Introduction

Context

The dataset that we will be working with is called moneyball (includes approximately 2200 records). Each record represents a professional baseball team from the years 1871 to 2006. Additionally, each record has the performance of the team for the given year, with all of the baseball statistics adjusted to match the performance of a 162 game regular season.

Objectives/Purpose

The purpose of unit 1 assignment is to analyze baseball team data using OLS ("Linear") Regression in order to predict the number of wins that a baseball team will have in a regular season. This will be accomplished by generating multiple regression models using different techniques (e.g., stepwise, etc.). From these techniques, the best model will be selected and will then be further analyzed to determine if it is an adequate model to predict baseball wins or if further analysis is necessary. First, an initial exploratory data analysis will be conducted using scatterplots, boxplots, summary statistics, etc. to help understand important characteristics and properties of the data that may be disguised by numerical summaries. Second, data preparation/transformations of the data will begin. This includes, but not limited to fixing missing values, conducting data transformations, and creating new variables. Third, we will begin building at least three different linear regression models using different variables (or the same variables with different transformations). This will be conducted by manually selecting the variables or using variable selection techniques. We will then discuss the coefficients in the model to ensure that it makes intuitive baseball sense. Fourth, we will then decide on the "best model" using metrics such as AIC, BIC, Adjusted R-Square, and MSE. Lastly, a Stand Alone scoring program will be conducted that will score the new data and predict the number of wins. The data step will include all the variable transformations such as fixing missing values and the regression formula.

Section 1: Data Exploration

Figure 1: Structure and Size of the Data

```
'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 ...
```

Observations: Figure 1 shows the structure of the data, which comes out to 2276 rows and 17 variables (integers). INDEX is not considered a true variable, while TARGET_WINS is considered our response variable, and the rest of the variables are considered our predictors.

Figure 2: Definitions of the Variables (Data Dictionary)

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

Observations: Figure 2 shows the definitions of the variables that are included in the dataset. The predictors highlighted in green denote positive impact on wins, while the predictors highlighted in red indicate negative impact on wins.

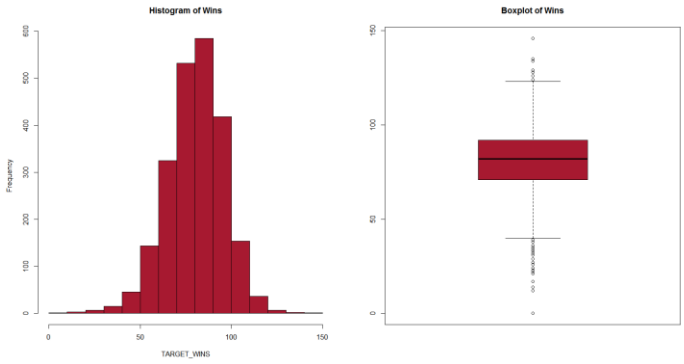
Figure 3: Data Quality Check *(see appendix)*

```
     INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B  TEAM_BATTING_HR
Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0   Min.   :  0.00   Min.   :  0.00
1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0   1st Qu.: 34.00   1st Qu.: 42.00
Median :1270.5   Median : 82.00   Median :1454   Median :238.0   Median : 47.00   Median :102.00
Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2   Mean   : 55.25   Mean   : 99.61
3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0   3rd Qu.: 72.00   3rd Qu.:147.00
Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0   Max.   :223.00   Max.   :264.00

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

Observations: Figure 3 *(also see appendix for additional data quality checks)* shows summary statistics so that we can check for missing values, outliers, etc. The data shows that the mean number of wins is 80.79, while the median number of wins is 82. The data quality check also revealed that there are missing values for 6 variables: TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_SO, and TEAM_FIELDING_DP. Furthermore, the data quality check also revealed outliers and a "wide range" of values for majority of the variables (e.g., very low or very high totals). For example, for TARGET_WINS some records have very low/unrealistic win totals such as 0, 12, 14, 17, and 21. Additionally, variables such as TEAM_PITCHING_H had records that have very high hits allowed totals such as 16038, 16871, 20088, 24057, and 30132. As we go on, we will have to investigate these outliers, missing values, and decide what to do with them (e.g., conducting imputation, etc.).

Wins

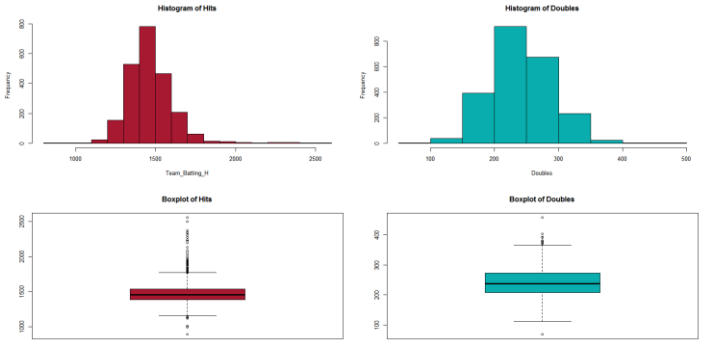
Figure 4: Histogram and Boxplot of Wins



Observations: Figure 4 shows a histogram and boxplot of wins (response variable). The histogram shows a symmetric bell shape with noticeable outliers around less than 40 wins and greater than 125 wins. Additionally, most of the values hover around the mean of 81. The box plot also shows that the median number of wins is 82 and shows the outliers that are seen in the histogram. Majority of the wins fall in-between 71 to 92.

Batting

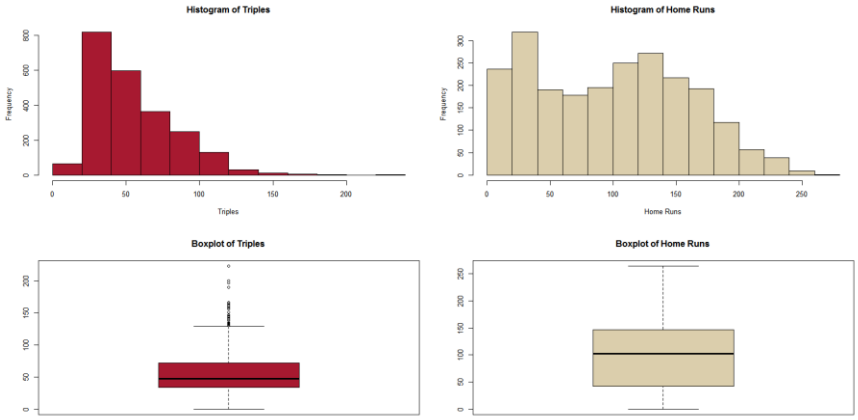
Figure 5: Histogram and Boxplot of Hits & Doubles



Observations: Figure 5 shows a histogram and boxplot of hits and doubles. The histogram of hits shows a slight right skew with noticeable outliers around less than 1110 hits and greater than 1750 hits. Additionally, most of the values hover around the mean of 1469. The box plot of hits also shows that the median number of hits is 1454 and shows the outliers that are seen in the histogram. Majority of the hits fall in-between 1383 to 1537.

The histogram of doubles shows a symmetric bell shape with some outliers around less than 100 doubles and greater than 375 doubles. Additionally, most of the values hover around the mean of 241. The box plot of doubles also shows that the median number of doubles is 238 and shows the outliers that are seen in the histogram. Majority of the doubles fall in-between 208 to 273.

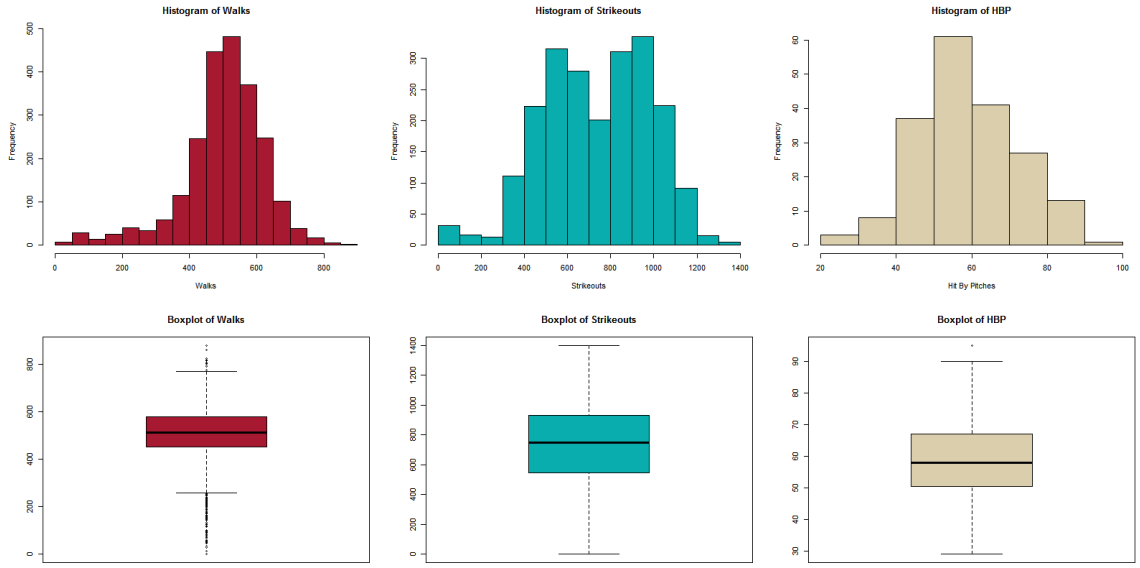
Figure 6: Histogram and Boxplot of Triples & Home Runs



Observations: Figure 6 shows a histogram and boxplot of and home runs. The histogram of triples shows a right skew with noticeable outliers around less than 20 triples and greater than 125 doubles. Additionally, most of the values hover around 30 to 55. The box plot of triples also shows that the median number of triples is 47 and shows the outliers that are seen in the histogram. Majority of the triples fall in-between 34 to 72.

The histogram of home runs shows a flat peak, slight right skew with some outliers on the right side. There is also a large amount of home runs in the tails of the histogram. For example, there are a lot of home runs between 0 to 40 and a high amount of home runs on the upper end of the histogram. This is something to investigate given that that the mean home runs are 100 and a regular season is 162 games. The box plot of home runs also shows a median number of home runs is 102. There is also a lot of variability in the number of home runs: 42 to 147.

Figure 7: Histogram and Boxplot of Walks, Strikeouts, HBP



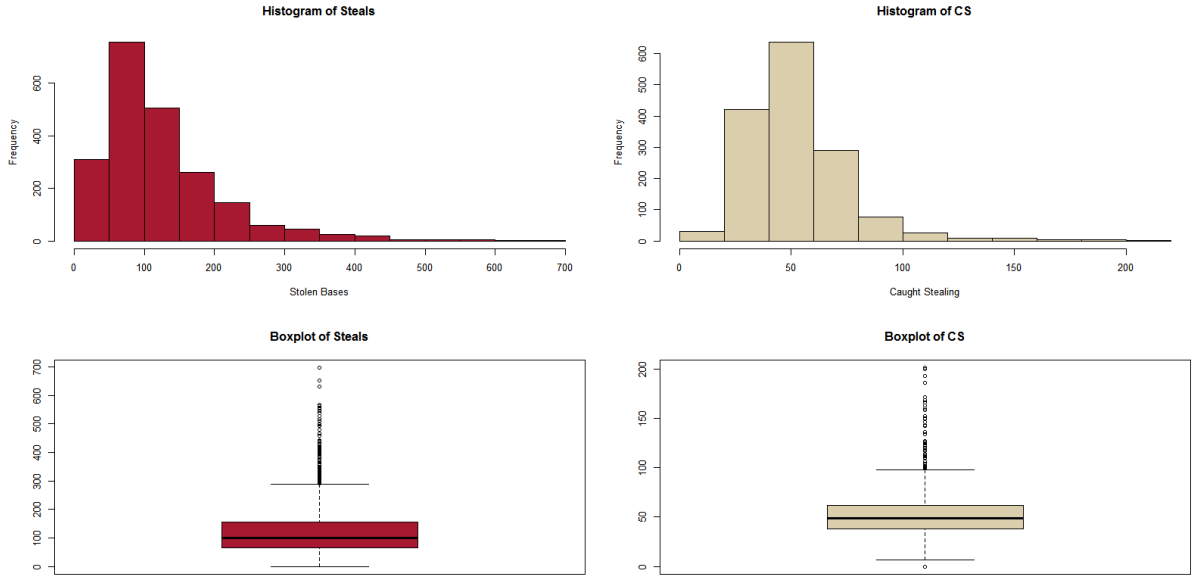
Observations: Figure 7 shows a histogram and boxplot of walks, strikeouts, and HBP. The histogram of walks show a left skew with noticeable outliers around less than 250 walks and greater than 775 walks. Additionally, most of the values hover around the mean of 502 walks. The box plot of walks also shows that the median number of walks is 512 and shows the outliers that are seen in the histogram. Majority of the walks fall in-between 451 to 580.

The histogram of strikeouts shows a bimodal pattern, with noticeable outliers towards the tails. Most of the values fall in-between 548 to 930. The box plot of strikeouts shows a median of 750 and a lot of variability.

The histogram of HBP shows a bell-shape with a few outliers around less than 25 HBP and greater than 90 HBP. Additionally, most of the values hover around the mean of 59 HBP. The box plot of HBP also shows that the median number of HBP is 58 and shows the outliers that are seen in the histogram. Majority of the HBP fall in-between 51 to 67. It's also important to note that the HBP variable is missing the most data out of all the other predictor variables.

Baserunning

Figure 8: Histogram and Boxplot of Steals and Caught Stealing

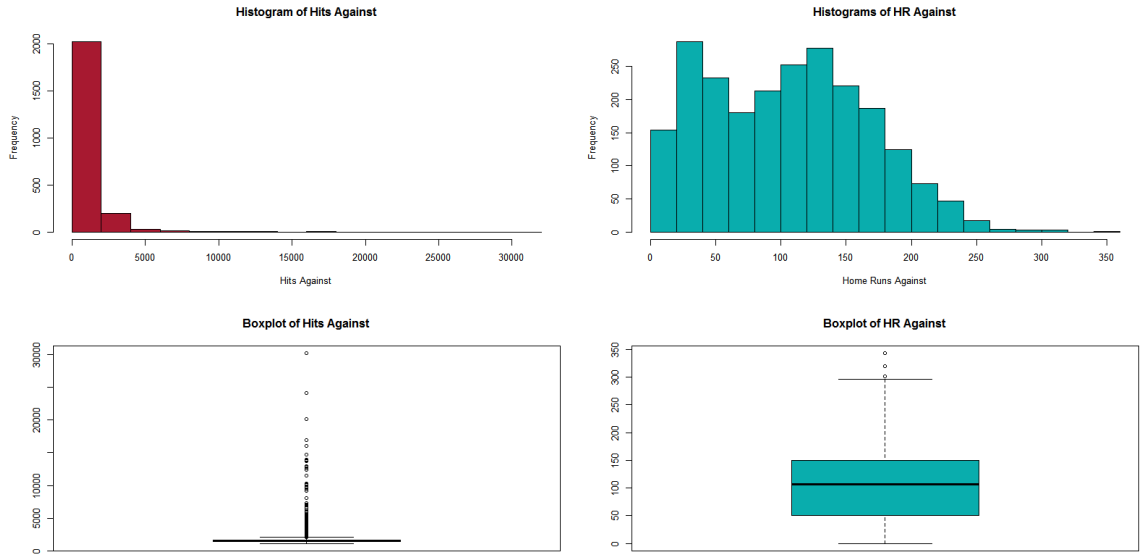


Observations: Figure 8 shows a histogram and boxplot of steals and caught stealing. The histogram of steals show a right skew with noticeable outliers around greater than 300 steals. Additionally, most of the values hover around the mean of 125 steals. The box plot of steals also shows that the median number of steals is 101 and shows the outliers that are seen in the histogram. Majority of the steals fall in-between 66 and 156.

The histogram of CS shows a right skew with noticeable outliers around less than 10 and greater than 100 CS. Additionally, most of the values hover around the mean of 53 CS. The box plot of CS also shows that the median number of CS is 49 and shows the outliers that are seen in the histogram. Majority of the CS fall in-between 38 to 62. It's also important to note that the CS variable is missing the second most data out of all the other predictor variables.

Pitching

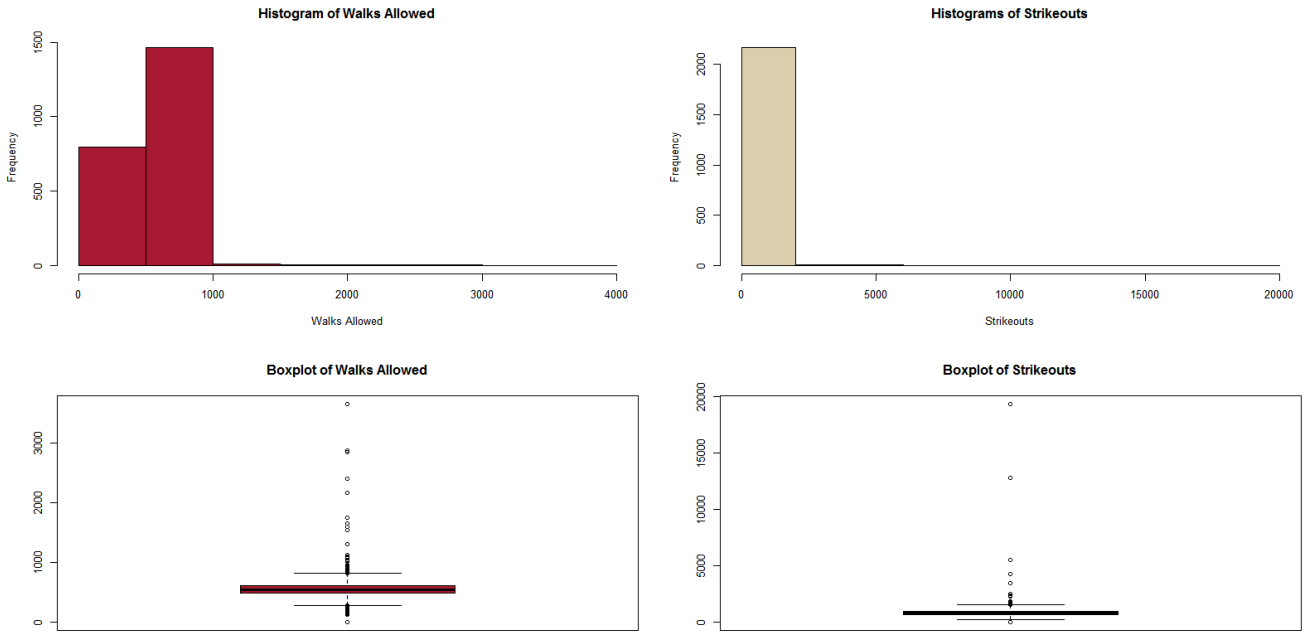
Figure 9: Histogram and Boxplot of Hits Against and HR Against



Observations: Figure 9 shows a histogram and boxplot of hits against and HR against. The histogram of hits against shows a right skew with extreme outliers (e.g., 30132) greater than 2500. Additionally, most of the values hover around the mean of 1779. The box plot of hits against also shows that the median number of hits against is 1518 and shows the extreme outliers that are seen in the histogram. Majority of the hits against fall in-between 1419 and 1682.

The histogram of HR against shows a right skew with noticeable outliers greater than 275. Additionally, most of the values hover around the mean of 106. There are also a large amount of values that fall in-between 0 to 50 HR against. Considering that the season is 160 games, this does not make intuitive baseball sense. The box plot of HR against also shows that the median number of HR against is 107 and shows the outliers that are seen in the histogram. Majority of the HR against fall in-between 50 and 150 (lots of variability).

Figure 10: Histogram and Boxplot of Walks Allowed and Strikeouts

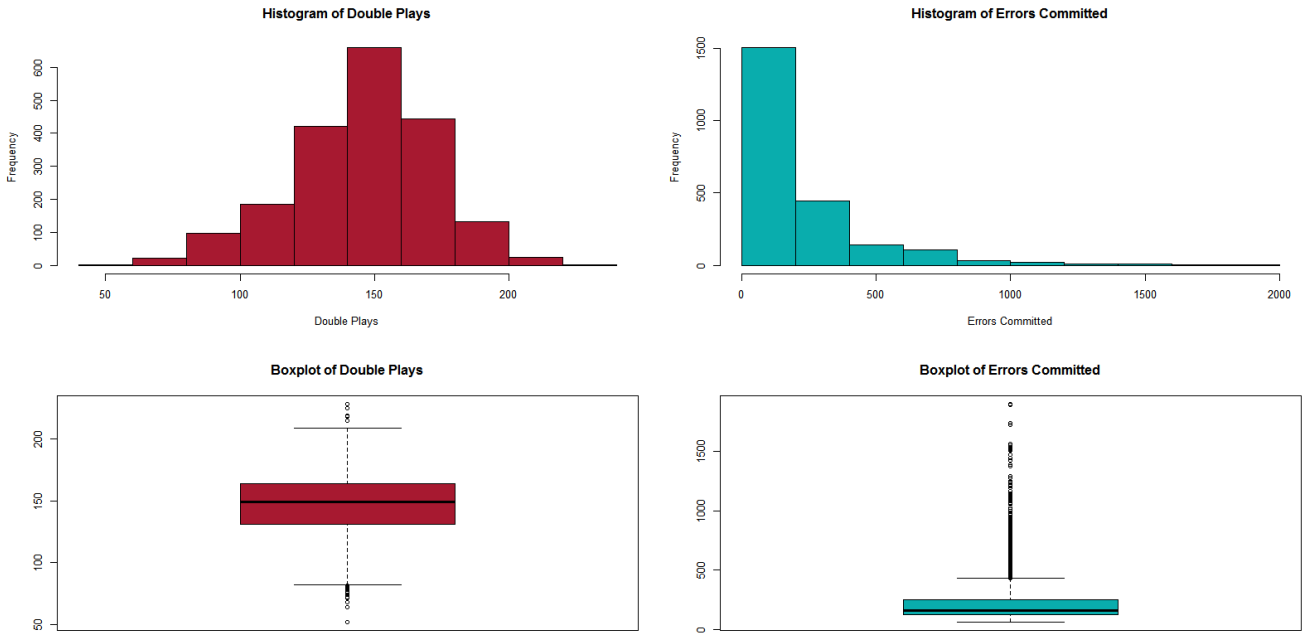


Observations: Figure 10 shows a histogram and boxplot of walks allowed and strikeouts. The histogram of walks allowed shows a lot of extreme outliers in the tails. Most of the values hover around the mean of 553. The box plot of walks allowed also shows that the median number of walks allowed is 537 and shows the extreme outliers (e.g., 3645) that are seen in the histogram. Majority of the walks allowed fall in-between 476 to 611.

The histogram of strikeouts shows very extreme outliers (e.g., 19278) as well. Most of the strikeouts hover around the mean of 818.The box plot of strikeouts also shows that the median number of strikeouts is 814 and shows the very extreme outliers that are seen in the histogram. Majority of the strikeouts fall in-between 615 and 968.

Fielding

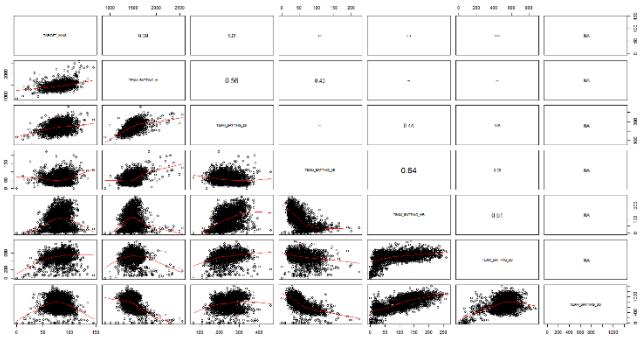
Figure 11: Histogram and Boxplot of Double Plays and Errors Committed



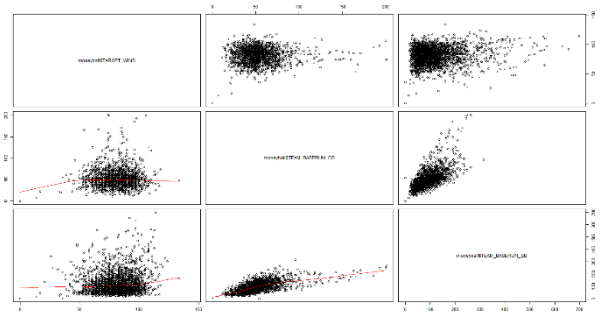
Observations: Figure 11 shows a histogram and boxplot of doubles plays and errors committed. The histogram of double plays shows slight left skew and outliers in the tails (less than 75 and greater than 225 double plays). Most of the values hover around the mean of 146. The box plot of double plays also shows that the median number of double plays is 149 and shows the outliers that are seen in the histogram. Majority of the double plays fall in-between 131 to 164.

The histogram of errors committed shows a right skew and a lot of outliers around greater than 475 errors committed. Most of the values hover around the mean of 247. The box plot of errors committed also shows that the median number of errors committed is 159 and shows the outliers that are seen in the histogram. Majority of the errors committed fall in-between 127 to 249.

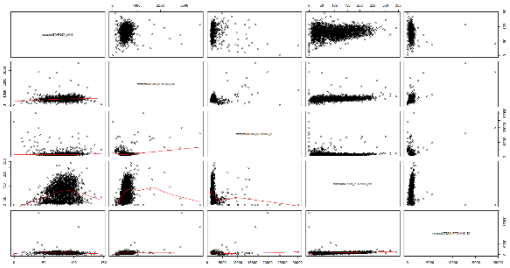
Figure 12: Scatterplot Matrices and Correlation Matrix



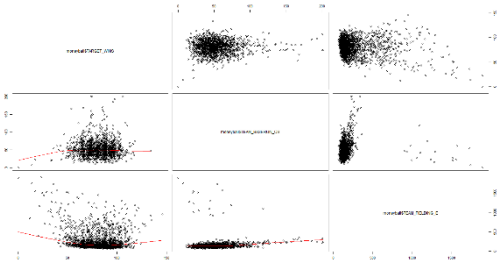
Scatterplot Matrix of Batting Stats and Wins



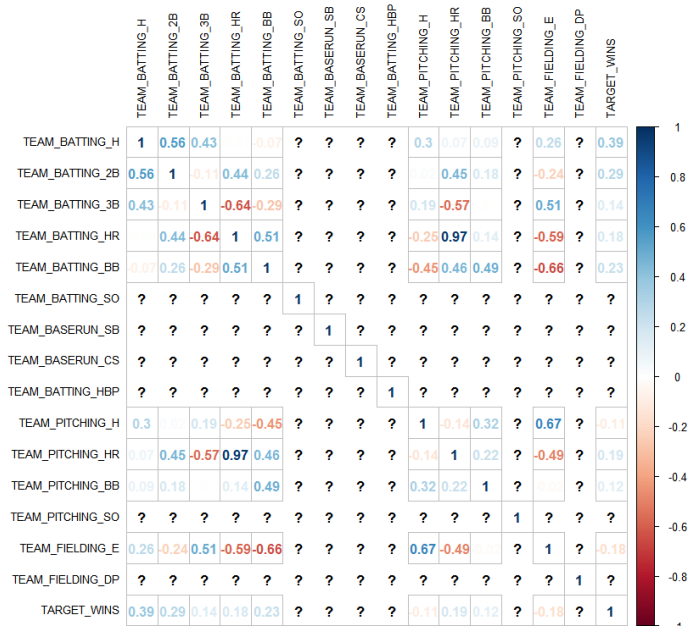
Scatterplot Matrix of Baserunning Stats and Wins



Scatterplot Matrix of Pitching Stats and Wins



Fielding Stats and Wins



Correlation Matrix of all Variables

Observations: Figure 12 shows scatterplot matrices and a correlation matrix of the variables that were included in the dataset (excluding INDEX). This gives us an idea of the most promising predictor variables based on the predictors that are most correlated with TARGET_WINS. This also allows us to see which variables may be correlated with each other (e.g., potential multicollinearity concerns).

Also, note that the correlation matrix is incomplete due to the missing values for the following 6 variables: TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_SO, and TEAM_FIELDING_DP. The scatterplot matrix for these variables also show N/A or no correlations. The scatterplot matrix and correlation matrix shows that TEAM_BATTING_H had the strongest positive correlation with TARGET_WINS, followed by TEAM_BATTING_2B, and then TEAM_BATTING_BB. TEAM_BATTING_3B and TEAM_BATTING_HR also have a weak positive correlation with TARGET_WINS. Interestingly, TEAM_PITCHING_HR and TEAM_PITCHING_BB have a weak positive correlation with TARGET_WINS. However, this does not make intuitive baseball sense considering that these should have a negative impact TARGET_WINS. This will have to be explored further. Additionally, TEAM_PITCHING_H and TEAM_FIELDING_E had moderate negatively correlated, which makes intuitive baseball sense. Lastly, the correlation matrix also revealed some potential multicollinearity concerns. For example, most of the team batting variables had high correlations with other team batting variables.

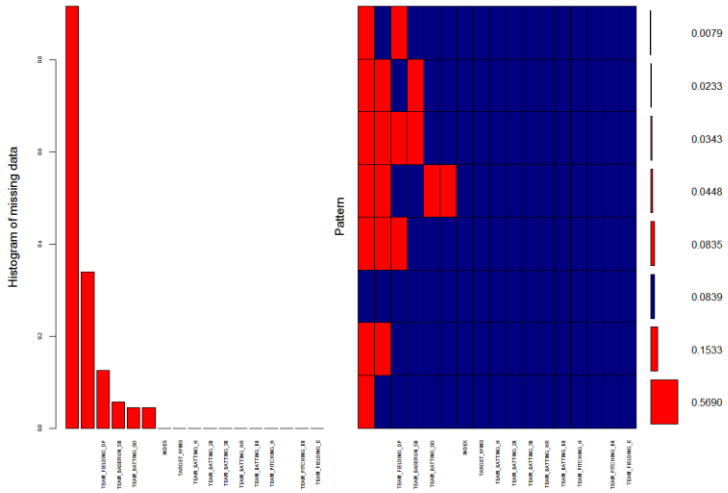
Section 2: Data Preparation

Figure 13: Missing Values for Variables

INDEX	TARGET_WINS	TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B
0	0	0	0	0
TEAM_BATTING_HR	**TEAM_BATTING_BB**	**TEAM_BATTING_SO**	**TEAM_BASERUN_SB**	**TEAM_BASERUN_CS**
0	0	102	131	772
TEAM_BATTING_HBP	**TEAM_PITCHING_H**	**TEAM_PITCHING_HR**	**TEAM_PITCHING_BB**	**TEAM_PITCHING_SO**
2085	0	0	0	102
TEAM_FIELDING_E	**TEAM_FIELDING_DP**			
0	286			

Observations: Figure 13 shows variables in the moneyball data set that have missing data. We will use the MICE package (pmm = predictive mean matching) to impute the missing data. The MICE package uses an algorithm in such a way that uses information from other variables in dataset to predict and impute the missing values. We need to address the missing values because linear regression cannot handle missing values and must be addressed prior to utilizing this modeling technique.

Figure 14: Percentage of Missing Values for Variables



INDEX	TARGET_WINS	TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B	TEAM_BATTING_HR
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
TEAM_BATTING_BB	**TEAM_BATTING_SO**	**TEAM_BASERUN_SB**	**TEAM_BASERUN_CS**	**TEAM_BATTING_HBP**	**TEAM_PITCHING_H**
0.000000	4.481547	5.755712	33.919156	91.608084	0.000000
TEAM_PITCHING_HR	**TEAM_PITCHING_BB**	**TEAM_PITCHING_SO**	**TEAM_FIELDING_E**	**TEAM_FIELDING_DP**	
0.000000	0.000000	4.481547	0.000000	12.565905	

Observations: Figure 14 shows the percentage of missing variables in the moneyball data set. Since the percentage of missing data is close to 92% for TEAM_BATTING_HBP, that variable will not be used in any of my models.

Figure 15: Summary of Imputation using Predictive Mean Matching

```
Multiply imputed data set
Call:
mice(data = moneyball2, m = 5, method = "pmm", maxit = 50, seed = 500)
Number of multiple imputations:  5
Missing cells per column:
  TEAM_BATTING_H      TARGET_WINS TEAM_BATTING_HBP  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
               0                0             2085                0                0                0                0
 TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
             102              131              772                0                0                0              102
 TEAM_FIELDING_E TEAM_FIELDING_DP
               0              286
Imputation methods:
  TEAM_BATTING_H      TARGET_WINS TEAM_BATTING_HBP  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
           "pmm"            "pmm"            "pmm"            "pmm"            "pmm"            "pmm"            "pmm"
 TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
           "pmm"            "pmm"            "pmm"            "pmm"            "pmm"            "pmm"            "pmm"
 TEAM_FIELDING_E TEAM_FIELDING_DP
           "pmm"            "pmm"


Number of N/A's:

  TEAM_BATTING_H      TARGET_WINS TEAM_BATTING_HBP  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
               0                0                0                0                0                0                0
 TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
               0                0                0                0                0                0                0
 TEAM_FIELDING_E TEAM_FIELDING_DP
               0                0
```

Observations: Figure 15 shows imputation being applied to the missing values using predictive mean matching. The result shows that all the missing values have been replaced.

Figure 16: Transformation of Variables

Discussion: TEAM_BATTING_1B and SB_PCT were new variables that were created by combining exist ing variables. Here are the formulas that I used:

- TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_HR - TEAM_BATTING_3B - TEAM_BATTING_2B
- SB_PCT = TEAM_BASERUN_SB/(1.0* TEAM_BASERUN_SB+ TEAM_BASERUN_CS)

I created these variables because TEAM_BATTING exists for doubles, triples, and HR, but not for singles. Additionally, I created SB_PCT to provide a more comprehensive metric that would tell the "full story" when it comes to stealing bases. For instance, a team could have a high amount of TEAM_BASERUN_SB, but they could also have a high amount of TEAM_BASERUN_CS. This metric shows whether a team was good at stealing bases or not. Furthermore, I also conducted log transformations on the following variables: TEAM_BATTING_1B, TEAM_BATTING_2B, TEAM_BATTING_3B, TEAM_BATTING_HR, TEAM_BATTING_BB, TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_PITCHING_BB, and TEAM_FIELDING_E, and TEAM_FIELDING_DP. I conducted log transformations on these variables since these variables were somewhat correlated to TARGET_WINS or contained missing data from the original data set. I will experiment with these transformations later on in the analysis.

Figure 17: Handling Outliers

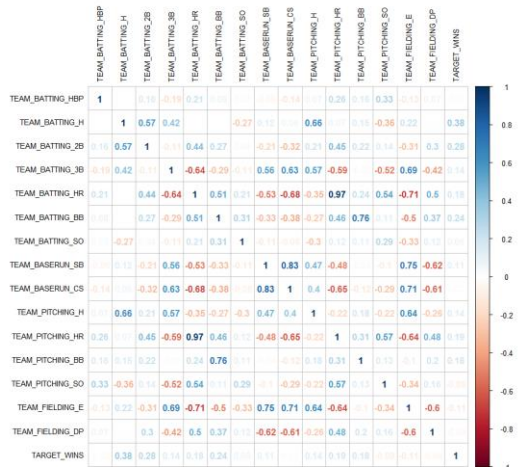
- TARGET_WINS >= 120 = 120
- TARGET_WINS <= 21 = 21
- TEAM_BATTING_H >= 2000 = 2000
- TEAM_BATTING_H <= 1000 = 1000
- TEAM_BATTING_2B >= 400 = 400
- TEAM_BATTING_2B <= 100 = 100
- TEAM_BATTING_3B >= 160 = 160
- TEAM_BATTING_3B <= 10 = 10
- TEAM_BATTING_HR <= 3 = 3
- TEAM_BATTING_BB >= 825 = 825
- TEAM_BATTING_BB <= 280 = 280
- TEAM_BATTING_SO >= 300 = 300
- TEAM_BASERUN_SB >= 350 = 350
- TEAM_BASERUN_SB <= 14 = 14
- TEAM_BASERUN_CS >= 125 = 125
- TEAM_BASERUN_CS <= 10 = 10
- TEAM_PITCHING_H >= 2000 = 2000
- TEAM_PITCHING_HR >= 260 = 260
- TEAM_PITCHING_HR <= 25 = 25
- TEAM_PITCHING_BB >= 1000 = 1000
- TEAM_PITCHING_BB <= 300 = 300
- TEAM_PITCHING_SO >= 1550 = 1550
- TEAM_PITCHING_SO <= 100 = 100
- TEAM_FIELDING_E >= 500 = 500

Discussion: Figure 17 shows how I handled the outliers (e.g., bad data) from the moneyball data set based on the EDA in section 1 (e.g., box plots, bar graphs, and summary statistics). I also conducted separate research online to give me an idea of what is considered “normal” and “not normal”. Addressing outliers is important because outliers can exert significant influence on model parameters. For instance, the model may be less accurate and the model may give a different interpretation or understanding that actually exists. Additionally, outliers can significantly impact a predictive model. For example, an outlier can cause a large difference in the coefficient or “Beta” value in a regression model. As a result, the primary technique that I used to handle the outliers was trimming the data (e.g., when a variable exceeds a certain limit, it is simply truncated so that it cannot exceed the limit).

Section 3: Build Models

Model 1A & 1B: Multiple Regression Models

Figure 18: Model MLRResult1



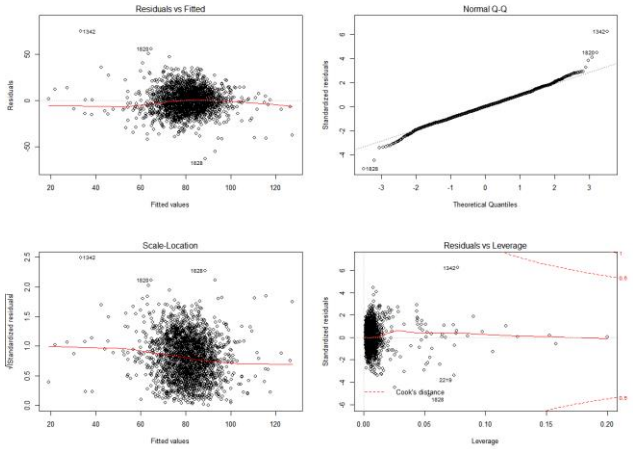

```
Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BASERUN_SB +
    TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_FIELDING_E,
    data = moneyball3)

Residuals:
    Min      1Q  Median      3Q     Max
-65.659  -8.372  -0.055   8.410  68.951

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.323285   3.629047   0.640 0.522113
TEAM_BATTING_H    0.045453   0.003722  12.211  < 2e-16 ***
TEAM_BATTING_2B  -0.028742   0.008799  -3.267 0.001105 **
TEAM_BATTING_3B   0.118547   0.016530   7.172 9.98e-13 ***
TEAM_BATTING_HR  -0.050289   0.030237  -1.663 0.096421 .
TEAM_BATTING_BB   0.041529   0.007078   5.867 5.08e-09 ***
TEAM_BASERUN_SB   0.079866   0.004817  16.580  < 2e-16 ***
TEAM_PITCHING_H   0.001329   0.002825   0.471 0.637988
TEAM_PITCHING_HR  0.081050   0.028329   2.861 0.004262 **
TEAM_PITCHING_BB -0.020739   0.005855  -3.542 0.000405 ***
TEAM_FIELDING_E  -0.064239   0.004895 -13.123  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.8 on 2265 degrees of freedom
Multiple R-squared:  0.3259,    Adjusted R-squared:  0.3229
F-statistic: 109.5 on 10 and 2265 DF,  p-value: < 2.2e-16
```

```
> vif(MLRResult1)
  TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BASERUN_SB  TEAM_PITCHING_H
        3.502208         2.338356         2.878842        46.469835         7.696608         2.379515         4.911599
TEAM_PITCHING_HR TEAM_PITCHING_BB  TEAM_FIELDING_E
       39.534277         6.462131         5.463255
```

Observations: Figure 18 shows a summary of the multiple regression model TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_FIELDING_E. These variables were chosen based on the correlation matrix *(correlation greater than 0.10)*, which helps us choose the most promising predictor variables based on the predictors that are most correlated with TARGET_WINS. It's important to note that there is multicollinearity issues in this model (large VIF numbers > 10) and as a result this has caused the model to produce unstable regression coefficients. For example, the coefficients on TEAM_BATTING_2B and TEAM_BATTING_HR are negative, when they should be positive. Additionally, coefficients such as TEAM_PITCHING_H and TEAM_PITCHING_HR should be negative, but they are showing up as positive. As a result, given these issues I will create another model to address these issues (e.g., removing highly correlated predictor variables and variables that were not highly significant). This means that we are making a tradeoff – settling for less accuracy, but more precision. The residual standard error of 12.8, shows us that when predicting TARGET_WINS, one standard error = 12.8. The multiple R-squared value of 0.3229, indicates that 32.29% of the variation in TARGET_WINS is explained by the predictor variables.

Figure 19: Model MLRResult2

```
> anova(MLRResult2)
Analysis of Variance Table

Response: TARGET_WINS
                Df Sum Sq Mean Sq  F value    Pr(>F)
TEAM_BATTING_H   1  79823   79823 486.7025 < 2.2e-16 ***
TEAM_BATTING_2B  1   3584    3584  21.8527 3.117e-06 ***
TEAM_BATTING_3B  1    249     249   1.5207    0.2176
TEAM_BATTING_BB  1  30075   30075 183.3792 < 2.2e-16 ***
TEAM_BASERUN_SB  1  10131   10131  61.7743 5.901e-15 ***
TEAM_PITCHING_HR 1  10231   10231  62.3791 4.377e-15 ***
TEAM_PITCHING_BB 1  13666   13666  83.3257 < 2.2e-16 ***
TEAM_FIELDING_E  1  31152   31152 189.9452 < 2.2e-16 ***
Residuals     2267 371804     164
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> summary(MLRResult2)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_HR +
    TEAM_PITCHING_BB + TEAM_FIELDING_E, data = moneyball3)

Residuals:
     Min       1Q   Median       3Q      Max
-66.618  -8.465  -0.171   8.350  65.812

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      2.628492   3.514605   0.748 0.454612
TEAM_BATTING_H   0.046641   0.003178  14.677  < 2e-16 ***
TEAM_BATTING_2B -0.029640   0.008701  -3.407 0.000669 ***
TEAM_BATTING_3B  0.123138   0.016335   7.538 6.84e-14 ***
TEAM_BATTING_BB  0.032862   0.005147   6.385 2.07e-10 ***
TEAM_BASERUN_SB  0.079434   0.004809  16.516  < 2e-16 ***
TEAM_PITCHING_HR  0.035914   0.007101   5.057 4.59e-07 ***
TEAM_PITCHING_BB -0.013470   0.004161  -3.237 0.001225 **
TEAM_FIELDING_E  -0.062987   0.004570 -13.782  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.81 on 2267 degrees of freedom
Multiple R-squared:  0.3249,    Adjusted R-squared:  0.3225
F-statistic: 136.4 on 8 and 2267 DF,  p-value: < 2.2e-16
```




```
> vif(MLRResult2)
  TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_BB  TEAM_BASERUN_SB TEAM_PITCHING_HR TEAM_PITCHING_BB
        2.551351         2.285044         2.809715         4.066630         2.370729         2.482643         3.262316
TEAM_FIELDING_E
        4.759357
```

Observations: Figure 19 shows an anova, summary, and diagonostic plots of the multiple regression model MLRResult2<- lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_FIELDING_E, data = moneyball3). This model adjusts for multicollinearity issues and variables that were not highly significant. A statistically significant result was obtained overall as indicated by the F-statistic which is 136.4 with a p-value = < 2.2e-16. This indicates the model has produced statistically significant results to be investigated. Additionally, the t-test of all the predictor variables are statistically significant. The residual standard error of 12.81, shows us that when predicting TARGET_WINS, one standard error = 12.81. The multiple R-squared value of 0.3225, indicates that 32.25% of the variation in TARGET_WINS is explained by the predictor variables. Overall, the multiple R-squared value is slightly lower than the previous model, but the model is less complex. Additionally, there is no evidence of multicollinearity issues in this model. Furthermore, most of the coefficients in the model makes intuitive baseball sense. For instance most of the TEAM_BATTING variables are positive, with the exception of TEAM_BATTING_2B, while TEAM_FIELDING_E and TEAM_PITCHING_BB are negative. This means that if a team gets a lot of hits, hits a lot of triples, obtains a lot of walks, steals a lot of bases, and hits a lot of homeruns, it would reasonably expected that such a team would win more games. On the other hand, if a team commits a lot of errors and walks a lot of batters, it will have a negative impact on wins and as a result would lose more games. The only point of concern is with the variable TEAM_BATTING_2B. This value has a negative coefficient associated with it which suggests that when a team hits more doubles (which is good) then the team will lose more games. This is counterintuitive, so this would have to be explored further. However, a possible explanation of this is that the coefficient signs are with regards to the other variables in the model so it's possible that the coefficient is negative based on the other batting variables. Before officially deploying this model into production it may be a good idea to consult a

baseball expert and determine if the value should be positive. If no explanation can be found, it may be wise to remove the variable from the model. Lastly, the scatterplots with residuals and qq-plots of residuals is shown so that we can check to make sure the model is meeting all the assumptions. The QQ plot reveals that the density distribution is slightly non-normal. This is present in the plot where some of the data points are departing from the line in the upper right hand corner. The scatterplot of residuals vs. fitted also shows us a relatively healthy plot. The plot is relatively linear and has random scatter of data over the range of values for the independent variable.

Model 2: Stepwise Regression Model

Figure 20: Stepwise Model

```
> summary(stepwise)
Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BATTING_SO +
    TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_PITCHING_HR + TEAM_PITCHING_BB +
    TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP + SB_PCT,
    data = moneyball3)

Residuals:
    Min      1Q  Median      3Q     Max
-62.633  -8.027   0.063   7.936  74.569

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       30.554247   6.996035   4.367 1.31e-05 ***
TEAM_BATTING_1B    0.059547   0.010708   5.561 3.00e-08 ***
TEAM_BATTING_2B    0.038074   0.013587   2.802 0.005117 **
TEAM_BATTING_3B    0.150400   0.018828   7.988 2.16e-15 ***
TEAM_BATTING_H    -0.017402   0.011389  -1.528 0.126671
TEAM_BATTING_BB    0.031425   0.005995   5.241 1.74e-07 ***
TEAM_BATTING_SO    0.035212   0.010290   3.422 0.000633 ***
TEAM_BASERUN_SB    0.100701   0.010958   9.190  < 2e-16 ***
TEAM_BASERUN_CS   -0.069073   0.024600  -2.808 0.005030 **
TEAM_PITCHING_HR   0.107121   0.012738   8.409  < 2e-16 ***
TEAM_PITCHING_BB  -0.012576   0.004625  -2.719 0.006595 **
TEAM_PITCHING_SO  -0.007747   0.001957  -3.959 7.75e-05 ***
TEAM_FIELDING_E   -0.071929   0.004842 -14.854  < 2e-16 ***
TEAM_FIELDING_DP  -0.107432   0.012541  -8.566  < 2e-16 ***
SB_PCT           -13.199122   5.380361  -2.453 0.014234 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.5 on 2260 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.3549, Adjusted R-squared:  0.3509
F-statistic: 88.82 on 14 and 2260 DF,  p-value: < 2.2e-16
```

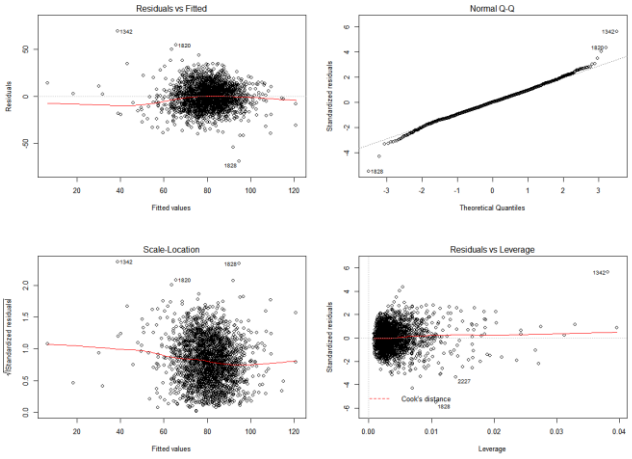
```
> vif(stepwise)
 TEAM_BATTING_1B  TEAM_BATTING_2B  TEAM_BATTING_3B   TEAM_BATTING_H  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
       27.691378         5.838282         3.915368        34.233764         5.783773         1.830412        12.913853
 TEAM_BASERUN_CS TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP           SB_PCT
       9.165895         8.382636         4.223666         3.346607         5.598508         1.911174         3.422588
```

Observations: Figure 20 shows a summary of the multiple regression model TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP + SB_PCT. These variables were chosen using stepwise regression. It's important to note that there are some multicollinearity issues in this model (large VIF numbers > 10) and as a result this has caused the model to produce unstable regression coefficients. This is evident in the model where some of the regression coefficients do not make intuitive baseball sense. As a result, given these issues I will create another model to address these issues (e.g., removing highly correlated predictor variables and predictor variables that were not highly significant, etc.). I will also experiment with log transformations on some of the variables as well. This means that we are making a tradeoff – settling for less accuracy, but more precision and simplicity. The residual standard error of 12.5, shows us that when predicting TARGET_WINS, one standard error = 12.5. The multiple R-squared value of 0.3509, indicates that 35.09% of the variation in TARGET_WINS is explained by the predictor variables.

Model 3

Figure 21: Model 3

```
> anova(Model3)
Analysis of Variance Table

Response: TARGET_WINS
                    Df Sum Sq Mean Sq F value    Pr(>F)
logTEAM_BATTING_1B    1  24112   24112 152.602 < 2.2e-16 ***
logTEAM_BATTING_2B    1  44171   44171 279.554 < 2.2e-16 ***
TEAM_BATTING_3B       1   3269    3269  20.687 5.694e-06 ***
TEAM_BATTING_BB       1  32496   32496 205.666 < 2.2e-16 ***
TEAM_BATTING_HR       1  23479   23479 148.597 < 2.2e-16 ***
TEAM_BASERUN_SB       1  14757   14757  93.393 < 2.2e-16 ***
TEAM_FIELDING_E       1  37679   37679 238.464 < 2.2e-16 ***
logTEAM_FIELDING_DP   1  12555   12555  79.459 < 2.2e-16 ***
Residuals          2267 358198     158
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> summary(Model3)

Call:
lm(formula = TARGET_WINS ~ logTEAM_BATTING_1B + logTEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_BB + TEAM_BATTING_HR + TEAM_BASERUN_SB +
    TEAM_FIELDING_E + logTEAM_FIELDING_DP, data = moneyball3)

Residuals:
    Min      1Q  Median      3Q     Max
-68.523  -8.226   0.186   8.001  69.197

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.764e+02  2.249e+01 -12.289  < 2e-16 ***
logTEAM_BATTING_1B  5.452e+01  3.392e+00  16.071  < 2e-16 ***
logTEAM_BATTING_2B  5.273e+00  1.661e+00   3.174  0.00152 **
TEAM_BATTING_3B     1.609e-01  1.519e-02  10.590  < 2e-16 ***
TEAM_BATTING_BB     2.138e-02  3.060e-03   6.986 3.69e-12 ***
TEAM_BATTING_HR     8.643e-02  7.540e-03  11.462  < 2e-16 ***
TEAM_BASERUN_SB     6.535e-02  4.969e-03  13.151  < 2e-16 ***
TEAM_FIELDING_E    -7.216e-02  4.307e-03 -16.756  < 2e-16 ***
logTEAM_FIELDING_DP -1.484e+01  1.665e+00  -8.914  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.57 on 2267 degrees of freedom
Multiple R-squared:  0.3496,    Adjusted R-squared:  0.3473
F-statistic: 152.3 on 8 and 2267 DF,  p-value: < 2.2e-16
```




```
> vif(Model3)
 logTEAM_BATTING_1B  logTEAM_BATTING_2B     TEAM_BATTING_3B     TEAM_BATTING_BB     TEAM_BATTING_HR     TEAM_BASERUN_SB
           2.033446            1.581406            2.523482            1.492480            2.997891            2.626954
    TEAM_FIELDING_E logTEAM_FIELDING_DP
           4.386834            1.951070
```

Observations: Figure 21 shows an anova, summary, and diagonostic plots of the multiple regression logTEAM_BATTING_1B + logTEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_BB + TEAM_BATTING_HR + TEAM_BASERUN_SB + TEAM_FIELDING_E + logTEAM_FIELDING_D. This model adjusts for multicollinearity issues and removes variables that were not highly significant, which has resulted in a simpler model. A statistically significant result was obtained overall as indicated by the F-

statistic which is 152.3 with a p-value = < 2.2e-16. This indicates the model has produced statistically significant results to be investigated. Additionally, the t-test of all the predictor variables are statistically significant. The residual standard error of 12.57, shows us that when predicting TARGET_WINS, one standard error = 12.57. The multiple R-squared value of 0.3473, indicates that 34.73% of the variation in TARGET_WINS is explained by the predictor variables. Overall, the multiple R-squared value is slightly lower than the stepwise model, but there is no evidence of multicollinearity issues in this model and the model is less complex. Furthermore, most of the coefficients in the model makes intuitive baseball sense. For instance, TEAM_FIELDING_E is negative, while the TEAM_BATTING HIT variables and TEAM_BASERUN SB are positive. This means that if a team gets a lot of hits (1Bs, 2Bs, 3Bs), obtains a lot of walks, hits a lot of homeruns, and steals a lot of bases, it would reasonably expected that such a team would win more games. On the other hand, if a team commits a lot of errors, it will have a negative impact on wins and as a result would lose more games. The only point of concern is with the variable “double plays”. This value has a negative coefficient associated with it which suggests that when a team generates more double plays (which is good) then the team will lose more games. This is counterintuitive, so this would have to be explored further. However, a possible explanation of this is that in order for double plays to occur…the team would have to give up hits (which is bad). This could possibly explain the negative coefficient. Before officially deploying this model into production it may be a good idea to consult a baseball expert and determine if the value should be positive. If no explanation can be found, it may be wise to remove the variable from the model. Additionally, it’s important to note that when this variable is removed, the adjusted r-squared decreases, which is why I’ve left it in the model. Lastly, the scatterplots with residuals and qq-plots of residuals is shown so that we can check to make sure the model is meeting all the assumptions. The QQ plot reveals that the density distribution is slightly non-normal. This is present in the plot where some of the data parts are departing from the line in the upper right hand corner. The scatterplot of residuals vs. fitted also shows us a relatively healthy plot. The plot is relatively linear and has random scatter of data over the range of values for the independent variable.

Section 4: Selection Models

Figure 22: Model Comparison and Criteria for Selecting the "Best Model"

Model Name	Adj R-Squared	Rank	AIC	Rank	BIC	Rank	MSE	Rank	Total Points & Best Model*
MLRResult1	0.3229	3	18078.06	4	18146.82	4	163.1199	3	6
MLRResult2	0.3225	4	18077.38	3	18134.68	3	163.3584	4	6
stepwisemodel	0.3509	1	17965.64	1	18063.05	2	155.1144	1	15
Model3	0.3473	1	17992.53	2	18049.84	1	157.3806	2	14*

**Points: Rank 1 = 4 points, Rank 2 = 3 points, Rank 3 = 2 points, Rank 4 = 1 point*

Observations: Figure 22 shows the model comparisons so that we can compare the in-sample fit and predictive accuracy of our models so that we can select the best model. The results above show the computations for adjusted R-Squared, AIC, BIC, and mean squared error for each of these models. Each of these metrics represent some concept of 'fit' (e.g., rewarding for accuracy and penalizing for complexity). Additionally, each model was ranked on each metric. Points were then allotted to each model based on how they ranked on each metric.

As a result, given the criteria above. Model3 is the best model despite it having 1 fewer point than the stepwisemodel because it ranked in the upper echelon on all the metrics and was more intuitive than the stepwisemodel (e.g., the model made more baseball sense…the regression coefficients made more sense than the stepwise model, no multicollinearity existed, and therefore would be more explainable to the boss, etc.). This model was also the least complex and as a result had more precision, but slightly less accuracy than the stepwisemodel.

The formula given for the number of wins in a season is:

WINS = -2.764e+02
+ 5.452e+01* X1 Singles by Batters (log)
+ 5.273e+00* X2 Doubles by Batters (log)
+ 1.609e-01* X3 Triples by Batters
+ 2.138e-02* X4 Walks by Batters
+ 8.643e-02* X5 Homeruns by Batters
+ 6.535e-02* X5 Stolen Bases
- 7.216e-02* X6 Errors Committed
- 1.484e+01* X7 Double Plays Turned (log)

For the most part, this formula makes intuitive baseball sense because good performance by the team (singles, doubles, triples, homeruns, stolen bases, and walks by batters) are all rewarded with a positive coefficient indicating that the results would be associated with winning. Likewise, when a team commits errors, then there will be less wins. The only point of concern is with the variable "double plays". This value has a negative coefficient associated with it which suggests that when a team generates more double plays (which is good) then the team will lose more games. This is counterintuitive, so this would have to be explored further. However, a possible explanation of this is that in order for double plays to occur…the team would have to give up hits (which is bad). This could possibly explain the negative coefficient. Before officially deploying this model into production it may be a good idea to consult a

baseball expert and determine if the value should be positive. If it no explanation can be found, it may be wise to remove the variable from the model.

Section 5: Stand Alone Scoring Program

```
#Test Data
setwd("~/R/Moneyball")
moneyball_test=read.csv("moneyball_test.csv",header=T)

#Fixing na's
library(mice)

#Check for missing values
sapply(moneyball_test, function(x) sum(is.na(x)))

#Check missing data percentage
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(moneyball_test,2,pMiss)

library(VIM)
aggr_plot <- aggr(moneyball_test, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
labels=names(moneyball_test), cex.axis=.5, gap=2, ylab=c("Histogram of missing data","Pattern"))

#Run imputation
tempData <- mice(moneyball_test,m=5,maxit=50,meth='pmm',seed=500)
summary(tempData)

#Check N/A values have been removed
moneyball5  <- complete(tempData,1)
apply(moneyball5 ,2,pMiss)
summary(moneyball5 )

densityplot(tempData)

stripplot(tempData, pch = 20, cex = 1.2)

#Straighten Relationships – Create transformed variables that we can look at later

moneyball5$TEAM_BATTING_1B <- moneyball5$TEAM_BATTING_H - moneyball5$TEAM_BATTING_HR -
moneyball5$TEAM_BATTING_3B - moneyball5$TEAM_BATTING_2B


moneyball5$logTEAM_BATTING_1B <- log(moneyball5$TEAM_BATTING_1B)
moneyball5$logTEAM_BATTING_2B <- log(moneyball5$TEAM_BATTING_2B)
moneyball5$logTEAM_FIELDING_DP <- log(moneyball5$TEAM_FIELDING_DP)
```

#Trim Data

```
moneyball5$TEAM_BATTING_H[(moneyball5$TEAM_BATTING_H >= 2000)] = 2000
moneyball5$TEAM_BATTING_H[(moneyball5$TEAM_BATTING_H <= 1000)] = 1000
moneyball5$TEAM_BATTING_2B[(moneyball5$TEAM_BATTING_2B >= 400)] = 400
moneyball5$TEAM_BATTING_2B[(moneyball5$TEAM_BATTING_2B <= 100)] = 100
moneyball5$TEAM_BATTING_3B[(moneyball5$TEAM_BATTING_3B >= 160)] = 160
moneyball5$TEAM_BATTING_3B[(moneyball5$TEAM_BATTING_3B <= 10)] = 10
moneyball5$TEAM_BATTING_HR[(moneyball5$TEAM_BATTING_HR <= 3)] = 3
moneyball5$TEAM_BATTING_BB[(moneyball5$TEAM_BATTING_BB >= 825)] = 825
moneyball5$TEAM_BATTING_BB[(moneyball5$TEAM_BATTING_BB <= 280)] = 280
moneyball5$TEAM_BATTING_SO[(moneyball5$TEAM_BATTING_SO >= 300)] = 300
moneyball5$TEAM_BASERUN_SB[(moneyball5$TEAM_BASERUN_SB >= 350)] = 350
moneyball5$TEAM_BASERUN_SB[(moneyball5$TEAM_BASERUN_SB <= 14)] = 14
moneyball5$TEAM_BASERUN_CS[(moneyball5$TEAM_BASERUN_CS >= 125)] = 125
moneyball5$TEAM_BASERUN_CS[(moneyball5$TEAM_BASERUN_CS <= 10)] = 10
moneyball5$TEAM_PITCHING_H[(moneyball5$TEAM_PITCHING_H >= 2000 )] = 2000
moneyball5$TEAM_PITCHING_HR[(moneyball5$TEAM_PITCHING_HR >= 260)] = 260
moneyball5$TEAM_PITCHING_HR[(moneyball5$TEAM_PITCHING_HR <= 25)] = 25
moneyball5$TEAM_PITCHING_BB[(moneyball5$TEAM_PITCHING_BB >= 1000)] = 1000
moneyball5$TEAM_PITCHING_BB[(moneyball5$TEAM_PITCHING_BB <= 300)] = 300
moneyball5$TEAM_PITCHING_SO[(moneyball5$TEAM_PITCHING_SO >= 1550)] = 1550
moneyball5$TEAM_PITCHING_SO[(moneyball5$TEAM_PITCHING_SO <= 100)] = 100
moneyball5$TEAM_FIELDING_E[(moneyball5$TEAM_FIELDING_E >= 500)] = 500
```

 # Stand Alone Scoring

```
moneyball5$P_TARGET_WINS <- -2.764e+02 +
  5.452e+01*moneyball5$logTEAM_BATTING_1B+
  5.273e+00*moneyball5$logTEAM_BATTING_2B+
  1.609e-01*moneyball5$TEAM_BATTING_3B +
  8.643e-02*moneyball5$TEAM_BATTING_HR+
  2.138e-02*moneyball5$TEAM_BATTING_BB+
  6.535e-02*moneyball5$TEAM_BASERUN_SB-
  7.216e-02*moneyball5$TEAM_FIELDING_E-
1.484e+01*moneyball5$logTEAM_FIELDING_DP
```

#subset of data set for the deliverable "Scored data file"

```
prediction <- moneyball5[c("INDEX","P_TARGET_WINS")]

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

#Prediction File

```
write.xlsx(prediction, file = "writeQ.xlsx", sheetName = "Predictions",
        col.names = TRUE)
```

Section 6: Scored Data File

Summary Statistics ***Predicted Number of Wins for Quality Control Purposes***	
MEAN	80.12
MEDIAN	80.55
MAX	109.28
MIN	34.37

Conclusion

In section 1, we conducted an initial exploratory data analysis using scatterplots, boxplots, summary statistics, etc. to help understand important characteristics and properties of the data that may be disguised by numerical summaries. The EDA revealed outliers and missing values for 6 variables: TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_SO, and TEAM_FIELDING_DP.

In section 2, we conducted data preparation/transformations of the data by fixing the missing values using predictive mean matching, conducting data transformations, creating new variables, and handling outliers.

In section 3, we built 4 linear regression models using different variables (or the same variables with different transformations). This was conducted by manually selecting the variables or using variable selection techniques. We then ran model diagnostics and discussed the coefficients in the model to ensure that it makes intuitive baseball sense.

In section 4, we selected Model3 as our “best model” based on ‘fit’ (adjusted r-squared, AIC, BIC, MSE) metrics and intuitiveness of the model.

Lastly, in section 5, a Stand Alone scoring program was conducted that scored the new data and predicted the number of wins. The summary statistics showed the following: mean (80.12), median (80.55), max (109.28), and min (34.37). The data step also included all the variable transformations such as fixing missing values and the regression formula.

Fulll Code

```
library(rJava)
library(readr)
library(pbkrtest)
library(car)
library(leaps)
library(MASS)
library(xlsxjars)
library(xlsx)

setwd("~/R/Moneyball")

moneyball=read.csv("moneyball.csv",header=T)

#Part 1: Data Exploration

#Data Quality Check
str(moneyball)
summary(moneyball)

library(Hmisc)
describe(moneyball)

#Wins
par(mfrow=c(1,2))

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

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

par(mfrow = c(1,1))

#Batting
# Hits and Doubles
par(mfrow=c(2,2))
hist(moneyball$TEAM_BATTING_H, col = "#A71930", xlab = "Team_Batting_H", main = "Histogram of Hits")
hist(moneyball$TEAM_BATTING_2B, col = "#09ADAD", xlab = "Doubles", main = "Histogram of Doubles")
boxplot(moneyball$TEAM_BATTING_H, col = "#A71930", main = "Boxplot of Hits")
boxplot(moneyball$TEAM_BATTING_2B, col = "#09ADAD", main = "Boxplot of Doubles")
par(mfrow=c(1,1))

# Triples and Home Runs

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

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

# Walks, Strikeouts, HBP

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

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

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

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

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

```
boxplot(moneyball$TEAM_FIELDING_E, col = "#09ADAD", main = "Boxplot of Errors Committed")
par(mfrow=c(1,1))

# Scatterplot Matrix

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

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

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

#Fielding Stats and Wins
pairs(~ moneyball$TARGET_WINS + moneyball$TEAM_BASERUN_CS + moneyball$TEAM_FIELDING_E,
lower.panel = panel.smooth)

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

#Correlation Matrix
subdatnum <- subset(moneyball, select=c(
"TEAM_BATTING_H",
"TEAM_BATTING_2B",
"TEAM_BATTING_3B",
"TEAM_BATTING_HR",
"TEAM_BATTING_BB",
"TEAM_BATTING_SO",
"TEAM_BASERUN_SB",
"TEAM_BASERUN_CS",
"TEAM_BATTING_HBP",
"TEAM_PITCHING_H",
"TEAM_PITCHING_HR",
"TEAM_PITCHING_BB",
"TEAM_PITCHING_SO",
"TEAM_FIELDING_E",
"TEAM_FIELDING_DP",
```

```
"TARGET_WINS"))

require(corrplot)
mcor <- cor(subdatnum)
corrplot(mcor, method="number", shade.col=NA, tl.col="black",tl.cex=0.8)

#Part 2: Data Preparation
library(mice)

#Check for missing values
sapply(moneyball, function(x) sum(is.na(x)))

#Check missing data percentage
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(moneyball,2,pMiss)

library(VIM)
aggr_plot <- aggr(moneyball, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
labels=names(moneyball), cex.axis=.5, gap=2, ylab=c("Histogram of missing data","Pattern"))

moneyball2 <- subset(moneyball, select=c(
  "TEAM_BATTING_H",
  "TARGET_WINS",
  "TEAM_BATTING_HBP",
  "TEAM_BATTING_2B",
  "TEAM_BATTING_3B",
  "TEAM_BATTING_HR",
  "TEAM_BATTING_BB",
  "TEAM_BATTING_SO",
  "TEAM_BASERUN_SB",
  "TEAM_BASERUN_CS",
  "TEAM_PITCHING_H",
  "TEAM_PITCHING_HR",
  "TEAM_PITCHING_BB",
  "TEAM_PITCHING_SO",
  "TEAM_FIELDING_E",
  "TEAM_FIELDING_DP"))

#Run imputation
tempData <- mice(moneyball2,m=5,maxit=50,meth='pmm',seed=500)
summary(tempData)

#Check N/A values have been removed
moneyball3 <- complete(tempData,1)
apply(moneyball3,2,pMiss)
summary(moneyball3)
```

```
# Inspecting the distribution of original and imputed data for the variables that contained N/A
xyplot(tempData,TARGET_WINS~ TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_BASERUN_CS +
TEAM_PITCHING_SO + TEAM_FIELDING_DP + TEAM_BATTING_HBP,pch=18,cex=1)

densityplot(tempData)

stripplot(tempData, pch = 20, cex = 1.2)
```

#Straighten Relationships – Create transformed variables that we can look at later

```
moneyball3$TEAM_BATTING_1B <- moneyball3$TEAM_BATTING_H - moneyball3 $TEAM_BATTING_HR -
moneyball3$TEAM_BATTING_3B - moneyball3$TEAM_BATTING_2B

moneyball3$SB_PCT <- moneyball3$TEAM_BASERUN_SB/(1.0* moneyball3$TEAM_BASERUN_SB+
moneyball3$TEAM_BASERUN_CS)

moneyball3$logTEAM_BATTING_1B <- log(moneyball3$TEAM_BATTING_1B)
moneyball3$logTEAM_BATTING_2B <- log(moneyball3$TEAM_BATTING_2B)
moneyball3$logTEAM_BATTING_3B <- log(moneyball3$TEAM_BATTING_3B)
moneyball3$logTEAM_BATTING_HR<- log(moneyball3$TEAM_BATTING_HR)
moneyball3$logTEAM_BATTING_BB <- log(moneyball3$TEAM_BATTING_BB)
moneyball3$logTEAM_BATTING_SO<- log(moneyball3$TEAM_BATTING_SO)
moneyball3$logTEAM_BASERUN_SB <- log(moneyball3$TEAM_BASERUN_SB)
moneyball3$logTEAM_BASERUN_CS <- log(moneyball3$TEAM_BASERUN_CS)
moneyball3$logTEAM_PITCHING_BB <- log(moneyball3$TEAM_PITCHING_BB)
moneyball3$logTEAM_FIELDING_E <- log(moneyball3$TEAM_FIELDING_E)
moneyball3$logTEAM_FIELDING_DP <- log(moneyball3$TEAM_FIELDING_DP)
```

#Trim Data

```
moneyball3$TARGET_WINS[(moneyball3$TARGET_WINS >= 120)] = 120
moneyball3$TARGET_WINS[(moneyball3$TARGET_WINS <= 21)] = 21
moneyball3$TEAM_BATTING_H[(moneyball3$TEAM_BATTING_H >= 2000)] = 2000
moneyball3$TEAM_BATTING_H[(moneyball3$TEAM_BATTING_H <= 1000)] = 1000
moneyball3$TEAM_BATTING_2B[(moneyball3$TEAM_BATTING_2B >= 400)] = 400
moneyball3$TEAM_BATTING_2B[(moneyball3$TEAM_BATTING_2B <= 100)] = 100
moneyball3$TEAM_BATTING_3B[(moneyball3$TEAM_BATTING_3B >= 160)] = 160
moneyball3$TEAM_BATTING_3B[(moneyball3$TEAM_BATTING_3B <= 10)] = 10
moneyball3$TEAM_BATTING_HR[(moneyball3$TEAM_BATTING_HR <= 3)] = 3
moneyball3$TEAM_BATTING_BB[(moneyball3$TEAM_BATTING_BB >= 825)] = 825
moneyball3$TEAM_BATTING_BB[(moneyball3$TEAM_BATTING_BB <= 280)] = 280
moneyball3$TEAM_BATTING_SO[(moneyball3$TEAM_BATTING_SO >= 300)] = 300
moneyball3$TEAM_BASERUN_SB[(moneyball3$TEAM_BASERUN_SB >= 350)] = 350
moneyball3$TEAM_BASERUN_SB[(moneyball3$TEAM_BASERUN_SB <= 14)] = 14
moneyball3$TEAM_BASERUN_CS[(moneyball3$TEAM_BASERUN_CS >= 125)] = 125
moneyball3$TEAM_BASERUN_CS[(moneyball3$TEAM_BASERUN_CS <= 10)] = 10
moneyball3$TEAM_PITCHING_H[(moneyball3$TEAM_PITCHING_H >= 2000 )] = 2000
moneyball3$TEAM_PITCHING_HR[(moneyball3$TEAM_PITCHING_HR >= 260)] = 260
```

```
moneyball3$TEAM_PITCHING_HR[(moneyball3$TEAM_PITCHING_HR <= 25)] = 25
moneyball3$TEAM_PITCHING_BB[(moneyball3$TEAM_PITCHING_BB >= 1000)] = 1000
moneyball3$TEAM_PITCHING_BB[(moneyball3$TEAM_PITCHING_BB <= 300)] = 300
moneyball3$TEAM_PITCHING_SO[(moneyball3$TEAM_PITCHING_SO >= 1550)] = 1550
moneyball3$TEAM_PITCHING_SO[(moneyball3$TEAM_PITCHING_SO <= 100)] = 100
moneyball3$TEAM_FIELDING_E[(moneyball3$TEAM_FIELDING_E >= 500)] = 500
summary(moneyball3)
```

#Part 3: Model Creation
 # Manual Approach

#Correlation Matrix

```
subdatnum2 <- subset(moneyball3, select=c(
"TEAM_BATTING_HBP",
"TEAM_BATTING_H",
"TEAM_BATTING_2B",
"TEAM_BATTING_3B",
"TEAM_BATTING_HR",
"TEAM_BATTING_BB",
"TEAM_BATTING_SO",
"TEAM_BASERUN_SB",
"TEAM_BASERUN_CS",
"TEAM_PITCHING_H",
"TEAM_PITCHING_HR",
"TEAM_PITCHING_BB",
"TEAM_PITCHING_SO",
"TEAM_FIELDING_E",
"TEAM_FIELDING_DP",
"TARGET_WINS"))

require(corrplot)
mcor <- cor(subdatnum2)
corrplot(mcor, method="number", shade.col=NA, tl.col="black",tl.cex=0.8)
par(mfrow=c(1,1))


MLRResult1<- lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_HR
+ TEAM_PITCHING_BB + TEAM_FIELDING_E, data = moneyball3)

summary(MLRResult1)
vif(MLRResult1)

MLRResult2<- lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_FIELDING_E,
data = moneyball3)
```

```
anova(MLRResult2)
summary(MLRResult2)
par(mfrow=c(2,2))  # visualize four graphs at once
plot(MLRResult2)
vif(MLRResult2)
```

 # Stepwise Approach

```
stepwisemodel <- lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_2B +
TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BATTING_SO +
TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_PITCHING_HR + TEAM_PITCHING_BB +
TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP + SB_PCT, data = moneyball3)

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

summary(stepwise)

anova(stepwise)
summary(stepwise)
par(mfrow=c(2,2))  # visualize four graphs at once
plot(stepwise)
vif(stepwise)
```

#Model 3

```
Model3 <- lm(formula = TARGET_WINS ~ logTEAM_BATTING_1B + logTEAM_BATTING_2B +
   TEAM_BATTING_3B + TEAM_BATTING_BB + TEAM_BATTING_HR + TEAM_BASERUN_SB + TEAM_FIELDING_E
+ logTEAM_FIELDING_DP, data = moneyball3)

anova(Model3)
summary(Model3)
par(mfrow=c(2,2))  # visualize four graphs at once
plot(Model3)
vif(Model3)

subsets <- regsubsets(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_2B +
   TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_PITCHING_BB +TEAM_FIELDING_E + TEAM_FIELDING_DP,
data = moneyball3, nbest = 2)
plot(subsets, scale="adjr2")
```

#Part 4: Performance

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

AIC(MLRResult1)
```

```
AIC(MLRResult2)
AIC(stepwisemodel)
AIC(Model3)

BIC(MLRResult1)
BIC(MLRResult2)
BIC(stepwisemodel)
BIC(Model3)

mse(MLRResult1)
mse(MLRResult2)
mse(stepwisemodel)
mse(Model3)

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

#Part 5: Test Data
setwd("~/R/Moneyball")
moneyball_test=read.csv("moneyball_test.csv",header=T)

# Fixing na's
library(mice)

#Check for missing values
sapply(moneyball_test, function(x) sum(is.na(x)))

#Check missing data percentage
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(moneyball_test,2,pMiss)

library(VIM)
aggr_plot <- aggr(moneyball_test, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
labels=names(moneyball_test), cex.axis=.5, gap=2, ylab=c("Histogram of missing data","Pattern"))

#Run imputation
tempData <- mice(moneyball_test,m=5,maxit=50,meth='pmm',seed=500)
summary(tempData)

#Check N/A values have been removed
moneyball5  <- complete(tempData,1)
apply(moneyball5 ,2,pMiss)
summary(moneyball5 )

densityplot(tempData)
```

```
stripplot(tempData, pch = 20, cex = 1.2)

#Straighten Relationships – Create transformed variables that we can look at later

moneyball5$TEAM_BATTING_1B <- moneyball5$TEAM_BATTING_H - moneyball5$TEAM_BATTING_HR -
moneyball5$TEAM_BATTING_3B - moneyball5$TEAM_BATTING_2B

moneyball5$logTEAM_BATTING_1B <- log(moneyball5$TEAM_BATTING_1B)
moneyball5$logTEAM_BATTING_2B <- log(moneyball5$TEAM_BATTING_2B)
moneyball5$logTEAM_FIELDING_DP <- log(moneyball5$TEAM_FIELDING_DP)

#Trim Data

moneyball5$TEAM_BATTING_H[(moneyball5$TEAM_BATTING_H >= 2000)] = 2000
moneyball5$TEAM_BATTING_H[(moneyball5$TEAM_BATTING_H <= 1000)] = 1000
moneyball5$TEAM_BATTING_2B[(moneyball5$TEAM_BATTING_2B >= 400)] = 400
moneyball5$TEAM_BATTING_2B[(moneyball5$TEAM_BATTING_2B <= 100)] = 100
moneyball5$TEAM_BATTING_3B[(moneyball5$TEAM_BATTING_3B >= 160)] = 160
moneyball5$TEAM_BATTING_3B[(moneyball5$TEAM_BATTING_3B <= 10)] = 10
moneyball5$TEAM_BATTING_HR[(moneyball5$TEAM_BATTING_HR <= 3)] = 3
moneyball5$TEAM_BATTING_BB[(moneyball5$TEAM_BATTING_BB >= 825)] = 825
moneyball5$TEAM_BATTING_BB[(moneyball5$TEAM_BATTING_BB <= 280)] = 280
moneyball5$TEAM_BATTING_SO[(moneyball5$TEAM_BATTING_SO >= 300)] = 300
moneyball5$TEAM_BASERUN_SB[(moneyball5$TEAM_BASERUN_SB >= 350)] = 350
moneyball5$TEAM_BASERUN_SB[(moneyball5$TEAM_BASERUN_SB <= 14)] = 14
moneyball5$TEAM_BASERUN_CS[(moneyball5$TEAM_BASERUN_CS >= 125)] = 125
moneyball5$TEAM_BASERUN_CS[(moneyball5$TEAM_BASERUN_CS <= 10)] = 10
moneyball5$TEAM_PITCHING_H[(moneyball5$TEAM_PITCHING_H >= 2000 )] = 2000
moneyball5$TEAM_PITCHING_HR[(moneyball5$TEAM_PITCHING_HR >= 260)] = 260
moneyball5$TEAM_PITCHING_HR[(moneyball5$TEAM_PITCHING_HR <= 25)] = 25
moneyball5$TEAM_PITCHING_BB[(moneyball5$TEAM_PITCHING_BB >= 1000)] = 1000
moneyball5$TEAM_PITCHING_BB[(moneyball5$TEAM_PITCHING_BB <= 300)] = 300
moneyball5$TEAM_PITCHING_SO[(moneyball5$TEAM_PITCHING_SO >= 1550)] = 1550
moneyball5$TEAM_PITCHING_SO[(moneyball5$TEAM_PITCHING_SO <= 100)] = 100
moneyball5$TEAM_FIELDING_E[(moneyball5$TEAM_FIELDING_E >= 500)] = 500

# Stand Alone Scoring
moneyball5$P_TARGET_WINS <- -2.764e+02 +
  5.452e+01*moneyball5$logTEAM_BATTING_1B+
  5.273e+00*moneyball5$logTEAM_BATTING_2B+
  1.609e-01*moneyball5$TEAM_BATTING_3B +
  8.643e-02*moneyball5$TEAM_BATTING_HR+
  2.138e-02*moneyball5$TEAM_BATTING_BB+
  6.535e-02*moneyball5$TEAM_BASERUN_SB-
  7.216e-02*moneyball5$TEAM_FIELDING_E-
1.484e+01*moneyball5$logTEAM_FIELDING_DP

#subset of data set for the deliverable "Scored data file"
prediction <- moneyball5[c("INDEX","P_TARGET_WINS")]
```

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

#Prediction File
write.xlsx(prediction, file = "writeQ.xlsx", sheetName = "Predictions",
        col.names = TRUE)
```

Appendix

Data Quality Check (Figure 3)

```
> describe(moneyball)
moneyball

 17  Variables      2276  Observations
---------------------------------------------------------------------------------------------------
INDEX
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0     2276        1     1268    850.4    125.8    252.5    630.8   1270.5   1915.5
     .90      .95
  2287.5   2407.2

lowest :    1    2    3    4    5, highest: 2531 2532 2533 2534 2535
---------------------------------------------------------------------------------------------------
TARGET_WINS
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      108        1    80.79    17.47     54.0     61.0     71.0     82.0     92.0
     .90      .95
    99.5    104.0

lowest :   0  12  14  17  21, highest: 128 129 134 135 146
---------------------------------------------------------------------------------------------------
TEAM_BATTING_H
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      569        1     1469    149.8     1282     1315     1383     1454     1537
     .90      .95
    1636     1695

lowest :  891  992 1009 1116 1122, highest: 2333 2343 2372 2496 2554
---------------------------------------------------------------------------------------------------
TEAM_BATTING_2B
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      240        1    241.2    52.89      167      182      208      238      273
     .90      .95
     303      320

lowest :  69 112 113 118 123, highest: 382 392 393 403 458
---------------------------------------------------------------------------------------------------
TEAM_BATTING_3B
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      144        1    55.25    30.34       23       27       34       47       72
     .90      .95
      96      108

lowest :   0   8   9  11  12, highest: 166 190 197 200 223
---------------------------------------------------------------------------------------------------
TEAM_BATTING_HR
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      243        1    99.61    69.49     14.0     20.0     42.0    102.0    147.0
     .90      .95
   179.5    199.0

lowest :   0   3   4   5   6, highest: 247 249 257 260 264
---------------------------------------------------------------------------------------------------
```

```
TEAM_BATTING_BB
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      533        1    501.6    130.1    248.2    363.5    451.0    512.0    580.0
     .90      .95
   635.0    670.2

lowest :   0  12  29  34  45, highest: 815 819 824 860 878
------------------------------------------------------------------------------------------------
TEAM_BATTING_SO
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2174      102      822        1    735.6    282.2      359      421      548      750      930
     .90      .95
    1049     1103

lowest :    0   66   67   72   74, highest: 1303 1320 1326 1335 1399
------------------------------------------------------------------------------------------------
TEAM_BASERUN_SB
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2145      131      348        1    124.8    87.96     35.0     44.0     66.0    101.0    156.0
     .90      .95
   231.0    301.8

lowest :   0  14  18  19  20, highest: 562 567 632 654 697
------------------------------------------------------------------------------------------------
TEAM_BASERUN_CS
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    1504      772      128        1     52.8    23.24       24       30       38       49       62
     .90      .95
      77       91

lowest :   0   7  11  12  14, highest: 171 186 193 200 201
------------------------------------------------------------------------------------------------
TEAM_BATTING_HBP
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
     191     2085       55    0.999    59.36    14.61     40.0     44.0     50.5     58.0     67.0
     .90      .95
    76.0     82.5

lowest : 29 30 35 38 39, highest: 87 88 89 90 95
------------------------------------------------------------------------------------------------
TEAM_PITCHING_H
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      843        1     1779    628.1     1316     1356     1419     1518     1682
     .90      .95
    2058     2563

lowest :  1137  1168  1184  1187  1202, highest: 16038 16871 20088 24057 30132
------------------------------------------------------------------------------------------------
TEAM_PITCHING_HR
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      256        1    105.7    70.02     18.0     25.0     50.0    107.0    150.0
     .90      .95
   187.0    209.2

lowest :   0   3   4   5   6, highest: 291 297 301 320 343
------------------------------------------------------------------------------------------------
```

```
TEAM_PITCHING_BB
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      535        1      553    140.7    377.0    417.5    476.0    536.5    611.0
     .90      .95
   693.5    757.0

lowest :    0  119  124  131  140, highest: 2169 2396 2840 2876 3645
---------------------------------------------------------------------------------------------------
TEAM_PITCHING_SO
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2174      102      823        1    817.7    316.9    421.3    490.0    615.0    813.5    968.0
     .90      .95
  1095.0   1173.0

Value          0   200   400   600   800  1000  1200  1400  1600  1800  2200  2400  3400  4200  5400
Frequency     20     7   211   554   593   580   156    35     7     2     1     3     1     1     1
Proportion 0.009 0.003 0.097 0.255 0.273 0.267 0.072 0.016 0.003 0.001 0.000 0.001 0.000 0.000 0.000

Value      12800 19200
Frequency      1     1
Proportion 0.000 0.000
---------------------------------------------------------------------------------------------------
TEAM_FIELDING_E
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    2276        0      549        1    246.5    190.4    100.0    109.0    127.0    159.0    249.2
     .90      .95
   542.0    716.0

lowest :   65   66   68   72   74, highest: 1567 1728 1740 1890 1898
---------------------------------------------------------------------------------------------------
TEAM_FIELDING_DP
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
    1990      286      144        1    146.4    29.29       98      109      131      149      164
     .90      .95
     178      186

lowest :  52  64  68  71  72, highest: 215 218 219 225 228
```