transformations to address assumptions of linear regression. The process continues with the specification of diverse models using AVS, PCA and Random Forest regression methods accompanied with parameter estimation and supported by ANOVA test and goodness-of-feet plots to ensure model adequacy checking. Separate analysis on multicollinearity and PCA, assisted the regression modeling since many variables in the full set evidenced to be highly correlated.

The final focus of the study relied on model deployment, as the final step before real-life use of the regression modeling. For cross-validation, the selection the Stepwise AVS model, as the highest model performance based on fit and accuracy, together with the Moneyball test data set prove the prediction power of the model out-of-sample useful for actual model use.

References

Chatterjee, S. (2012). Regression Analysis by Example. (5th Edition). New York, NY: Wiley.

Davies, T. (2016). The Book of R: A First Course in Programming and Statistics (Kindle edition). San Francisco, CA: No Starch Press.

Everitt, B. (2009). Multivariable Modeling and Multivariate Analysis for the Behavioral Sciences (Chapman & Hall/CRC Statistics in the Social and Behavioral Sciences). Boca Raton, FL: CRC Press.

Fox, J. and Weisberg, S. (2011). An R Companion to Applied Regression. (2nd Edition). Thousand Oaks, CA: SAGE Press.

Hoffmann, J (2004). Generalized Linear Models, An Applied Approach. (Custom Edition for Northwestern University). Boston, MA: Pearson Education.

Lander, J (2014). R for Everyone: Advanced Analytics and Graphics (Addison-Wesley Data & Analytics Series). Upper Saddle River, NJ: Pearson Education.

Appendix: R Codes

The complete set of R codes in the analysis replicate the production of outputs from the report.

```
# Packages
library(moments)
library(ggplot2)
library(coefplot)
library(tcltk)
library(asbio)
library(RColorBrewer)
library(gridExtra)
library(qcc)
library(stats)
library(outliers)
library(PerformanceAnalytics)
library(data.table)
library(flux)
library(GGally)
library(lattice)
library(corrplot)
library(testthat)
library(e1071)
library(magrittr)
library(BAS)
require(cluster)
require(useful)
require(Hmisc)
library(HSAUR)
library(MVA)
library(HSAUR2)
library(fpc)
library(mclust)

# Unit 1 packages
require(ggplot2)
library(rJava)
library(readr)
library(pbkrtest)
library(car)
library(leaps)
library(MASS)
library(xlsxjars)
library(xlsx)

# Table formating packages
library(formattable)
library(kableExtra)

# Decision tree packages
library(party)
library(rpart)
library(randomForest)
library(tree)

# Missing data package
library(mice)


##################################### Section 1: Data Exploration

# Read the Moneyball training data set
moneyball=read.csv("moneyball.csv",header=T)

# Structure of the Moneyball training data set
str(moneyball)

# Read the Data Dictionary and plot in a table
library(openxlsx)
DataDictionary_Baseball=read.xlsx("DataDictionary_Baseball.xlsx", sheet = 1, startRow = 1, colNames = TRUE)

DataDictionary_Baseball %>%
  kable(caption = "Data Dictionary") %>%
  kable_styling(bootstrap_options = c("condensed","responsive", "hover", "bordered"),
                font_size = 9, full_width = F, "responsive", position = "center") %>%
row_spec(c(9,11,12,14,15,16), bold = T, color = "red") %>%
row_spec(c(3,4,5,6,7,8,10,13,17), bold = T, color = "green")
```

```
# Histogram of missing values by variables
missing_values <- data.frame(col = as.character(colnames(moneyball[2:17])),
                    pct_null = colSums(is.na(moneyball[2:17]))*100/(colSums(is.na(moneyball[2:17]))+
                                                                    colSums(!is.na(moneyball[2:17]))))
ggplot(missing_values, aes(x = col, y = pct_null, label_value()))+
  geom_bar(stat = "identity", fill=I("salmon"), alpha=0.6, color=I("gray60"))+
  coord_flip(ylim = c(0, 100))+
  labs(title = "Histogram of Missing Values", x = element_blank(), y = "Percent")+
  theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
  theme(axis.text = element_text(size=7)) +
  theme(axis.title = element_text(size=8)) +
  geom_hline(yintercept = 25, color="red")

# Descriptive and Graphic Statistics of TARGET_WINS
library(fBasics)
kable(t(basicStats(moneyball[2])), digits=2, nsmall=2, caption = "Descriptive and Graphic Statistics of
TARGET_WINS") %>%
  kable_styling(bootstrap_options = c("striped","condensed","responsive", "hover", "bordered"),font_size =
9, full_width = F, position = "center")%>%
column_spec(3, width = "1cm")

sample <- quantile(moneyball$TARGET_WINS, c(0.25, 0.75))
theoretical <- qnorm(c(0.25, 0.75))
slope <- diff(sample)/diff(theoretical)
intercept <- sample - slope*theoretical

grid.arrange(
  ggplot (moneyball, aes(TARGET_WINS)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TARGET_WINS \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TARGET_WINS), color="red") +
    geom_vline(xintercept=median(moneyball$TARGET_WINS), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TARGET_WINS)) +
    geom_boxplot(aes(color=TARGET_WINS), outlier.colour="indianred", outlier.shape=19, outlier.alpha=0.2,
outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TARGET_WINS \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(NULL, aes(sample=moneyball$TARGET_WINS)) + stat_qq(shape=21, fill="lightgrey", color="gray60") +
    ggtitle("QQ-plot") + labs(x="Theoretical Quantiles", y="TARGET_WINS \n Sample Quantiles") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_abline(slope=slope, intercept=intercept, color="gray70"),
    ncol=3)

# Descriptive and Graphic Statistics of Batting variables
kable(t(basicStats(moneyball[c(3,4,5,6,7,11,8)])), digits=2, nsmall=2, caption = "Descriptive and Graphic
Statistics of Batting variables") %>%
  kable_styling(bootstrap_options = c("striped","condensed","responsive", "hover", "bordered"),font_size =
9, full_width = F, position = "center")%>%
column_spec(3, width = "1cm")

grid.arrange(
  ggplot (moneyball, aes(TEAM_BATTING_H)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_H \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BATTING_H), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BATTING_H), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BATTING_H)) +
    geom_boxplot(aes(color=TEAM_BATTING_H), outlier.colour="indianred", outlier.shape=19, outlier.alpha=0.2,
outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BATTING_H  \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BATTING_H, moneyball$TARGET_WINS, color=TEAM_BATTING_H)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
```

```
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BATTING_H", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_BATTING_2B)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_2B \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BATTING_2B), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BATTING_2B), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BATTING_2B)) +
    geom_boxplot(aes(color=TEAM_BATTING_2B), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BATTING_2B  \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BATTING_2B, moneyball$TARGET_WINS, color=TEAM_BATTING_2B)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BATTING_2B", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_BATTING_3B)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_3B \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BATTING_3B), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BATTING_3B), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BATTING_3B)) +
    geom_boxplot(aes(color=TEAM_BATTING_3B), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BATTING_3B  \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BATTING_3B, moneyball$TARGET_WINS, color=TEAM_BATTING_3B)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BATTING_3B", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_BATTING_HR)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_HR \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
```

```
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BATTING_HR), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BATTING_HR), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BATTING_HR)) +
    geom_boxplot(aes(color=TEAM_BATTING_HR), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BATTING_HR  \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BATTING_HR, moneyball$TARGET_WINS, color=TEAM_BATTING_HR)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BATTING_HR", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_BATTING_BB)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_BB \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BATTING_BB), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BATTING_BB), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BATTING_BB)) +
    geom_boxplot(aes(color=TEAM_BATTING_BB), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BATTING_BB  \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BATTING_BB, moneyball$TARGET_WINS, color=TEAM_BATTING_BB)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BATTING_BB", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_BATTING_HBP)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_HBP \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BATTING_HBP), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BATTING_HBP), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BATTING_HBP)) +
    geom_boxplot(aes(color=TEAM_BATTING_HBP), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BATTING_HBP \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BATTING_HBP, moneyball$TARGET_WINS, color=TEAM_BATTING_HBP)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
```

```
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BATTING_HBP", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_BATTING_SO)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_SO \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BATTING_SO), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BATTING_SO), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BATTING_SO)) +
    geom_boxplot(aes(color=TEAM_BATTING_SO), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BATTING_SO \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BATTING_SO, moneyball$TARGET_WINS, color=TEAM_BATTING_SO)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BATTING_SO", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

# Descriptive and Graphic Statistics of Baserun variables
kable(t(basicStats(moneyball[9:10])), digits=2, nsmall=2, caption = "Descriptive and Graphic Statistics of
Baserun variables") %>%
  kable_styling(bootstrap_options = c("striped","condensed","responsive", "hover", "bordered"),font_size =
9, full_width = F, position = "center")%>%
column_spec(3, width = "1cm")

grid.arrange(
  ggplot (moneyball, aes(TEAM_BASERUN_SB)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BASERUN_SB \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BASERUN_SB), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BASERUN_SB), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BASERUN_SB)) +
    geom_boxplot(aes(color=TEAM_BASERUN_SB), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BASERUN_SB \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BASERUN_SB, moneyball$TARGET_WINS, color=TEAM_BASERUN_SB)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BASERUN_SB", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
```

```
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_BASERUN_CS)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BASERUN_CS \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_BASERUN_CS), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_BASERUN_CS), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_BASERUN_CS)) +
    geom_boxplot(aes(color=TEAM_BASERUN_CS), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_BASERUN_CS \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_BASERUN_CS, moneyball$TARGET_WINS, color=TEAM_BASERUN_CS)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_BASERUN_CS", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

# Descriptive and Graphic Statistics of Fielding variables
kable(t(basicStats(moneyball[16:17])), digits=2, nsmall=2, caption = "Descriptive and Graphic Statistics of
Fielding variables") %>%
  kable_styling(bootstrap_options = c("striped","condensed","responsive", "hover", "bordered"),font_size =
9, full_width = F, position = "center")%>%
column_spec(3, width = "1cm")

grid.arrange(
  ggplot (moneyball, aes(TEAM_FIELDING_E)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_FIELDING_E \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_FIELDING_E), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_FIELDING_E), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_FIELDING_E)) +
    geom_boxplot(aes(color=TEAM_FIELDING_E), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_FIELDING_E \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_FIELDING_E, moneyball$TARGET_WINS, color=TEAM_FIELDING_E)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_FIELDING_E", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_FIELDING_DP)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_FIELDING_DP \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_FIELDING_DP), color="red") +
```

```
    geom_vline(xintercept=median(moneyball$TEAM_FIELDING_DP), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_FIELDING_DP)) +
    geom_boxplot(aes(color=TEAM_FIELDING_DP), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_FIELDING_DP \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_FIELDING_DP, moneyball$TARGET_WINS, color=TEAM_FIELDING_DP)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_FIELDING_DP", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

# Descriptive and Graphic Statistics of Pitching variables
kable(t(basicStats(moneyball[c(14,12,13,15)])),digits=2, nsmall=2, caption = "Descriptive and Graphic
Statistics of Pitching variables") %>%
  kable_styling(bootstrap_options = c("striped","condensed","responsive", "hover", "bordered"),font_size =
9, full_width = F, position = "center")%>%
column_spec(3, width = "1cm")

grid.arrange(
  ggplot (moneyball, aes(TEAM_PITCHING_BB)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_PITCHING_BB \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_PITCHING_BB), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_PITCHING_BB), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_PITCHING_BB)) +
    geom_boxplot(aes(color=TEAM_PITCHING_BB), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_PITCHING_BB \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_PITCHING_BB, moneyball$TARGET_WINS, color=TEAM_PITCHING_BB)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_PITCHING_BB", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_PITCHING_H)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_PITCHING_H \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_PITCHING_H), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_PITCHING_H), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_PITCHING_H)) +
    geom_boxplot(aes(color=TEAM_PITCHING_H), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_PITCHING_H \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
```

```
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_PITCHING_H, moneyball$TARGET_WINS, color=TEAM_PITCHING_H)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_PITCHING_H", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_PITCHING_HR)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_PITCHING_HR \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_PITCHING_HR), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_PITCHING_HR), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_PITCHING_HR)) +
    geom_boxplot(aes(color=TEAM_PITCHING_HR), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_PITCHING_HR \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_PITCHING_HR, moneyball$TARGET_WINS, color=TEAM_PITCHING_HR)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_PITCHING_HR", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)

grid.arrange(
  ggplot (moneyball, aes(TEAM_PITCHING_SO)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_PITCHING_SO \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball$TEAM_PITCHING_SO), color="red") +
    geom_vline(xintercept=median(moneyball$TEAM_PITCHING_SO), color="mediumblue", linetype="dashed"),
  ggplot(moneyball, aes(x="", y=moneyball$TEAM_PITCHING_SO)) +
    geom_boxplot(aes(color=TEAM_PITCHING_SO), outlier.colour="indianred", outlier.shape=19,
outlier.alpha=0.2, outlier.size=2) +
    geom_jitter(width=0.1, shape=19, alpha=0.30, color="gray50") +
    ggtitle("Boxplot") +
    labs(x="Distribution", y="TEAM_PITCHING_SO \n Distribution") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7, hjust=1)) +
    theme(legend.position="none"),
  ggplot(moneyball, aes(moneyball$TEAM_PITCHING_SO, moneyball$TARGET_WINS, color=TEAM_PITCHING_SO)) +
    geom_point(color="indianred", width=0.1, shape=19, alpha=0.2) +
    geom_smooth(method="lm", color="gray50") +
    geom_rug(color="indianred") +
    ggtitle("Scatter Plot") +
    labs(x="TEAM_PITCHING_SO", y="TARGET_WINS") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    theme(axis.text.x = element_text(size=7)) +
    theme(legend.title=element_text(size=7)) +
    theme(legend.text=element_text(size=6)) +
    stat_ellipse(color="gray50"),
    ncol=3)
```

```r
# Histogram of Correlations with TARGET_WINS
library(corrplot)
mcor <- cor(moneyball[2:17])
cor_WINS <- data.frame(col = as.character(colnames(moneyball[c(2,3,4,7,13,6,5,14,12,16)])),
                    s = sort(mcor[,c("TARGET_WINS")], decreasing = TRUE))



ggplot(cor_WINS, aes(x = col, y = s, label_value()))+
  geom_bar(stat = "identity", fill=I("salmon"), alpha=0.6, color=I("gray60"))+
  coord_flip(ylim = c(-1, 1))+
  labs(title = "Histogram of Correlations with TARGET_WINS", x = element_blank(), y = "Correlation
Coefficient")+
  theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
  theme(axis.text = element_text(size=7)) +
  theme(axis.title = element_text(size=8)) +
  geom_hline(yintercept = 0, color="gray50")

# Pair-wise scatterplot of correlations between Moneyball variables
chart.Correlation(moneyball[,2:17], na.rm = TRUE, use = "everything", histogram=TRUE, pch=".", title =
"Pairwise scatterplot of Correlations between Moneyball Variables", cex.main=0.9) + ggplot2::labs(title =
"Pairwise scatterplot of Correlations between Moneyball Variables")

# Heatmap Matrix for Moneyball Variables
mcor <- cor(moneyball[,2:length(names(moneyball))])
corrplot(mcor, method="square", tl.col="black",tl.cex=0.6, label_alpha = TRUE)



################################### Section 2: Data Preparation

# Identification of missing values
library(mice)
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(moneyball,2,pMiss)

# Plot patterns of missing values
library(VIM)
aggr_plot <- aggr(moneyball, col=c("gray90","salmon"), border="white", space = 0.8, numbers=TRUE,
sortVars=TRUE, labels=names(moneyball), ylim =c(0,1), cex.axis=0.4, cex.names=0.6, cex.lab = 0.6,
cex.main=0.6, cex=0.6, gap=2, ylab=c("Histogram on Missing Data Proportions","Pattern"))

# Initiate missing-data imputations
library(mice)
init = mice(moneyball, maxit=0)
meth = init$method
predM = init$predictorMatrix

# Removal of INDEX as predictor
predM[, c("INDEX")]=0

# 1st. options for missing-data imputation
method1 <- mice(moneyball, m=5, maxit=5, meth="rf", predictorMatrix=predM, seed=500)

# 2nd. options for missing-data imputation
method2 <- mice(moneyball, m=5, maxit=5, meth="cart", predictorMatrix=predM, seed=500)

# Imputation of the estimated density and comparison
par(mfrow=c(1,3))
densityplot(method1)

# Completed data matrix after imputations
moneyball1 <- complete(method1,1)

# Flag variables with imputation as IMP_X:
names(moneyball1)[names(moneyball1)=="TEAM_BATTING_HBP"] <- "IMP_TEAM_BATTING_HBP"
names(moneyball1)[names(moneyball1)=="TEAM_BASERUN_CS"] <- "IMP_TEAM_BASERUN_CS"
names(moneyball1)[names(moneyball1)=="TEAM_FIELDING_DP"] <- "IMP_TEAM_FIELDING_DP"
names(moneyball1)[names(moneyball1)=="TEAM_BASERUN_SB"] <- "IMP_TEAM_BASERUN_SB"
names(moneyball1)[names(moneyball1)=="TEAM_BATTING_SO"] <- "IMP_TEAM_BATTING_SO"
names(moneyball1)[names(moneyball1)=="TEAM_PITCHING_SO"] <- "IMP_TEAM_PITCHING_SO"

# Flag variables with missing values M_X:
moneyball1$X_TEAM_BATTING_HBP <- ifelse(is.na(moneyball$TEAM_BATTING_HBP), 1, 0)
moneyball1$X_TEAM_BASERUN_CS <- ifelse(is.na(moneyball$TEAM_BASERUN_CS), 1, 0)
moneyball1$X_TEAM_FIELDING_DP <- ifelse(is.na(moneyball$TEAM_FIELDING_DP), 1, 0)
moneyball1$X_TEAM_BASERUN_SB <- ifelse(is.na(moneyball$TEAM_BASERUN_SB), 1, 0)
moneyball1$X_TEAM_BATTING_SO <- ifelse(is.na(moneyball$TEAM_BATTING_SO), 1, 0)
moneyball1$X_TEAM_PITCHING_SO <- ifelse(is.na(moneyball$TEAM_PITCHING_SO), 1, 0)
```

```
# Comparison between the original data and imputed data
stripplot(method1)

# Variable elimination
moneyball1 <- subset(moneyball1, select = -IMP_TEAM_BATTING_HBP)
moneyball1 <- subset(moneyball1, select = -X_TEAM_BATTING_HBP)



# Histograms of imputed variables
grid.arrange(
  ggplot (moneyball1, aes(IMP_TEAM_BATTING_SO)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="IMP_TEAM_BATTING_SO \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$IMP_TEAM_BATTING_SO), color="red") +
    geom_vline(xintercept=median(moneyball1$IMP_TEAM_BATTING_SO), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(IMP_TEAM_BASERUN_SB)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="IMP_TEAM_BASERUN_SB \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$IMP_TEAM_BASERUN_SB), color="red") +
    geom_vline(xintercept=median(moneyball1$IMP_TEAM_BASERUN_SB), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(IMP_TEAM_BASERUN_CS)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="IMP_TEAM_BASERUN_CS \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$IMP_TEAM_BASERUN_CS), color="red") +
    geom_vline(xintercept=median(moneyball1$IMP_TEAM_BASERUN_CS), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(IMP_TEAM_PITCHING_SO)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="IMP_TEAM_PITCHING_SO \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$IMP_TEAM_PITCHING_SO), color="red") +
    geom_vline(xintercept=median(moneyball1$IMP_TEAM_PITCHING_SO), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(IMP_TEAM_FIELDING_DP)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="IMP_TEAM_FIELDING_DP \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$IMP_TEAM_FIELDING_DP), color="red") +
    geom_vline(xintercept=median(moneyball1$IMP_TEAM_FIELDING_DP), color="mediumblue", linetype="dashed"),
    ncol=3)

# Trimming to 1st or 5th percentiles on variables with zeros or unlikely lower values
quantile(moneyball1$TEAM_BATTING_3B, c(0.01, 0.05))
quantile(moneyball1$TEAM_BATTING_HR, c(0.01, 0.05))
quantile(moneyball1$TEAM_BATTING_BB, c(0.01, 0.05))
quantile(moneyball1$IMP_TEAM_BATTING_SO, c(0.01, 0.05))
quantile(moneyball1$IMP_TEAM_BASERUN_SB, c(0.01, 0.05))
quantile(moneyball1$IMP_TEAM_BASERUN_CS, c(0.01, 0.05))
quantile(moneyball1$TEAM_PITCHING_BB, c(0.01, 0.05))
quantile(moneyball1$TEAM_PITCHING_HR, c(0.01, 0.05))
quantile(moneyball1$IMP_TEAM_PITCHING_SO, c(0.01, 0.05))

moneyball1$TEAM_BATTING_3B[(moneyball1$TEAM_BATTING_3B < quantile(moneyball1$TEAM_BATTING_3B, 0.05))] =
quantile(moneyball1$TEAM_BATTING_3B, 0.05)
moneyball1$TEAM_BATTING_HR[(moneyball1$TEAM_BATTING_HR < quantile(moneyball1$TEAM_BATTING_HR, 0.05))] =
quantile(moneyball1$TEAM_BATTING_HR, 0.05)
moneyball1$TEAM_BATTING_BB[(moneyball1$TEAM_BATTING_BB < quantile(moneyball1$TEAM_BATTING_BB, 0.05))] =
quantile(moneyball1$TEAM_BATTING_BB, 0.05)
moneyball1$IMP_TEAM_BATTING_SO[(moneyball1$IMP_TEAM_BATTING_SO < quantile(moneyball1$IMP_TEAM_BATTING_SO,
0.05))] = quantile(moneyball1$IMP_TEAM_BATTING_SO, 0.05)
moneyball1$IMP_TEAM_BASERUN_SB[(moneyball1$IMP_TEAM_BASERUN_SB < quantile(moneyball1$IMP_TEAM_BASERUN_SB,
0.05))] = quantile(moneyball1$IMP_TEAM_BASERUN_SB, 0.05)
moneyball1$IMP_TEAM_BASERUN_CS[(moneyball1$IMP_TEAM_BASERUN_CS < quantile(moneyball1$IMP_TEAM_BASERUN_CS,
0.05))] = quantile(moneyball1$IMP_TEAM_BASERUN_CS, 0.05)
moneyball1$TEAM_PITCHING_BB[(moneyball1$TEAM_PITCHING_BB < quantile(moneyball1$TEAM_PITCHING_BB, 0.01))] =
quantile(moneyball1$TEAM_PITCHING_BB, 0.01)
moneyball1$TEAM_PITCHING_HR[(moneyball1$TEAM_PITCHING_HR < quantile(moneyball1$TEAM_PITCHING_HR, 0.05))] =
quantile(moneyball1$TEAM_PITCHING_HR, 0.05)
moneyball1$IMP_TEAM_PITCHING_SO[(moneyball1$IMP_TEAM_PITCHING_SO < quantile(moneyball1$IMP_TEAM_PITCHING_SO,
0.01))] = quantile(moneyball1$IMP_TEAM_PITCHING_SO, 0.01)

# Trimming to 95th or 99th percentiles on variables with unrealistic higher values after the limit indicated
quantile(moneyball1$TEAM_BATTING_3B, c(0.95, 0.99))
```

```
quantile(moneyball1$IMP_TEAM_BASERUN_SB, c(0.95, 0.99))
quantile(moneyball1$IMP_TEAM_BASERUN_CS, c(0.95, 0.99))
quantile(moneyball1$TEAM_FIELDING_E, c(0.95, 0.99))
quantile(moneyball1$TEAM_PITCHING_BB, c(0.95, 0.99))
quantile(moneyball1$TEAM_PITCHING_H, c(0.95, 0.99))
quantile(moneyball1$IMP_TEAM_PITCHING_SO, c(0.95, 0.99))

moneyball1$ TEAM_BATTING_3B[(moneyball1$ TEAM_BATTING_3B > quantile(moneyball1$ TEAM_BATTING_3B, 0.99))] =
quantile(moneyball1$ TEAM_BATTING_3B, 0.99)
moneyball1$ IMP_TEAM_BASERUN_SB[(moneyball1$ IMP_TEAM_BASERUN_SB > quantile(moneyball1$ IMP_TEAM_BASERUN_SB,
0.99))] = quantile(moneyball1$ IMP_TEAM_BASERUN_SB, 0.99)
moneyball1$ IMP_TEAM_BASERUN_CS[(moneyball1$ IMP_TEAM_BASERUN_CS > quantile(moneyball1$ IMP_TEAM_BASERUN_CS,
0.99))] = quantile(moneyball1$ IMP_TEAM_BASERUN_CS, 0.99)
moneyball1$TEAM_FIELDING_E[(moneyball1$TEAM_FIELDING_E > 500)] = 500
moneyball1$TEAM_PITCHING_BB[(moneyball1$TEAM_PITCHING_BB > quantile(moneyball1$TEAM_PITCHING_BB, 0.99))] =
quantile(moneyball1$TEAM_PITCHING_BB, 0.99)
moneyball1$TEAM_PITCHING_H[(moneyball1$TEAM_PITCHING_H > 5000)] = 5000
moneyball1$IMP_TEAM_PITCHING_SO[(moneyball1$IMP_TEAM_PITCHING_SO > quantile(moneyball1$IMP_TEAM_PITCHING_SO,
0.99))] = quantile(moneyball1$IMP_TEAM_PITCHING_SO, 0.99)

# Histograms of variables before and after square-root transformation
grid.arrange(
  ggplot (moneyball1, aes(TEAM_BATTING_H)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_H \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$TEAM_BATTING_H), color="red") +
    geom_vline(xintercept=median(moneyball1$TEAM_BATTING_H), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(sqrt(moneyball1$TEAM_BATTING_H))) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="Sqrt_TEAM_BATTING_H \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(sqrt(moneyball1$TEAM_BATTING_H)), color="red") +
    geom_vline(xintercept=median(sqrt(moneyball1$TEAM_BATTING_H)), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(TEAM_PITCHING_HR)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_PITCHING_HR \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$TEAM_PITCHING_HR), color="red") +
    geom_vline(xintercept=median(moneyball1$TEAM_PITCHING_HR), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(sqrt(moneyball1$TEAM_PITCHING_HR))) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="Sqrt_TEAM_PITCHING_HR \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(sqrt(moneyball1$TEAM_PITCHING_HR)), color="red") +
    geom_vline(xintercept=median(sqrt(moneyball1$TEAM_PITCHING_HR)), color="mediumblue", linetype="dashed"),
    ncol=2)

# Square-root transformation
moneyball1$Sqrt_TEAM_BATTING_H <- sqrt(moneyball1$TEAM_BATTING_H)
moneyball1$Sqrt_TEAM_PITCHING_HR <- sqrt(moneyball1$TEAM_PITCHING_HR)

# Histograms of variables before and after log transformation
grid.arrange(
  ggplot (moneyball1, aes(TEAM_BATTING_3B)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="TEAM_BATTING_3B \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$TEAM_BATTING_3B), color="red") +
    geom_vline(xintercept=median(moneyball1$TEAM_BATTING_3B), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(log(moneyball1$TEAM_BATTING_3B))) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="Log_TEAM_BATTING_3B \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(log(moneyball1$TEAM_BATTING_3B)), color="red") +
    geom_vline(xintercept=median(log(moneyball1$TEAM_BATTING_3B)), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(IMP_TEAM_BASERUN_SB)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="IMP_TEAM_BASERUN_SB \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$IMP_TEAM_BASERUN_SB), color="red") +
    geom_vline(xintercept=median(moneyball1$IMP_TEAM_BASERUN_SB), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(log(moneyball1$IMP_TEAM_BASERUN_SB))) +
```

```
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="Log_IMP_TEAM_BASERUN_SB \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(log(moneyball1$IMP_TEAM_BASERUN_SB)), color="red") +
    geom_vline(xintercept=median(log(moneyball1$IMP_TEAM_BASERUN_SB)), color="mediumblue",
linetype="dashed"),
  ggplot (moneyball1, aes(IMP_TEAM_BASERUN_CS)) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="IMP_TEAM_BASERUN_CS \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(moneyball1$IMP_TEAM_BASERUN_CS), color="red") +
    geom_vline(xintercept=median(moneyball1$IMP_TEAM_BASERUN_CS), color="mediumblue", linetype="dashed"),
  ggplot (moneyball1, aes(log(moneyball1$IMP_TEAM_BASERUN_CS))) +
    geom_histogram (fill=I("lightgrey"), color=I("gray60")) +
    ggtitle("Histogram") + labs(x="Distribution", y="Log_IMP_TEAM_BASERUN_CS \n Count") +
    theme(plot.title=element_text(hjust=0.5, size=12, color="gray50")) +
    theme(axis.title=element_text(size=8)) +
    geom_vline(xintercept=mean(log(moneyball1$IMP_TEAM_BASERUN_CS)), color="red") +
    geom_vline(xintercept=median(log(moneyball1$IMP_TEAM_BASERUN_CS)), color="mediumblue",
linetype="dashed"),
    ncol=2)

# Log transformation
moneyball1$Log_TEAM_BATTING_3B <- log(moneyball1$TEAM_BATTING_3B)
moneyball1$Log_IMP_TEAM_BASERUN_SB <- log(moneyball1$IMP_TEAM_BASERUN_SB)
moneyball1$Log_IMP_TEAM_BASERUN_CS <- log(moneyball1$IMP_TEAM_BASERUN_CS)

# Variable addition
# Singles by batters (TEAM_BATTING_1B)
moneyball1$TEAM_BATTING_1B <- moneyball1$TEAM_BATTING_H - moneyball1$TEAM_BATTING_HR -
moneyball1$TEAM_BATTING_3B - moneyball1$TEAM_BATTING_2B

# Ratio of stealing bases (TEAM_SB_PTC)
moneyball1$TEAM_SB_PCT =
moneyball1$IMP_TEAM_BASERUN_SB/(1.0*moneyball1$IMP_TEAM_BASERUN_SB+moneyball1$IMP_TEAM_BASERUN_CS)
moneyball1$TEAM_SB_PCT[is.na(moneyball1$TEAM_SB_PCT)] = mean(moneyball1$TEAM_SB_PCT, na.rm = TRUE)

# Ratio of walks versus strikeouts by batters (TEAM_BATTING_BBSO)
moneyball1$TEAM_BATTING_BBSO <- moneyball1$TEAM_BATTING_BB/moneyball1$IMP_TEAM_BATTING_SO

# Ratio of walks allowed versus strikeouts by pitchers (TEAM_PITCHING_BBSO)
moneyball1$TEAM_PITCHING_BBSO <- moneyball1$TEAM_PITCHING_BB/moneyball1$IMP_TEAM_PITCHING_SO

# Check completeness of the data
summary(moneyball1)

# Heatmap Matrices for Moneyball Variables before and after transformations
par(mfrow=c(1,2))
mcor <- cor(moneyball[,2:length(names(moneyball))])
corrplot(mcor, method="square", tl.col="black",tl.cex=0.6)

mcor2 <- cor(moneyball1[,2:30])
corrplot(mcor2, method="square", tl.col="black",tl.cex=0.6)



################################### Section 3: Model Building

library(MASS)
# Forward AVS model
Forward_model <- stepAIC(lm(formula = TARGET_WINS ~ Sqrt_TEAM_BATTING_H + TEAM_BATTING_1B +
                              TEAM_BATTING_2B + Log_TEAM_BATTING_3B + TEAM_BATTING_HR +
                              TEAM_BATTING_BB + IMP_TEAM_BATTING_SO + Log_IMP_TEAM_BASERUN_SB +
                              Log_IMP_TEAM_BASERUN_CS + TEAM_FIELDING_E +
                              IMP_TEAM_FIELDING_DP + TEAM_PITCHING_BB + TEAM_PITCHING_H +
                              Sqrt_TEAM_PITCHING_HR + IMP_TEAM_PITCHING_SO + X_TEAM_BASERUN_CS +
                              X_TEAM_FIELDING_DP + X_TEAM_BASERUN_SB + X_TEAM_BATTING_SO +
                              X_TEAM_PITCHING_SO + TEAM_SB_PCT + TEAM_BATTING_BBSO +
                              TEAM_PITCHING_BBSO, data = moneyball1), direction = "forward")
anova(Forward_model)
summary(Forward_model)
# Forward AVS model revised
Forward_modelR <- stepAIC(lm(formula = TARGET_WINS ~ Sqrt_TEAM_BATTING_H + TEAM_BATTING_2B +
                               Log_TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB +
                               IMP_TEAM_BATTING_SO + Log_IMP_TEAM_BASERUN_SB +
                               Log_IMP_TEAM_BASERUN_CS + TEAM_FIELDING_E + IMP_TEAM_FIELDING_DP +
                               TEAM_PITCHING_BB + TEAM_PITCHING_H + Sqrt_TEAM_PITCHING_HR +
                               IMP_TEAM_PITCHING_SO + X_TEAM_BASERUN_CS + X_TEAM_FIELDING_DP +
```

```
                                X_TEAM_BASERUN_SB + X_TEAM_BATTING_SO + TEAM_SB_PCT +
                                TEAM_BATTING_BBSO + TEAM_PITCHING_BBSO, data = moneyball1),
                            direction = "forward")
anova(Forward_modelR)
summary(Forward_modelR)
# VIF values of the Forward AVS model
vif(Forward_modelR)
# Forward AVS model goodness of fit plots
par(mfrow=c(2,2))
plot(Forward_modelR)


# Stepwise AVS model
Stepwise_model <- stepAIC(lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + Sqrt_TEAM_BATTING_H +
                                TEAM_BATTING_2B + Log_TEAM_BATTING_3B + TEAM_BATTING_HR +
                                TEAM_BATTING_BB + IMP_TEAM_BATTING_SO + Log_IMP_TEAM_BASERUN_SB +
                                Log_IMP_TEAM_BASERUN_CS + TEAM_FIELDING_E + IMP_TEAM_FIELDING_DP +
                                TEAM_PITCHING_BB + TEAM_PITCHING_H + Sqrt_TEAM_PITCHING_HR +
                                IMP_TEAM_PITCHING_SO + X_TEAM_BASERUN_CS + X_TEAM_FIELDING_DP +
                                X_TEAM_BASERUN_SB + X_TEAM_BATTING_SO + X_TEAM_PITCHING_SO +
                                TEAM_SB_PCT + TEAM_BATTING_BBSO + TEAM_PITCHING_BBSO,
                              data = moneyball1), direction = "both")
anova(Stepwise_model)
summary(Stepwise_model)
# VIF values of the Stepwise AVS model
vif(Stepwise_model)
# Stepwise AVS model goodness of fit plots
par(mfrow=c(2,2))
plot(Stepwise_model)

# Selected Variables model
SV_model <- lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + Sqrt_TEAM_BATTING_H + TEAM_BATTING_2B +
                 Log_TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO +
                 Log_IMP_TEAM_BASERUN_SB + TEAM_FIELDING_E + TEAM_PITCHING_BB + TEAM_PITCHING_H +
                 IMP_TEAM_PITCHING_SO + X_TEAM_BASERUN_CS + X_TEAM_FIELDING_DP +
                 X_TEAM_BASERUN_SB + X_TEAM_BATTING_SO + TEAM_SB_PCT + TEAM_BATTING_BBSO +
                 TEAM_PITCHING_BBSO, data = moneyball1)
anova(SV_model)
summary(SV_model)
# VIF values of the Selected variables model
vif(SV_model)
# Selected variables model goodness of fit plots
par(mfrow=c(2,2))
plot(SV_model)

# PCA model
library(FactoMineR)
pca_moneyball1 = PCA(moneyball1, graph = FALSE)
# Matrix with eigenvalues
pca_moneyball1$eig
# Correlations between variables and PCs
pca_moneyball1$var$coord
# PCA model components and dimensions
pc = prcomp(moneyball1)
dim(pc$rotation)
dim(pc$x)
# PCA Model building
dim(pc$rotation)
dim(pc$x)
PCA_model <- lm(moneyball1$TARGET_WINS~pc$x[,1]+pc$x[,2]+pc$x[,3]+pc$x[,4]+pc$x[,5]+pc$x[,6]+
                  pc$x[,7]+pc$x[,8]+pc$x[,9]+pc$x[,10]+pc$x[,11]+pc$x[,12]+pc$x[,13])
anova(PCA_model)
summary(PCA_model)
# PCA model goodness of fit graphs
par(mfrow=c(2,2))
plot(PCA_model)

# Random Forest regression model
require(randomForest)
set.seed(101)
rf_model=randomForest(formula = TARGET_WINS ~ TEAM_BATTING_1B + Sqrt_TEAM_BATTING_H +
                        TEAM_BATTING_2B + Log_TEAM_BATTING_3B + TEAM_BATTING_HR +
                        TEAM_BATTING_BB + IMP_TEAM_BATTING_SO + Log_IMP_TEAM_BASERUN_SB +
                        Log_IMP_TEAM_BASERUN_CS + TEAM_FIELDING_E + IMP_TEAM_FIELDING_DP +
                        TEAM_PITCHING_BB + TEAM_PITCHING_H + Sqrt_TEAM_PITCHING_HR +
                        IMP_TEAM_PITCHING_SO + X_TEAM_BASERUN_CS + X_TEAM_FIELDING_DP +
                        X_TEAM_BASERUN_SB + X_TEAM_BATTING_SO + X_TEAM_PITCHING_SO +
                        TEAM_SB_PCT + TEAM_BATTING_BBSO + TEAM_PITCHING_BBSO,
                      data = moneyball1, ntree = 500, importance = TRUE)
```

```
rf_model
rf_model$importance
# Random Forest regression model goodness of fit plots
par(mfrow=c(1,2))
varImpPlot(rf_model, sort=TRUE, main = "Importance of Variables",pch=16,cex=0.5, col='gray50')
plot(rf_model, main='Error rate of random forest',col='salmon',cex=0.2)




################################### Section 4: Model Selection

# Adjusted R-squared
print("Adjusted R2 (Forward_modelR) = 0.3991 ~ 39.91% of the variation explained by 21 variables")
print("Adjusted R2 (Stepwise_model) = 0.4 ~ 40.00% of the variation explained by 17 variables")
print("Adjusted R2 (SV_model) = 0.382 ~ 38.20% of the variation explained by 19 variables")
print("Adjusted R2 (PCA_model) = 0.3827 ~ 38.27% of the variation explained by 13 PC or variables")
print("Adjusted R2 (rf_model) = 41.26% of the variation explained by 23 variables, mainly two of them")

# Akaike Information Criterion (AIC)
AIC(Forward_modelR)
AIC(Stepwise_model)
AIC(SV_model)
AIC(PCA_model)

# Bayesian Information Criterion (BIC)
BIC(Forward_modelR)
BIC(Stepwise_model)
BIC(SV_model)
BIC(PCA_model)

# Mean Square Error (MSE)
MSE <- function(sm)
  mean(sm$residuals^2)
MSE(Forward_modelR)
MSE(Stepwise_model)
MSE(SV_model)
MSE(PCA_model)

# Mean Absolute Error (MAE)
library(reshape)
forward.train <- predict(Forward_modelR,newdata=moneyball1);
moneyball1$res <- moneyball1$TARGET_WINS - forward.train
moneyball1$absres <- abs(moneyball1$res)
MAE <-mean(moneyball1$absres)
MAE

stepwise.train <- predict(Stepwise_model,newdata=moneyball1);
moneyball1$res <- moneyball1$TARGET_WINS - stepwise.train
moneyball1$absres <- abs(moneyball1$res)
MAE <-mean(moneyball1$absres)
MAE

stepwise.train <- predict(SV_model,newdata=moneyball1);
moneyball1$res <- moneyball1$TARGET_WINS - stepwise.train
moneyball1$absres <- abs(moneyball1$res)
MAE <-mean(moneyball1$absres)
MAE

stepwise.train <- predict(PCA_model,newdata=moneyball1);
moneyball1$res <- moneyball1$TARGET_WINS - stepwise.train
moneyball1$absres <- abs(moneyball1$res)
MAE <-mean(moneyball1$absres)
MAE

stepwise.train <- predict(rf_model,newdata=moneyball1);
moneyball1$res <- moneyball1$TARGET_WINS - stepwise.train
moneyball1$absres <- abs(moneyball1$res)
MAE <-mean(moneyball1$absres)
MAE
```