Sports Analytics – Wins Prediction Report

Project #1

Singh, Gurjeet

This report contains analysis done while building regression models for the professional baseball team to predict the number of wins for the team.

Table of Contents

Table of Figures

Introduction:

The purpose of this project is to build OLS ("Linear") Regression models using given statistics of the professional baseball team to predict the number of wins for each team.

For our purposes, we use the data set that contains performance information of the team for the given year. Each record in the data represents a professional baseball team from the years 1871 to 2006 inclusive. The performance statistics have been adjusted to match the performance of a 162 games season. The training dataset contains 2276 observations and 17 explanatory variables. The test dataset contains 259 observations and 16 explanatory variables. In the test dataset, there is no "Target Wins" information. We will be using our selected model to score the test data file to predict the number of wins.

Figure 1 gives the basic descriptions of each field and how those affect the winning chances of a team.

Figure 1: Statistical Values

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT
INDEX	Identification Variable	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

Now that we have some context for our analysis and dataset, let's look at the results in the next section.

Section 1: Data Exploration

The first step towards any modeling project is the Data Exploration i.e. Exploratory Data Analysis (EDA). This helps us to understand and analyze the data set to summarize the main characteristics of variables. For this purposes, we will look into the basic statistics to understand the data, examine the distribution, and examine relationships.

Section 1.1: Statistics

Figure 2 shows the statistical values of the Moneyball training data set. The number of observations in the data is 2276 records. We can clearly see that there are missing values (NAs) in the data for variables, TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_H, TEAM_PITCHING_SO, and TEAM_FIELDING_DP. For TEAM_BATTING_HBP, we can see that more than ninety percent (90%) of the values are missing. Hence, we will not be using this variable in any of our models. In Figure 2, we can also see outliers in various variables in the data. For instance, variables like TEAM_PITCHING_H and TEAM_PITCHING_SO have some extreme outliers. TEAM_PITCHING_H has a maximum value of 30,132. But the median value is only 1518. Therefore, we will need to fix the missing values and outliers prior to building our models.

We explained how we fixed the missing values and outliers in Section 2: Data Preparation.

Figure 2: Statistical Values

Variable Names	nobs	NAs	Minimum	Maximum	1. Quartile	3. Quartile	Mean	Median	Stdev	Skewness	Kurtosis
INDEX	2276	0	1	2535	630.75	1915.5	1268.46353	1270.5	736.34904	0.004215	-1.216756
TARGET_WINS	2276	0	0	146	71	92	80.79086	82	15.75215	-0.398723	1.027476
TEAM_BATTING_H	2276	0	891	2554	1383	1537.25	1469.26977	1454	144.59119	1.571333	7.278526
TEAM_BATTING_2B	2276	0	69	458	208	273	241.24692	238	46.80141	0.215102	0.006161
TEAM_BATTING_3B	2276	0	0	223	34	72	55.25	47	27.93856	1.109465	1.503242
TEAM_BATTING_HR	2276	0	0	264	42	147	99.61204	102	60.54687	0.186042	-0.963119
TEAM_BATTING_BB	2276	0	0	878	451	580	501.55888	512	122.67086	-1.02576	2.182854
TEAM_BATTING_SO	2276	102	0	1399	548	930	735.60534	750	248.52642	-0.2978	-0.320799
TEAM_BASERUN_SB	2276	131	0	697	66	156	124.76177	101	87.79117	1.972414	5.489675
TEAM_BASERUN_CS	2276	772	0	201	38	62	52.80386	49	22.95634	1.976218	7.620382
TEAM_BATTING_HBP	2276	2085	29	95	50.5	67	59.35602	58	12.96712	0.318575	-0.111983
TEAM_PITCHING_H	2276	0	1137	30132	1419	1682.5	1779.21046	1518	1406.84293	10.329511	141.839699
TEAM_PITCHING_HR	2276	0	0	343	50	150	105.69859	107	61.29875	0.287788	-0.604631
TEAM_PITCHING_BB	2276	0	0	3645	476	611	553.00791	536.5	166.35736	6.743899	96.96764
TEAM_PITCHING_SO	2276	102	0	19278	615	968	817.73045	813.5	553.08503	22.174554	671.189129
TEAM_FIELDING_E	2276	0	65	1898	127	249.25	246.48067	159	227.77097	2.990466	10.970272
TEAM_FIELDING_DP	2276	286	52	228	131	164	146.38794	149	26.22639	-0.388939	0.18174

Section 1.2: Examining Distributions

Figure 3 shows the histogram for each variable to help us understand the distribution of the variables. We can see that there are extreme outliers in the variables, TEAM_PITCHING_H, TEAM_PITCHING_BB, TEAM_PITCHING_SO, and TEAM_FIELDING_E. Variables like TEAM_BATTING_3B, TEAM_BASERUN_SB, TEAM_BASERUN_CS, and TEAM_FIELDING_E seems to be skewed right. Variables like TEAM_BATTING_BB and TEAM_FIELDING_DP seems to be skewed left. Both, right and left skewed are telling us that there are some outliers in the variables which need to be fixed. Variables like TEAM_BATTING_HR, TEAM_BATTING_SO, and TEAM_PITCHING_HR seem to bimodal shape.

Figure 3: Histogram: Review Distribution

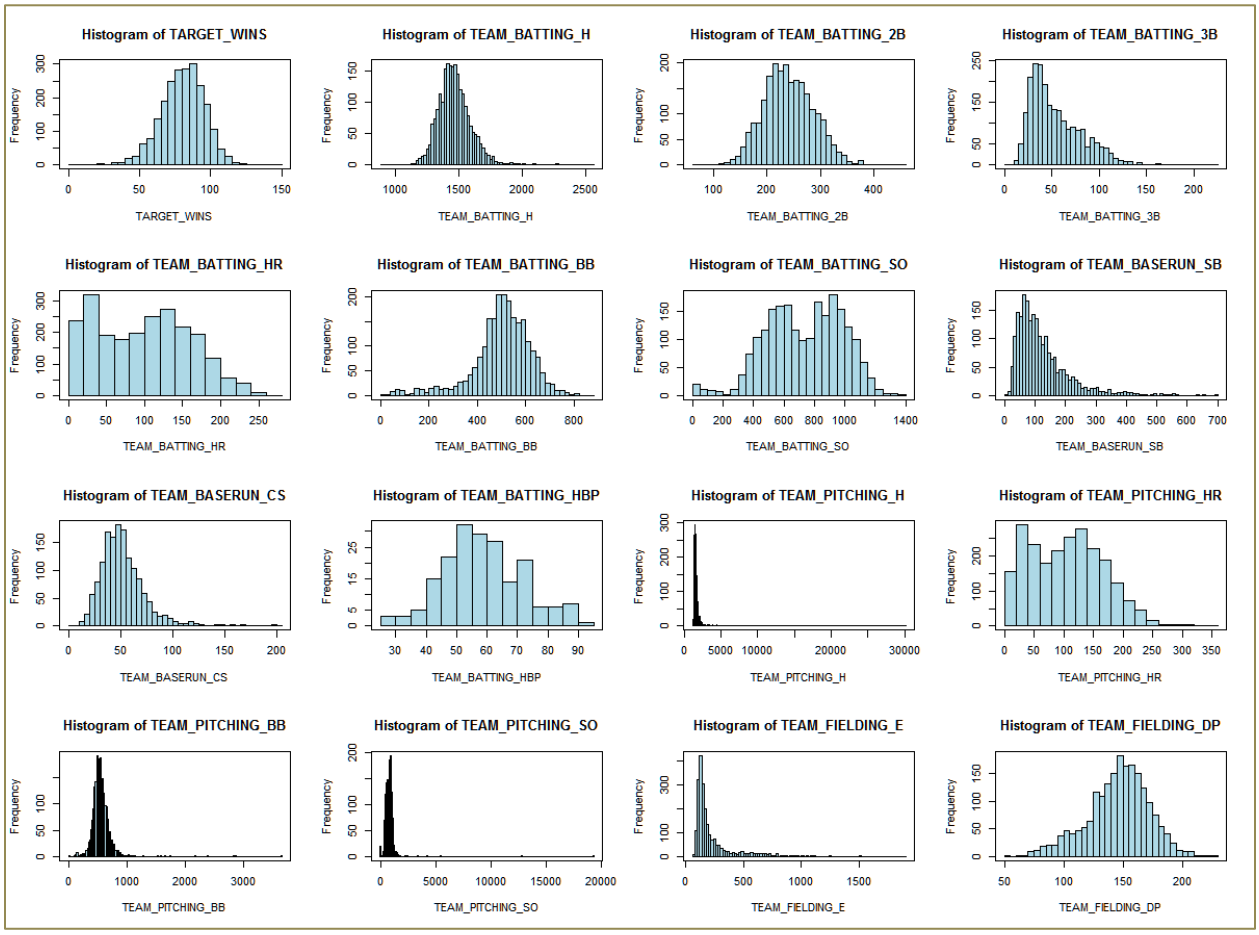
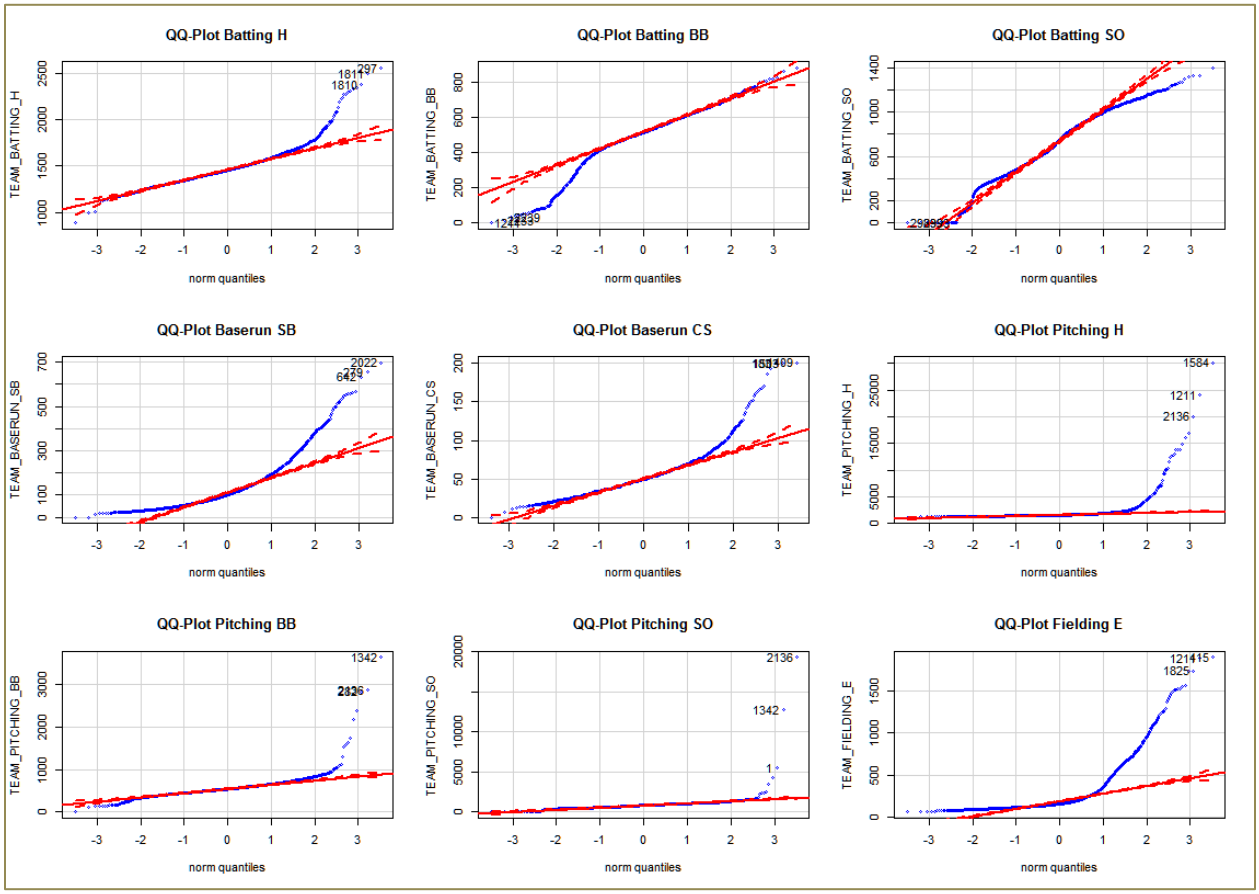


Figure 4 shows the QQ-plots to help us further review the distribution. These variables are selected based on some baseball knowledge I possess. Figure 4 confirms the present of outliers and the evidence of non-normality in the data.

Figure 4: QQ-Plot: Review Distribution



Section 1.3: Examine Relationships

Figure 4 shows the relationship between TARGET_WINS and other variables. Based on the result, it does not appear that there's a linear positive or negative relationship. These could be due to outliers and missing values. Once we clean the data with missing values and outliers, we will do the sanity check and see if there's any difference in the relationship with the transformed data. Checking the correlation would help us determine if there's any relationship.

Figure 5: Scatterplot: Examine Relationship

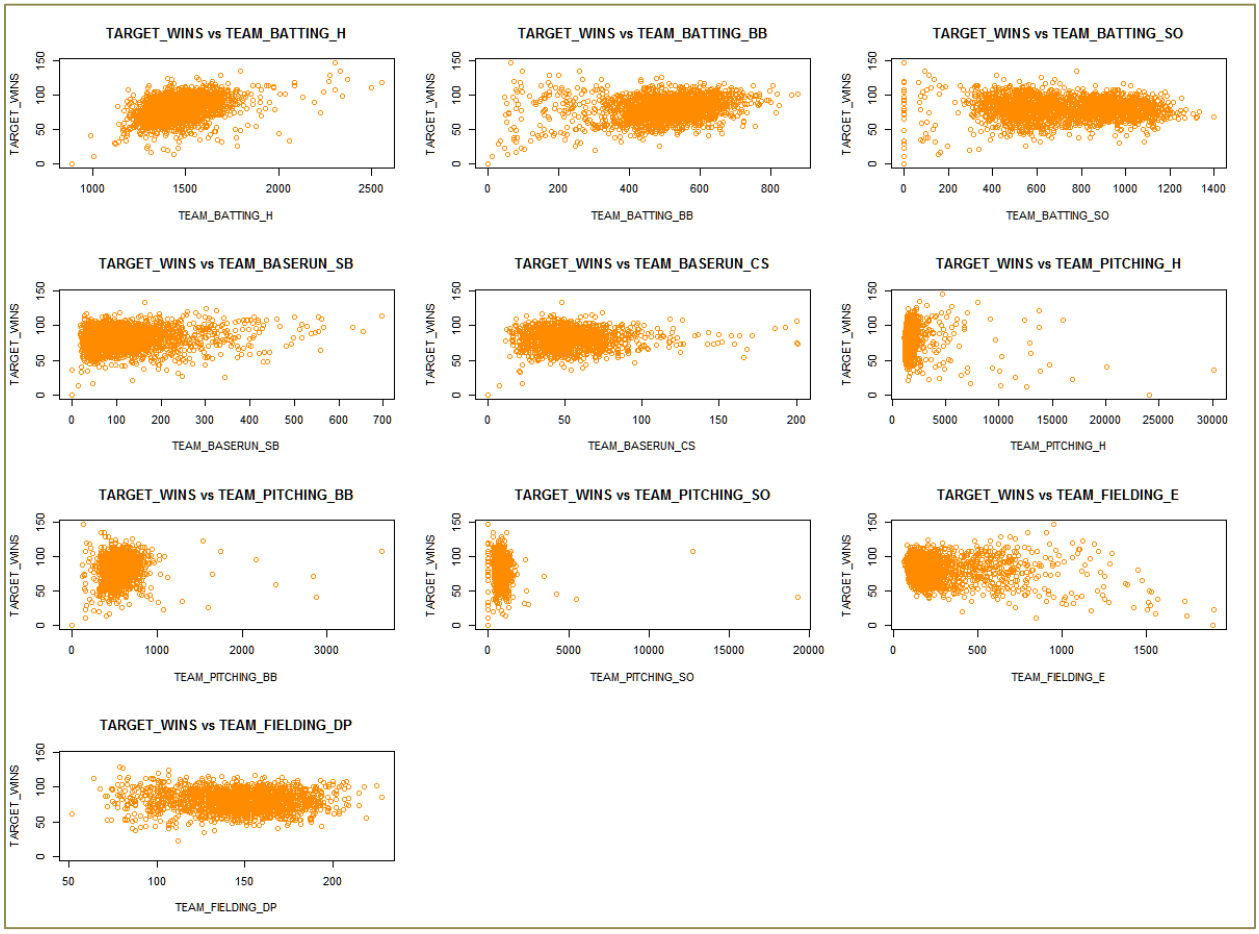
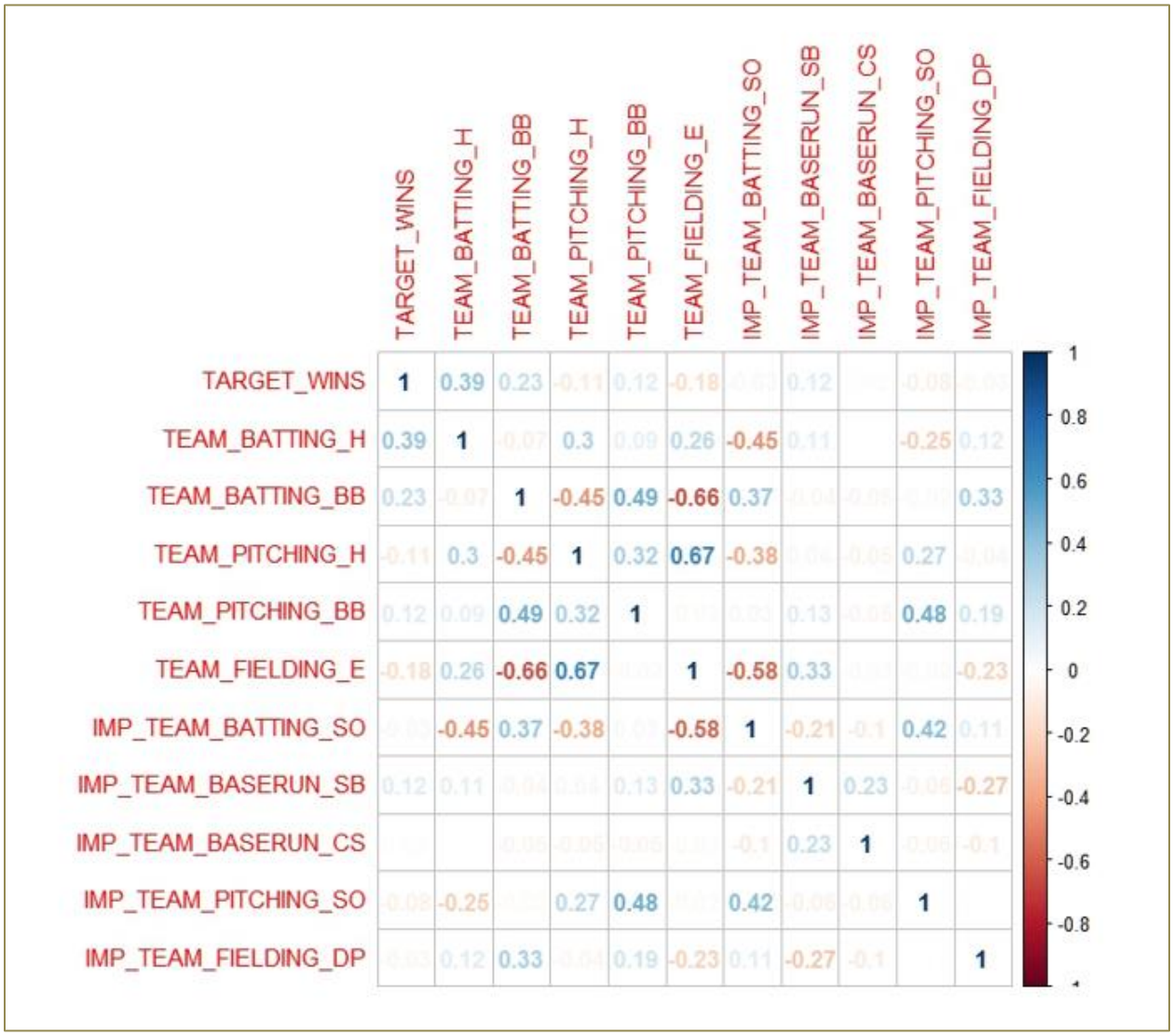


Figure 6 shows the correlation matrix after fixing the missing values. As we saw in Figure 4, Figure 5 also confirms that there was no strong positive or negative linear relationship between the response variable i.e. TARGET_WINS and other predictor variables. In Figure 6, we can also see the relationship of predictor variables with other predictor variables. Here we notice that TEAM_FIELDING_E is negatively related to TEAM_BATTING_BB and IMP_TEAM_BATTING_SO. We will use this information when building models so we can test for multicollinearity.

Figure 6: Correlation Matrix: Examine Relationship



Section 2: Data Preparation

In Section 1: Data Exploration, we noticed that we had some missing values and outliers in the data. This section explains the methodology we took to fix the missing values and transform data to eliminate outliers. First, we fixed the missing values by using the Median values for some variables. For variables, IMP_TEAM_BATTING_SO: we added two (2) additional Strike Out by Batters than the Median value; and IMP_TEAM_PITCHING_SO: we added one (1) additional Strike Out by Pitchers than the Median value. These changes will not impact a lot since the values are pretty close to the Median value. There are five (5) flag variables (indicator/dummy) for variables that had a missing value. These flag variables contain values zeros (0) and ones (1). Zero (0) for original missing values and one (1) for new values. At the bottom of Figure 6, we can see the new values used for the missing values (highlighted in green).

Next, we cleaned the data to fix extreme outliers. For this, we started with our approach to caps the lower and higher values with 1st percentile and 99th percentile respectively. However, after the detailed review of the source data, we used the values that made sense for each variable. For instance, 99th percentile for the variable, IMP_TEAM_PITCHING_H, is 7054 whereas we used the max value of 2200. Using 7000 as the max value did not help much with the distribution. We noticed a histogram plot to a right-skewed. Hence, after going back and forth for a while, we cap it at 2200. Likewise, the values for other variables were entered in a similar approach. Figure 7 also shows the new min and max values (highlighted in green) for each variable to fix the outliers.

Figure 7: Transforming Data: Fix Missing values and Outliers

Fix Outliers						
Predictor Variable	Old MIN Value	Old MAX Value	1-Percentile	99-Percentile	New MIN Value	New MAX Value
IMP_TEAM_BATTING_H	891	2554	1193.25	1945.5	1100	2000
IMP_TEAM_BATTING_BB	0	878	79	752.75	150	800
IMP_TEAM_BATTING_SO	0	1399	72	1191.25	72	1350
IMP_TEAM_BASERUN_SB	0	697	24	434.25	14	500
IMP_TEAM_BASERUN_CS	0	201	18.75	123.5	10	123
IMP_TEAM_PITCHING_H	1137	30132	1244	7054	1200	2200
IMP_TEAM_PITCHING_HR	0	343	8	244	8	260
IMP_TEAM_PITCHING_BB	0	3645	240	921	119	1000
IMP_TEAM_PITCHING_SO	0	19278	241	1461.75	241	1700
IMP_TEAM_FIELDING_E	65	1898	86	1228	65	500
IMP_TEAM_FIELDING_DP	52	228	80	202	71	220

Fix Missing Values	
Predictor Variable	New Value
IMP_TEAM_BATTING_SO	752
IMP_TEAM_BASERUN_SB	101
IMP_TEAM_BASERUN_CS	49
IMP_TEAM_PITCHING_SO	814
IMP_TEAM_FIELDING_DP	149

Section 3: Model Building

For the model building, I took the automated variable selections approach as well manually selected the variables to include in the model. After trying different combinations (of course, combinations that made sense), we selected three models that gave us decent metrics and also logically made sense.

This section explains the three (3) models (Model 1, Model 2, and Model 3) that we have created to predict the number of wins for each team. Out of these models, we have selected one (1) that fits the best and also logically correct to use for predicting the number of wins using test data.

Section 3.1: Model 1

Figure 8 shows the output of the Model 1. Residuals in Figure 8 are essentially the difference between the actual observed response values and the response values that the model predicted. The Residuals section of the model help us assess how well the model fits the data, we should look for a symmetrical distribution across these points on the mean value zero (0). In our case, we can see that the distribution of the residuals appears to be symmetrical since the Median value is very close to zero (0).

The coefficients of each parameter are behaving correctly (as per the theoretical effects in Data Dictionary). For instance, while holding the other predictors constant, if there is an increase of one (1) base hit by the batter (IMP_TEAM_BATTING_H), the winning chances increase by 0.0583 which is roughly 5.83% (approx.). Similarly, if a pitcher allowed one (1) additional hit (IMP_TEAM_PITCHING_H), the winning chances decrease by 0.007589 which is roughly 0.76% (approx.).
*** Significance stars in coefficients in Figure 8 shows that the predictor variables are statistically significant and it's unlikely that no relationship exists between the TARGET_WINS and predictor variables.

Figure 8: Model 1: Output

```
Call:
lm(formula = TARGET_WINS ~ IMP_TEAM_BATTING_H + IMP_TEAM_BATTING_BB +
    IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB + IMP_TEAM_FIELDING_E +
    IMP_TEAM_PITCHING_BB + IMP_TEAM_PITCHING_H, data = MoneyBall)

Residuals:
    Min      1Q  Median      3Q     Max
-55.473  -8.584  -0.326   8.337  52.355

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         -5.833726   3.407116  -1.712   0.0870 .
IMP_TEAM_BATTING_H   0.058274   0.003275  17.796  < 2e-16 ***
IMP_TEAM_BATTING_BB  0.034757   0.005728   6.068 1.51e-09 ***
IMP_TEAM_BASERUN_SB  0.074402   0.004675  15.915  < 2e-16 ***
M_TEAM_BASERUN_SB   28.650995   1.933428  14.819  < 2e-16 ***
IMP_TEAM_FIELDING_E -0.055349   0.004245 -13.038  < 2e-16 ***
IMP_TEAM_PITCHING_BB -0.005889   0.004937  -1.193   0.2331
IMP_TEAM_PITCHING_H -0.007589   0.002826  -2.685   0.0073 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.04 on 2268 degrees of freedom
Multiple R-squared:  0.3172,    Adjusted R-squared:  0.3151
F-statistic: 150.5 on 7 and 2268 DF,  p-value: < 2.2e-16
```

In Figure 8, we also look at the Adjusted R-Squared statistic to measure how well our model is fitting the actual data. The R-Squared is a measure of how close the data are to the fitted regression line. In general, the higher the R-Squared, the better the model fits the data. However, when dealing to predict human behavior, it is expected that the R-Squared value will be low. The Adjusted R-Squared we got is **0.3151** which is roughly 31% (approx.) of the variance found in the response variable (TARGET_WINS) can be explained by the predictor variables. We notice that even though our Adjusted R-Squared value is below 50%, the predictor variables are statistically significant. Hence, this model was selected as one of the candidates for final selection.

Figure 9 shows the distribution using Q-Q plot and histogram of residuals and we see some evidence of non-normality towards the upper and lower end. The histogram shows that the residuals do not form a bell-shaped which usually present a normally distributed. In my opinion, this distribution is a slightly bimodal shape.

Figure 9: Model 1: Q-Q plot and Histogram of Residuals

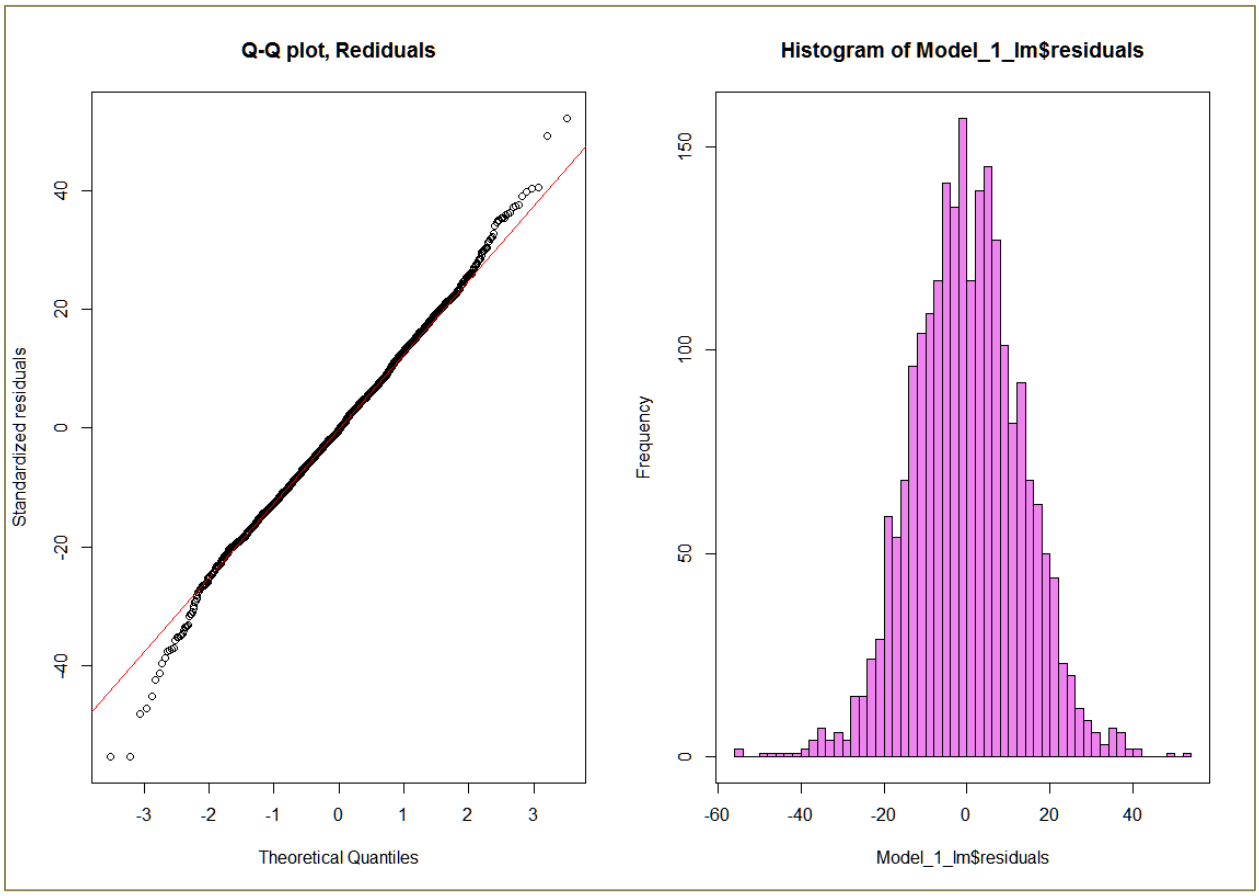
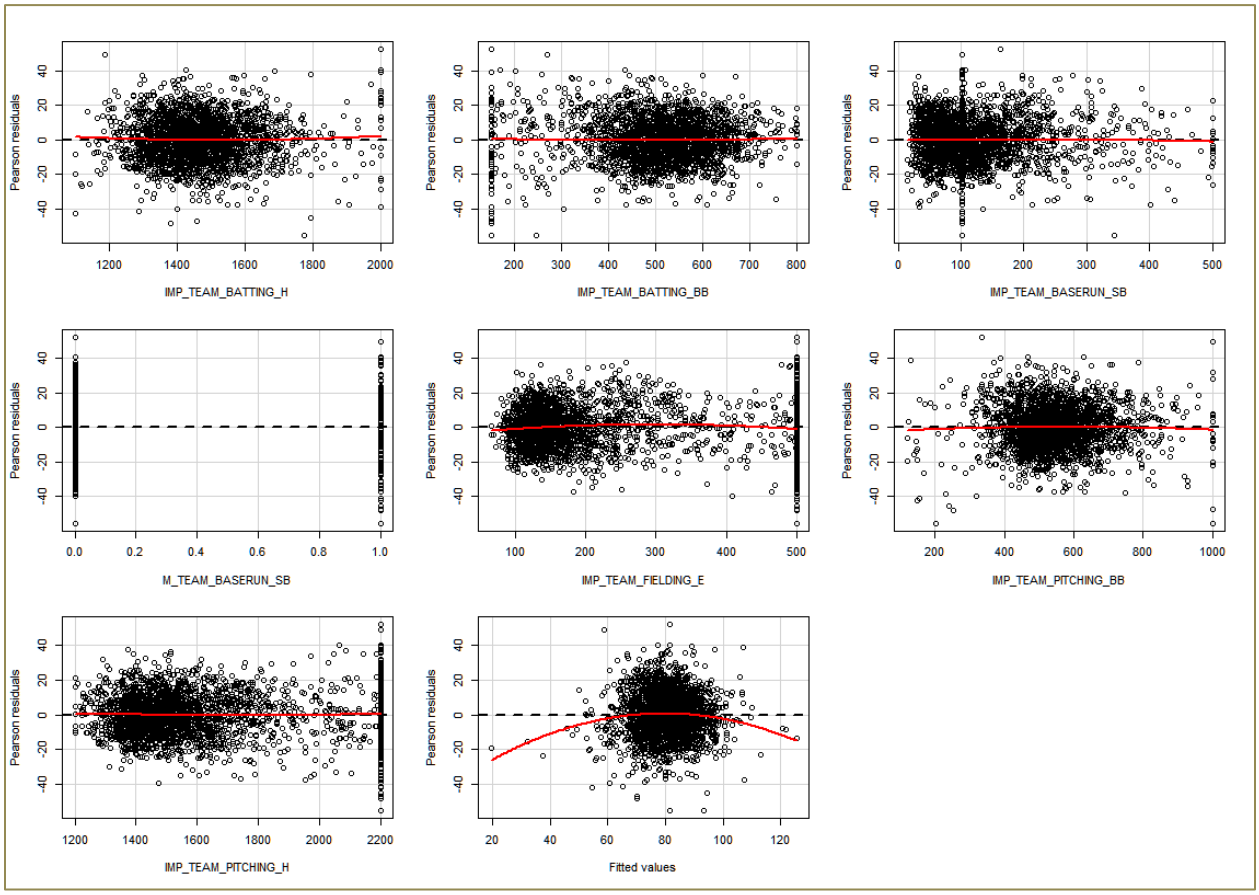


Figure 10 shows the structure of residuals against each predictor variables using scatterplot. We notice that there's no structure in these plots. Hence, we validate the homoscedasticity assumption meaning that the variance of the error term (Residuals) is constant for all combination of independent (predictor) variables.

Figure 10: Model 1: Scatterplot of Residuals and Predictor Variables



Section 3.2: Model 2

Figure 11 shows the output of the Model 2. Residuals in Figure 11 appears to be symmetrical since the Median value is very close to zero (0). We would further investigate this by plotting the residuals in Q-Q plot and histogram to see whether this is normally distributed.

In Model 2, the coefficients of each parameter are also behaving correctly (as per the theoretical effects in Data Dictionary). For instance, while holding the other predictors constant, if there is an increase of one (1) base hit by the batter (IMP_TEAM_BATTING_H), the winning chances increase by 0.0584 which is roughly 5.8% (approx.). Similarly, if a pitcher allowed one (1) additional hit (IMP_TEAM_PITCHING_H), the winning chances decrease by 0.007517 which is roughly 0.75% (approx.). Lastly, if the base is stolen (IMP_TEAM_BASERUN_SB), the winning chances increase by 0.0743 which is roughly 7.43% (approx.).

*** Significance stars in coefficients in Figure 11 shows that the predictor variables are statistically significant and it's unlikely that no relationship exists between the TARGET_WINS and predictor variables.

Figure 11: Model 2: Output

```
Call:
lm(formula = TARGET_WINS ~ IMP_TEAM_BATTING_H + IMP_TEAM_BATTING_BB +
    IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB + IMP_TEAM_FIELDING_E +
    IMP_TEAM_PITCHING_BB + IMP_TEAM_PITCHING_H + IMP_TEAM_PITCHING_HR,
    data = MoneyBall)

Residuals:
    Min      1Q  Median      3Q     Max
-55.344  -8.578  -0.282   8.335  52.521

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -5.872491   3.409313  -1.722  0.08512 .
IMP_TEAM_BATTING_H  0.058421   0.003298  17.713  < 2e-16 ***
IMP_TEAM_BATTING_BB  0.034810   0.005730   6.075 1.45e-09 ***
IMP_TEAM_BASERUN_SB  0.074371   0.004677  15.903  < 2e-16 ***
M_TEAM_BASERUN_SB  28.761232   1.955806  14.706  < 2e-16 ***
IMP_TEAM_FIELDING_E -0.056268   0.004897 -11.490  < 2e-16 ***
IMP_TEAM_PITCHING_BB -0.005635   0.004984  -1.131  0.25836
IMP_TEAM_PITCHING_H -0.007517   0.002833  -2.654  0.00802 **
IMP_TEAM_PITCHING_HR -0.002472   0.006562  -0.377  0.70640
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.04 on 2267 degrees of freedom
Multiple R-squared:  0.3172,    Adjusted R-squared:  0.3148
F-statistic: 131.6 on 8 and 2267 DF,  p-value: < 2.2e-16
```

In Figure 11, we again look at the Adjusted R-Squared statistic to measure how well our model is fitting the actual data. The Adjusted R-Squared we got is **0.3148** which is roughly 31% (approx.) of the variance found in the response variable (TARGET_WINS) can be explained by the predictor variables. We notice that even though our Adjusted R-Squared value is below 50%, the predictor variables are statistically significant. Hence, this model was also selected as one of the candidates for final selection. In Model 2, adding IMP_TEAM_PITCHING_HR seems to slightly decrease the Adjusted R-Squared.

Figure 12 shows the distribution using Q-Q plot and histogram of residuals and we see some evidence of non-normality towards the upper and lower end. The histogram shows that the residuals from Model 2 also does not form a bell-shaped which usually represent a normal distribution. In my opinion, this distribution is a slightly bimodal shape. Also, the distribution seems to be fairly similar to Model 1.

Figure 12: Model 2: Q-Q plot and Histogram of Residuals

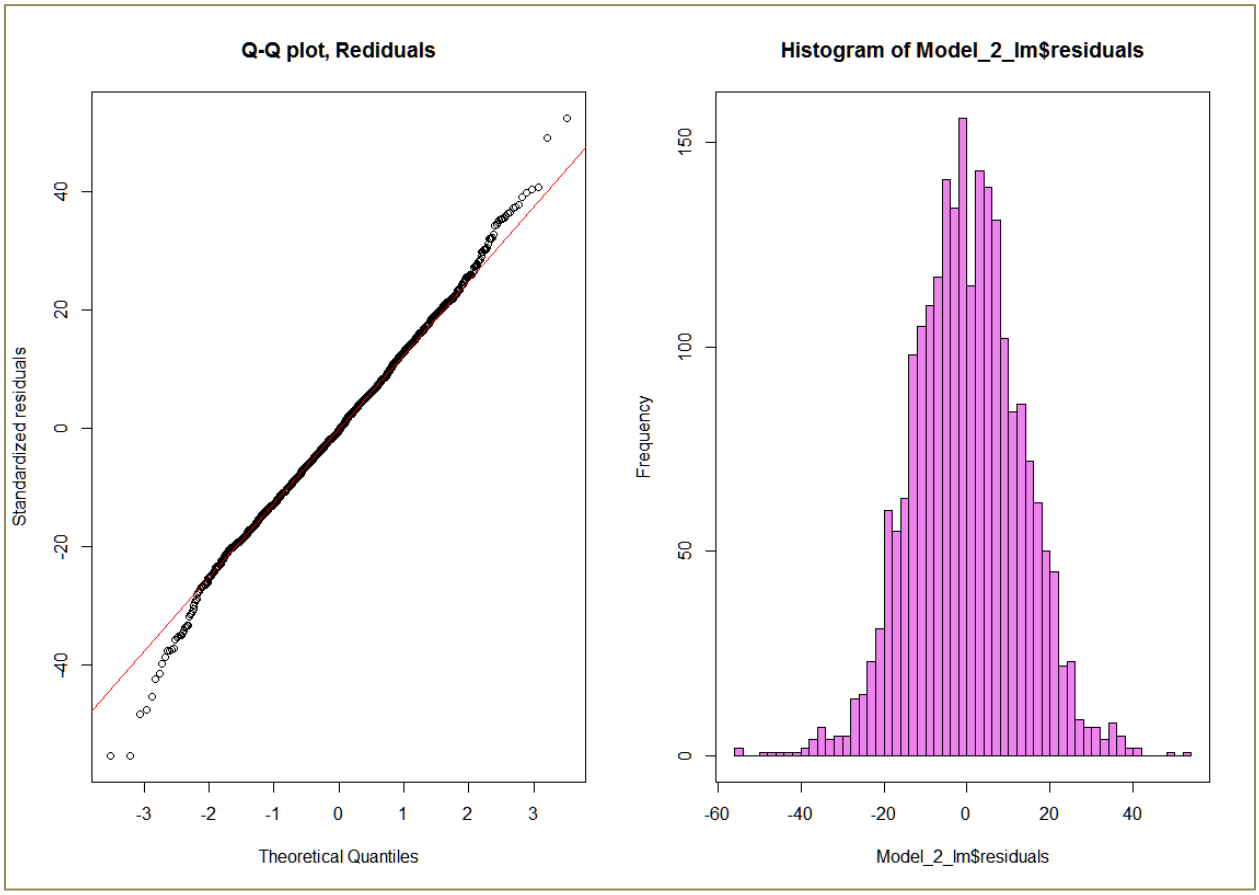
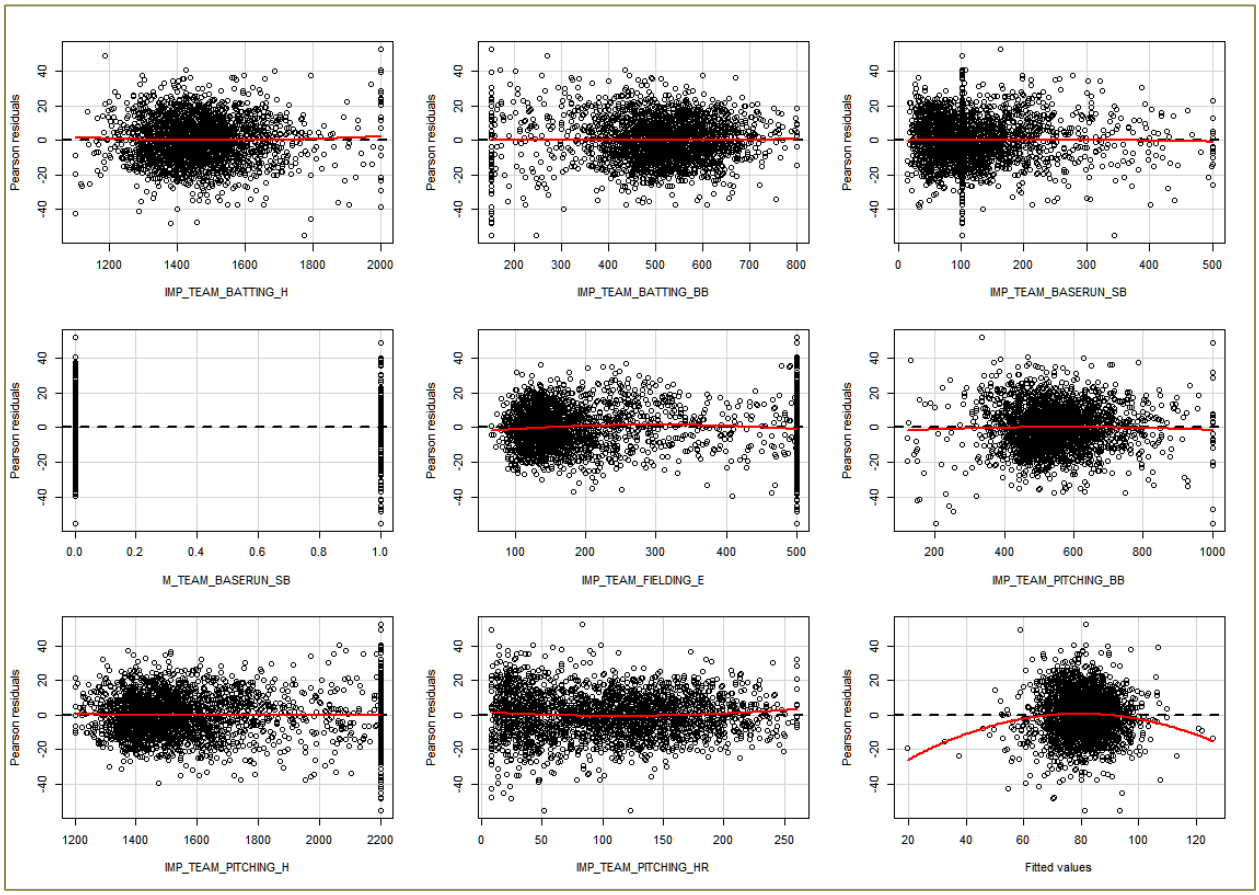


Figure 13 shows the structure of residuals against each predictor variables using scatterplot. We notice that there's no structure in these plots except in one. The plot with IMP_TEAM_PITCHING_HR shows some form of structure. Due to this, we believe that we could not validate the homoscedasticity assumption meaning that the variance of the error term (Residuals) is not constant for all combination of independent (predictor) variables. Now, we can either transform this IMP_TEAM_PITCHING_HR variable or eliminate it from the mode completely and try to refit the model again.

Figure 13: Model 2: Scatterplot of Residuals and Predictor Variables



Section 3.3: Model 3

Figure 14 shows the output of the Model 3. Residuals in Figure 14 appears to be symmetrical since the Median value is very close to zero (0). We would further investigate this by plotting the residuals in Q-Q plot and histogram to see whether this has a normal distribution.

In Model 3, the coefficients of each parameter are also behaving correctly (as per the theoretical effects in Data Dictionary). For instance, while holding the other predictors constant, if there is an increase of one (1) base hit by the batter (IMP_TEAM_BATTING_H), the winning chances increase by 0.06074 which is roughly 6.07% (approx.). Similarly, if a pitcher allowed one (1) additional hit (IMP_TEAM_PITCHING_H), the winning chances decrease by 0.00930 which is roughly 0.93% (approx.). Since we have included another predictor variable (IMP_TEAM_BASERUN_CS) in this mode, let's explain how that variable behaves in this model. While holding all the other variables constant, if the batter is caught stealing (IMP_TEAM_BASERUN_CS) once, the winning chances decrease by 0.03456 which is roughly 3.46% (approx.). Lastly, if the base is stolen (IMP_TEAM_BASERUN_SB), the winning chances increase by 0.07521 which is roughly 7.52% (approx.).

As you may have noticed, we have removed IMP_TEAM_PITCHING_HR predictor since it wasn't helping in Model 2. Instead, we added another predictor variable IMP_TEAM_BASERUN_CS and corresponding flag variable.

*** Significance stars in coefficients in Figure 14 shows that the predictor variables are statistically significant.

Figure 14: Model 3: Output

```
Call:
lm(formula = TARGET_WINS ~ IMP_TEAM_BATTING_H + IMP_TEAM_BATTING_BB +
    IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS +
    M_TEAM_BASERUN_CS + IMP_TEAM_PITCHING_H + IMP_TEAM_PITCHING_BB +
    IMP_TEAM_FIELDING_E, data = MoneyBall)

Residuals:
    Min      1Q  Median      3Q     Max
-56.979  -8.556  -0.275   8.213  59.265

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -3.960326   3.509798  -1.128 0.259286
IMP_TEAM_BATTING_H    0.060744   0.003242  18.737  < 2e-16 ***
IMP_TEAM_BATTING_BB   0.031502   0.005672   5.554 3.11e-08 ***
IMP_TEAM_BASERUN_SB   0.075213   0.004922  15.280  < 2e-16 ***
M_TEAM_BASERUN_SB    30.282484   1.917537  15.792  < 2e-16 ***
IMP_TEAM_BASERUN_CS  -0.034563   0.017396  -1.987 0.047058 *
M_TEAM_BASERUN_CS     6.428316   0.861711   7.460 1.23e-13 ***
IMP_TEAM_PITCHING_H  -0.009303   0.002793  -3.330 0.000882 ***
IMP_TEAM_PITCHING_BB -0.001774   0.004897  -0.362 0.717179
IMP_TEAM_FIELDING_E  -0.073967   0.004755 -15.555  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.85 on 2266 degrees of freedom
Multiple R-squared:  0.3371,    Adjusted R-squared:  0.3344
F-statistic:   128 on 9 and 2266 DF,  p-value: < 2.2e-16
```

In Figure 14, we look at the Adjusted R-Squared statistic to measure how well our model is fitting the actual data. The Adjusted R-Squared we got is **0.3344** which is roughly **33%** (approx.) of the variance found in the response variable (TARGET_WINS) can be explained by the predictor variables. This model by far performs the best. Removing IMP_TEAM_PITCHING_HR and adding IMP_TEAM_BASERUN_CS seems to increase the Adjusted R-Squared value. Earlier in Figure 14, we noticed that coefficients improved as well.

Figure 15 shows the distribution using Q-Q plot and histogram of residuals and we see some evidence of non-normality towards the upper and lower end. The histogram shows that the residuals from Model 3 form a bell-shaped which usually represent a normally distribution. Again, this by far better than both the previous models we have seen. Overall, the residuals look pretty normal distributed.

Figure 15: Model 3: Q-Q plot and Histogram of Residuals

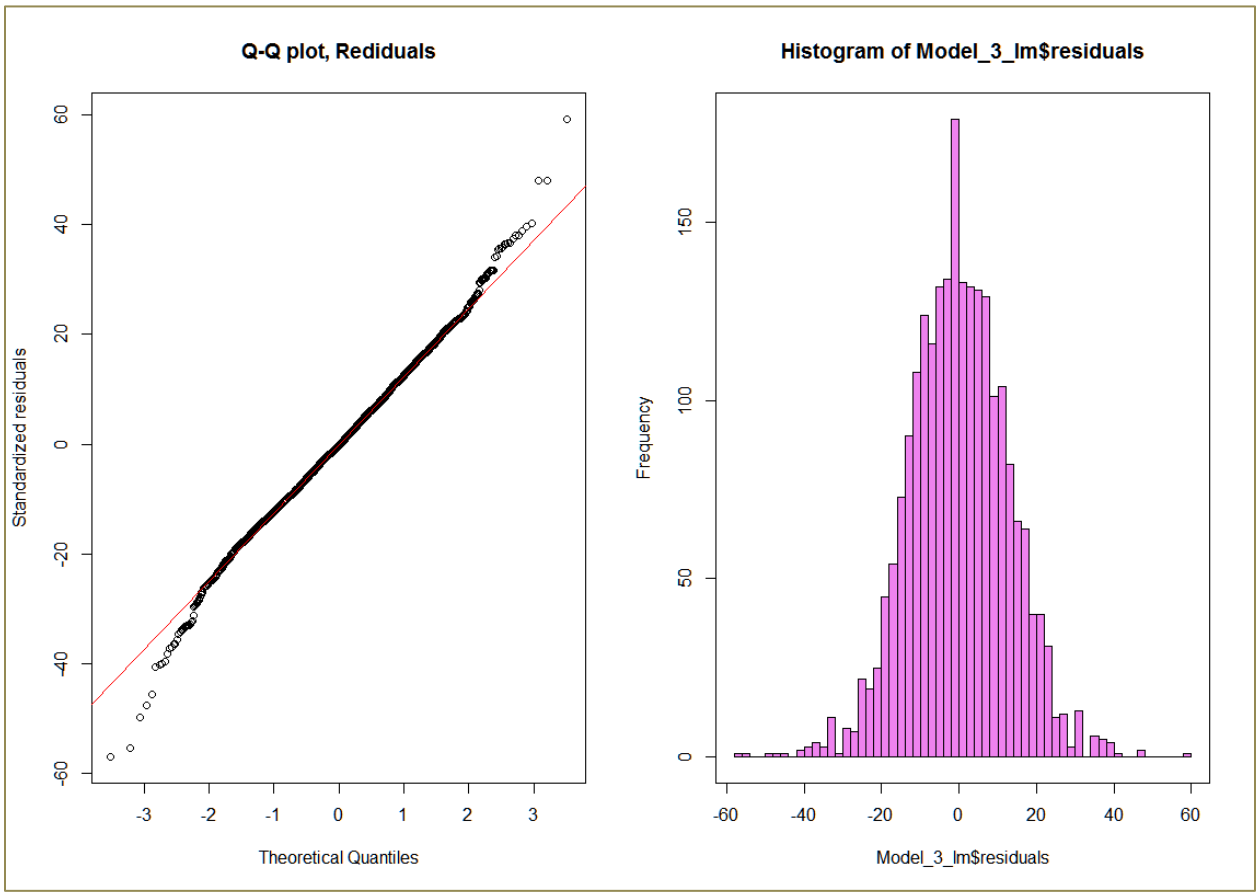
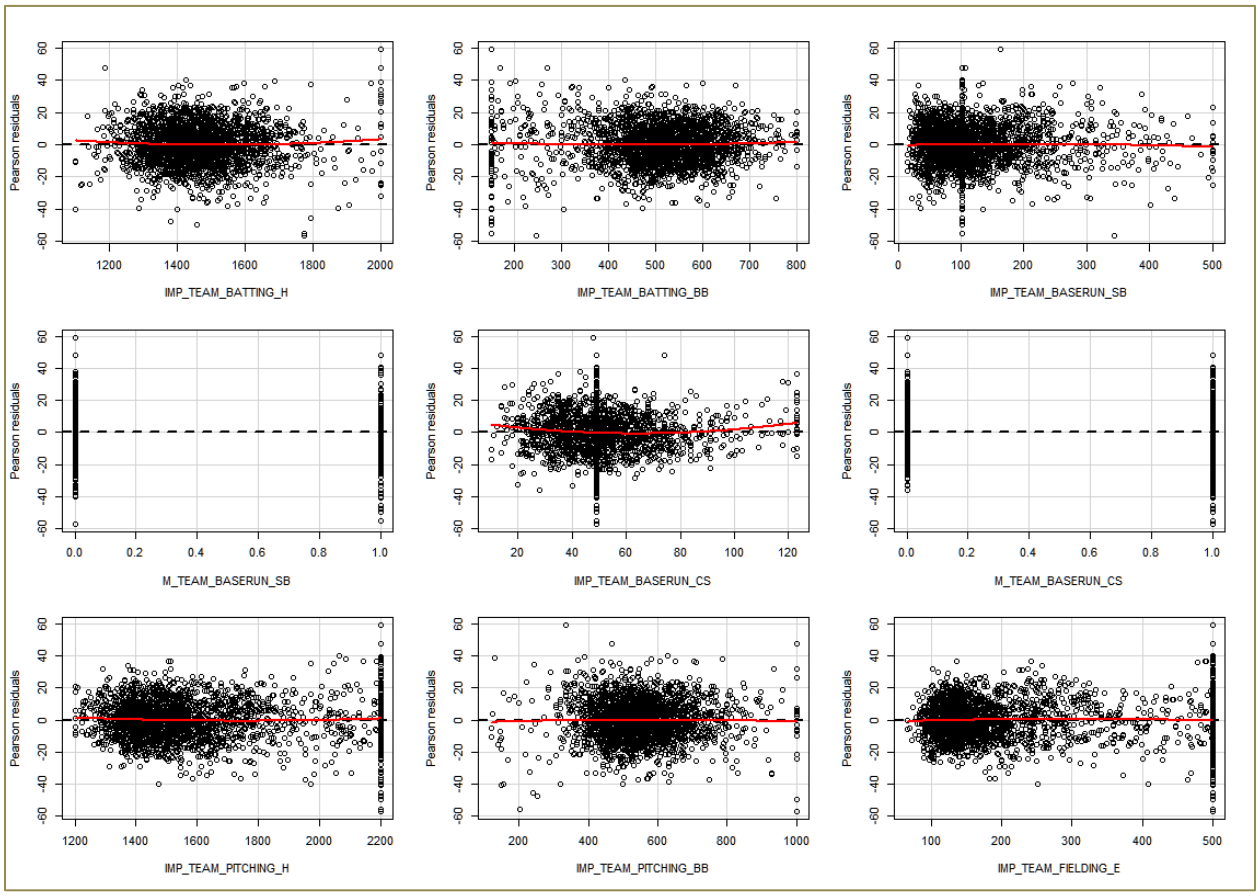
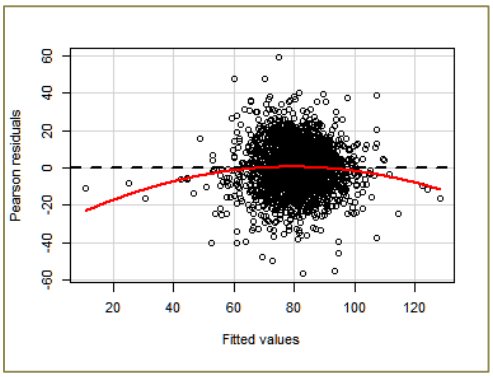


Figure 16 shows the structure of residuals against each predictor variables using scatterplot. We notice that there's no structure in these plots. We validate the homoscedasticity assumption meaning that the variance of the error term (Residuals) is constant for all combination of independent (predictor) variables.

Figure 16: Model 3: Scatterplot of Residuals and Predictor Variables





Section 4: Model Comparison and Selection

Figure 17 shows each of our three models i.e. Model 1, Model 2, and Model 3. Difference between these models is that Model 2 has all the variables from Model 1 and includes an additional variable, IMP_TEAM_PITCHING_HR. Model 3 also has all the variables from Model 1 but includes two additional variables, IMP_TEAM_BASERUN_CS and M_TEAM_BASERUN_CS (flag variable). Model 3 does not include IMP_TEAM_PITCHING_HR.

When comparing our models against Adjusted R-Squared, Model 3 here outperformed all the other models. However, there's no guarantee that Model 3 will be predictive. Hence, we check for the multicollinearity for each to the models to make sure the predictor variables used in these models have no high correlation among themselves. We will also calculate other measures to compare and based on the results, we will select our final model.

Figure 17: Models: Model 1, Model 2, and Model 3

```
> Model_1_call
lm(formula = TARGET_WINS ~ IMP_TEAM_BATTING_H + IMP_TEAM_BATTING_BB +
    IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB + IMP_TEAM_FIELDING_E +
    IMP_TEAM_PITCHING_BB + IMP_TEAM_PITCHING_H, data = MoneyBall)
>
>
> Model_2_call
lm(formula = TARGET_WINS ~ IMP_TEAM_BATTING_H + IMP_TEAM_BATTING_BB +
    IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB + IMP_TEAM_FIELDING_E +
    IMP_TEAM_PITCHING_BB + IMP_TEAM_PITCHING_H + IMP_TEAM_PITCHING_HR,
    data = MoneyBall)
>
>
> Model_3_call
lm(formula = TARGET_WINS ~ IMP_TEAM_BATTING_H + IMP_TEAM_BATTING_BB +
    IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS +
    M_TEAM_BASERUN_CS + IMP_TEAM_PITCHING_H + IMP_TEAM_PITCHING_BB +
    IMP_TEAM_FIELDING_E, data = MoneyBall)
>
```

Figure 18 shows the Variance Inflation Factor (VIF) values of all the models for the predictor variables. The VIF values of the predictor variables indicate the strength of the linear relationship between the variable and remaining predictor variables. A good rule of thumb is that VIF values greater than 10 give some cause for concern. A low VIF value means that there are no high correlations among some or all predictor variables. In Figure 14, we see that all the VIF values are below 7 for Model 1, Model 2, and Model 3. Therefore, multicollinearity is not a problem for these models.

Figure 18: VIF Values: Model 1, Model 2, and Model 3

Predictor Variables	Model 1 VIF Values	Model 2 VIF Values	Model 3 VIF Values
IMP_TEAM_PITCHING_H	6.722779	6.753197	6.760962
IMP_TEAM_BATTING_BB	6.107077	6.110652	6.162474
IMP_TEAM_PITCHING_BB	4.729828	4.818159	4.788496
IMP_TEAM_FIELDING_E	3.962756	5.271085	5.117097
M_TEAM_BASERUN_SB	2.715479	2.777652	2.748805
IMP_TEAM_BATTING_H	2.598941	2.635439	2.621315
IMP_TEAM_BASERUN_SB	2.029005	2.029641	2.314802
IMP_TEAM_PITCHING_HR		2.124017	
M_TEAM_BASERUN_CS			2.293748
IMP_TEAM_BASERUN_CS			1.179531

Figure 19 shows the adjusted R-Squared, AIC, BIC, mean squared error (MSE), and the mean absolute error (MSE) for each of these models i.e. Model 1, Model 2, and Model 3. Based on the results, Model 3 performs better than all the other models by every measure. Model 3 has Adjusted R-Squared is higher, low AIC and BIC values, low Mean Squared Error (MSE), and low Mean Absolute Error (MAE). Hence, we will use Model 3 as our final model to predict the number of wins in the test data.

Figure 19: Metrics: Model 1, Model 2, and Model 3

Models	Adjusted_R_Squared	AIC_Values	BIC_Values	MSE_Values	MAE_Values
Model_1	0.3150547	18157.48	18209.05	169.3583	10.26199
Model_2	0.3147955	18159.34	18216.64	169.3477	10.25872
Model_3	0.3344353	18094.14	18157.17	164.4212	10.02838

Section 5: Model Testing and Scoring

Next, we use our Model 3 to score the MONEYBALL Test data file ("test data"). Prior to running our model, we will clean the test data in a similar way as we cleaned our training dataset. We will fix missing values as well outliers in the data set. Note, a separate protocol document will accompany the stand-alone R program. This document will list the directions on how to import file and acquire results.

Figure 20 shows the summary statistics of the predicted wins. We notice that we have all the observations i.e. 259 observations, and there is not missing values. Our model predicted a minimum of 38 wins for a team and maximum of 111 wins. The average number of wins is about 80 wins. In my opinion, the results seem to be OK. I would have been surprised or alarmed if our model predicted more than 162 wins for any team in the 162 game seasons,

Therefore, we will accept this score data result and use it as our final output.

Figure 20: Moneyball Test Result Stats

Variable	nobs	NAs	Minimum	Maximum	1. Quartile	3. Quartile	Mean	Median	Stdev
P_TARGET_WINS	259	0	38	111	75	86	80.3475	81	9.74376

Conclusion:

In conclusion, I would like to state that all three of my models had a low Adjusted R-Squared value. However, each of them behaved logically correct meaning their coefficients in the model made sense for each parameter. We selected the Model 3 that outperformed the other two models by every measure. We assessed the Goodness-Of-Fit for these models i.e. validate the normality assumption, validate the homoscedasticity assumption (equal variance), and check for multicollinearity. Model 3 passed all the assumptions and had the highest Adjusted R-squared. Hence, it was picked as our final model for prediction.

Lastly, Model 3 states that a team would have a higher number of winning chances if they do the following:

- Bat well and get to more bases (get to 1^{st}, 2^{nd}, 3^{rd}, and 4^{th} base more)
- Get more walks by batters
- Steal more bases
- Do not get caught stealing
- Pitchers allow very minimum hits or no hits
- Pitchers allow very minimum walks or no walks
- There's minimum or no fielding errors.

Appendix I: Model Development R Code

```
#---------------------------------------------------------------------
# Predict 411 - MONEYBALL Data Analysis Project
# Singh, Gurjeet
# 10/14/2017
#---------------------------------------------------------------------

library(readr)
library(car)
library(fBasics)
library(ggplot2)
library(corrplot)

#-------------------------------------------------------------------------------
## 1 - DATA EXPLORATION
#-------------------------------------------------------------------------------

colnames(MoneyBall)[1] <- "INDEX"

#Understand the stats and summary
str(MoneyBall)
summary(MoneyBall)
View(t(basicStats(MoneyBall)))

##Check the distribution of the quantitative variables
par(mfrow = c(4,4))
with(MoneyBall, hist(TARGET_WINS, breaks = "FD", col = "light blue")); box();
with(MoneyBall, hist(TEAM_BATTING_H, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_BATTING_2B, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_BATTING_3B, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_BATTING_HR, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_BATTING_BB, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_BATTING_SO, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_BASERUN_SB, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_BASERUN_CS, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_BATTING_HBP, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_PITCHING_H, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_PITCHING_HR, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_PITCHING_BB, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_PITCHING_SO, breaks = "FD", col = "light blue"));
box();
with(MoneyBall, hist(TEAM_FIELDING_E, breaks = "FD", col = "light blue"));
box();
```

```
with(MoneyBall, hist(TEAM_FIELDING_DP, breaks = "FD", col = "light blue"));
box();
par(mfrow = c(1,1))

#check the normal distribution using QQplot
par(mfrow = c(3,3))
with(MoneyBall, qqPlot(TEAM_BATTING_H, labels=row.names(MoneyBall),
                       main="QQ-Plot Batting H", col = "blue",id.n=3));
with(MoneyBall, qqPlot(TEAM_BATTING_BB, labels=row.names(MoneyBall),
                       main="QQ-Plot Batting BB", col = "blue",id.n=3));
with(MoneyBall, qqPlot(TEAM_BATTING_SO, labels=row.names(MoneyBall),
                       main="QQ-Plot Batting SO", col = "blue",id.n=3));
with(MoneyBall, qqPlot(TEAM_BASERUN_SB, labels=row.names(MoneyBall),
                       main="QQ-Plot Baserun SB", col = "blue",id.n=3));
with(MoneyBall, qqPlot(TEAM_BASERUN_CS, labels=row.names(MoneyBall),
                       main="QQ-Plot Baserun CS", col = "blue",id.n=3));
with(MoneyBall, qqPlot(TEAM_PITCHING_H, labels=row.names(MoneyBall),
                       main="QQ-Plot Pitching H",col = "blue", id.n=3));
with(MoneyBall, qqPlot(TEAM_PITCHING_BB, labels=row.names(MoneyBall),
                       main="QQ-Plot Pitching BB",col = "blue",
id.n=3));
with(MoneyBall, qqPlot(TEAM_PITCHING_SO, labels=row.names(MoneyBall)
                       ,main="QQ-Plot Pitching SO",col = "blue",
id.n=3));
with(MoneyBall, qqPlot(TEAM_FIELDING_E, labels=row.names(MoneyBall),
                       main="QQ-Plot Fielding E",col = "blue", id.n=3));
par(mfrow = c(1,1))

##Examine Relationships
par(mfrow = c(4,3))
with(MoneyBall, plot(TEAM_BATTING_H ,TARGET_WINS,
                       main="TARGET_WINS vs TEAM_BATTING_H",
                       col = "dark orange"))
with(MoneyBall, plot(TEAM_BATTING_BB ,TARGET_WINS,
                     main="TARGET_WINS vs TEAM_BATTING_BB",
                     col = "dark orange"))
with(MoneyBall, plot(TEAM_BATTING_SO,TARGET_WINS ,
                     main="TARGET_WINS vs TEAM_BATTING_SO",
                     col = "dark orange"))
with(MoneyBall, plot(TEAM_BASERUN_SB,TARGET_WINS ,
                     main="TARGET_WINS vs TEAM_BASERUN_SB",
                     col = "dark orange"))
with(MoneyBall, plot(TEAM_BASERUN_CS,TARGET_WINS ,
                     main="TARGET_WINS vs TEAM_BASERUN_CS",
                     col = "dark orange"))
with(MoneyBall, plot(TEAM_PITCHING_H ,TARGET_WINS,
                     main="TARGET_WINS vs TEAM_PITCHING_H",
                     col = "dark orange"))
with(MoneyBall, plot(TEAM_PITCHING_BB,TARGET_WINS ,
                     main="TARGET_WINS vs TEAM_PITCHING_BB",
                     col = "dark orange"))
with(MoneyBall, plot(TEAM_PITCHING_SO,TARGET_WINS ,
                     main="TARGET_WINS vs TEAM_PITCHING_SO",
                     col = "dark orange"))
with(MoneyBall, plot(TEAM_FIELDING_E,TARGET_WINS,
                     main="TARGET_WINS vs TEAM_FIELDING_E",
                     col = "dark orange"))
```

```
with(MoneyBall, plot(TEAM_FIELDING_DP ,TARGET_WINS,
                           main="TARGET_WINS vs TEAM_FIELDING_DP",
                           col = "dark orange"))
par(mfrow = c(1,1))

#-------------------------------------------------------------------------------
## 2 - DATA PREPARATION
#-------------------------------------------------------------------------------
#---------------------------------------
##clean missing values with median values
#---------------------------------------

summary(MoneyBall)
##clean missing values with median values

MoneyBall$IMP_TEAM_BATTING_SO <-ifelse(is.na(MoneyBall$TEAM_BATTING_SO),
                                       752,
                                       MoneyBall$TEAM_BATTING_SO)
MoneyBall$M_TEAM_BATTING_SO <- ifelse(is.na(MoneyBall$TEAM_BATTING_SO),
                                       1, 0)

MoneyBall$IMP_TEAM_BASERUN_SB <- ifelse(is.na(MoneyBall$TEAM_BASERUN_SB),
                                       101,
                                       MoneyBall$TEAM_BASERUN_SB)
MoneyBall$M_TEAM_BASERUN_SB <- ifelse(is.na(MoneyBall$TEAM_BASERUN_SB),
                                       1, 0)

MoneyBall$IMP_TEAM_BASERUN_CS <- ifelse(is.na(MoneyBall$TEAM_BASERUN_CS),
                                       49,
                                       MoneyBall$TEAM_BASERUN_CS)
MoneyBall$M_TEAM_BASERUN_CS <- ifelse(is.na(MoneyBall$TEAM_BASERUN_CS),
                                       1, 0)

MoneyBall$IMP_TEAM_PITCHING_SO <- ifelse(is.na(MoneyBall$TEAM_PITCHING_SO),
                                       814,
                                       MoneyBall$TEAM_PITCHING_SO)
MoneyBall$M_TEAM_PITCHING_SO <- ifelse(is.na(MoneyBall$TEAM_PITCHING_SO),
                                       1, 0)

MoneyBall$IMP_TEAM_FIELDING_DP <- ifelse(is.na(MoneyBall$TEAM_FIELDING_DP),
                                       149,
                                       MoneyBall$TEAM_FIELDING_DP)
MoneyBall$M_TEAM_FIELDING_DP <- ifelse(is.na(MoneyBall$TEAM_FIELDING_DP),
                                        1, 0)


#---------------------------------
##Checking Correlation
#---------------------------------

#selectiong variables for correlation list
corr.list <- c('TARGET_WINS','TEAM_BATTING_H','TEAM_BATTING_BB',
               'IMP_TEAM_BATTING_SO','IMP_TEAM_BASERUN_SB',
               'IMP_TEAM_BASERUN_CS','TEAM_PITCHING_H',
               'TEAM_PITCHING_BB','IMP_TEAM_PITCHING_SO',
               'TEAM_FIELDING_E','IMP_TEAM_FIELDING_DP')
```

```
#Checking correlation only for the variables above.
MoneyBall.Corr <- MoneyBall[,(names(MoneyBall) %in% corr.list )]

corrplot(cor(MoneyBall.Corr), method="number")

#---------------------------------
##creating a 1st and 99th percentile
#---------------------------------

percentile.val <- matrix(with(MoneyBall,
                              c("TEAM_BATTING_H",
                                    min(TEAM_BATTING_H),
                                    max(TEAM_BATTING_H),
                                    quantile(TEAM_BATTING_H, (0.01))[[1]],
                                    quantile(TEAM_BATTING_H, (0.99))[[1]]
)),
                              ncol = 5, nrow = 1)

colnames(percentile.val) <- c("FieldName","Min","Max","1-Percentile",
                              "99-Percentile")

percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                              c("TEAM_BATTING_2B",
                                    min(TEAM_BATTING_2B),
                                    max(TEAM_BATTING_2B),
                                    quantile(TEAM_BATTING_2B, (0.01))[[1]],
                                    quantile(TEAM_BATTING_2B, (0.99))[[1]]
)),
                                ncol = 5, nrow = 1))

percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                              c("TEAM_BATTING_3B",
                                    min(TEAM_BATTING_3B),
                                    max(TEAM_BATTING_3B),
                                    quantile(TEAM_BATTING_3B, (0.01))[[1]],
                                    quantile(TEAM_BATTING_3B, (0.99))[[1]]
)),
                                ncol = 5, nrow = 1))

percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                              c("TEAM_BATTING_HR",
                                    min(TEAM_BATTING_HR),
                                    max(TEAM_BATTING_HR),
                                    quantile(TEAM_BATTING_HR, (0.01))[[1]],
                                    quantile(TEAM_BATTING_HR, (0.99))[[1]]
)),
                                ncol = 5, nrow = 1))

percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                              c("TEAM_BATTING_BB",
                                    min(TEAM_BATTING_BB),
                                    max(TEAM_BATTING_BB),
                                    quantile(TEAM_BATTING_BB, (0.01))[[1]],
```

```
                                        quantile(TEAM_BATTING_BB, (0.99))[[1]]
)),
                                    ncol = 5, nrow = 1))

percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                    c("IMP_TEAM_BATTING_SO",
                                        min(IMP_TEAM_BATTING_SO),
                                        max(IMP_TEAM_BATTING_SO),
                                        quantile(IMP_TEAM_BATTING_SO,
(0.01))[[1]],
                                        quantile(IMP_TEAM_BATTING_SO,
(0.99))[[1]] )),
                                    ncol = 5, nrow = 1))


percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                    c("IMP_TEAM_BASERUN_SB",
                                        min(IMP_TEAM_BASERUN_SB),
                                        max(IMP_TEAM_BASERUN_SB),
                                        quantile(IMP_TEAM_BASERUN_SB,
(0.01))[[1]],
                                        quantile(IMP_TEAM_BASERUN_SB,
(0.99))[[1]] )),
                                    ncol = 5, nrow = 1))


percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                    c("IMP_TEAM_BASERUN_CS",
                                        min(IMP_TEAM_BASERUN_CS),
                                        max(IMP_TEAM_BASERUN_CS),
                                        quantile(IMP_TEAM_BASERUN_CS,
(0.01))[[1]],
                                        quantile(IMP_TEAM_BASERUN_CS,
(0.99))[[1]] )),
                                    ncol = 5, nrow = 1))


percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                      c("TEAM_PITCHING_H",
                                        min(TEAM_PITCHING_H),
                                        max(TEAM_PITCHING_H),
                                        quantile(TEAM_PITCHING_H, (0.01))[[1]],
                                        quantile(TEAM_PITCHING_H, (0.99))[[1]]
)),
                                   ncol = 5, nrow = 1))


percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                      c("TEAM_PITCHING_HR",
                                        min(TEAM_PITCHING_HR),
                                        max(TEAM_PITCHING_HR),
```

```
                                        quantile(TEAM_PITCHING_HR,
(0.01))[[1]],
                                        quantile(TEAM_PITCHING_HR, (0.99))[[1]]
)),
                                ncol = 5, nrow = 1))


percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                    c("TEAM_PITCHING_BB",
                                      min(TEAM_PITCHING_BB),
                                      max(TEAM_PITCHING_BB),
                                      quantile(TEAM_PITCHING_BB,
(0.01))[[1]],
                                      quantile(TEAM_PITCHING_BB,
(0.99))[[1]] )),
                               ncol = 5, nrow = 1))

percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                    c("IMP_TEAM_PITCHING_SO",
                                      min(IMP_TEAM_PITCHING_SO),
                                      max(IMP_TEAM_PITCHING_SO),
                                      quantile(IMP_TEAM_PITCHING_SO,
(0.01))[[1]],
                                      quantile(IMP_TEAM_PITCHING_SO,
(0.99))[[1]] )),
                                ncol = 5, nrow = 1))

percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                    c("TEAM_FIELDING_E",
                                      min(TEAM_FIELDING_E),
                                      max(TEAM_FIELDING_E),
                                      quantile(TEAM_FIELDING_E, (0.01))[[1]],
                                      quantile(TEAM_FIELDING_E, (0.99))[[1]]
)),
                                ncol = 5, nrow = 1))

percentile.val <- rbind(percentile.val,
                        matrix(with(MoneyBall,
                                    c("IMP_TEAM_FIELDING_DP",
                                      min(IMP_TEAM_FIELDING_DP),
                                      max(IMP_TEAM_FIELDING_DP),
                                      quantile(IMP_TEAM_FIELDING_DP,
(0.01))[[1]],
                                      quantile(IMP_TEAM_FIELDING_DP,
(0.99))[[1]] )),
                                ncol = 5, nrow = 1))

View(percentile.val)
```

```
#-------------------------------------------------------------------------
##Cleaning outliers
#-------------------------------------------------------------------------

#--1
MoneyBall$IMP_TEAM_BATTING_H <- with(MoneyBall,
                                     ifelse(TEAM_BATTING_H < 1100, 1100,
                                          ifelse(TEAM_BATTING_H > 2000,
2000,
                                                 TEAM_BATTING_H)));

#--2
MoneyBall$IMP_TEAM_BATTING_BB <- with(MoneyBall, ifelse(TEAM_BATTING_BB <
150, 150,
                                     ifelse(TEAM_BATTING_BB > 800, 800,

TEAM_BATTING_BB)));

#--3
MoneyBall$IMP_TEAM_BATTING_SO <- with(MoneyBall, ifelse(IMP_TEAM_BATTING_SO <
72, 72,
                                     ifelse(IMP_TEAM_BATTING_SO > 1350,
1350,

IMP_TEAM_BATTING_SO)));

#--4
MoneyBall$IMP_TEAM_BASERUN_SB <- with(MoneyBall, ifelse(IMP_TEAM_BASERUN_SB <
14, 14,
                                     ifelse(IMP_TEAM_BASERUN_SB > 500,
500,

IMP_TEAM_BASERUN_SB)));

#--5
MoneyBall$IMP_TEAM_BASERUN_CS <- with(MoneyBall, ifelse(IMP_TEAM_BASERUN_CS <
10, 10,
                                     ifelse(IMP_TEAM_BASERUN_CS > 123,
123,

IMP_TEAM_BASERUN_CS)));

#--6
MoneyBall$IMP_TEAM_PITCHING_H <- with(MoneyBall, ifelse(TEAM_PITCHING_H <
1200, 1200,
                                     ifelse(TEAM_PITCHING_H > 2200,
2200,

TEAM_PITCHING_H)));
#--7
MoneyBall$IMP_TEAM_PITCHING_HR <- with(MoneyBall, ifelse(TEAM_PITCHING_HR <
8, 8,
                                     ifelse(TEAM_PITCHING_HR > 260, 260,

TEAM_PITCHING_HR)));
#--8
```

```
MoneyBall$IMP_TEAM_PITCHING_BB <- with(MoneyBall, ifelse(TEAM_PITCHING_BB <
119, 119,
                                                  ifelse(TEAM_PITCHING_BB > 1000,
1000,

TEAM_PITCHING_BB)));
#--9
MoneyBall$IMP_TEAM_PITCHING_SO <- with(MoneyBall, ifelse(IMP_TEAM_PITCHING_SO
< 241, 241,
                                                  ifelse(IMP_TEAM_PITCHING_SO >
1700, 1700,

IMP_TEAM_PITCHING_SO)));
#--10
MoneyBall$IMP_TEAM_FIELDING_E <- with(MoneyBall, ifelse(TEAM_FIELDING_E < 65,
65,
                                                  ifelse(TEAM_FIELDING_E > 500, 500,

TEAM_FIELDING_E)));
#--11
MoneyBall$IMP_TEAM_FIELDING_DP <- with(MoneyBall, ifelse(IMP_TEAM_FIELDING_DP
< 71, 71,
                                                   ifelse(IMP_TEAM_FIELDING_DP >
220, 220,

IMP_TEAM_FIELDING_DP)));

#options(scipen = 111)
View(t(basicStats(MoneyBall)))


###-------------------------------------------------------------------------
### Re-checking distributions and relationship after fixing outliers and
#missing values
###-------------------------------------------------------------------------


##Check the distribution of the quantitative variables
par(mfrow = c(4,3))
with(MoneyBall, hist(TARGET_WINS, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_BATTING_H, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_BATTING_BB, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_BATTING_SO, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_BASERUN_SB, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_BASERUN_CS, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_PITCHING_H, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_PITCHING_HR, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_PITCHING_BB, breaks = "FD",
                     col = "light blue")); box();
```

```
with(MoneyBall, hist(IMP_TEAM_PITCHING_SO, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_FIELDING_E, breaks = "FD",
                     col = "light blue")); box();
with(MoneyBall, hist(IMP_TEAM_FIELDING_DP, breaks = "FD",
                     col = "light blue")); box();
par(mfrow = c(1,1))

##Examine Relationships
par(mfrow = c(4,3))
with(MoneyBall, plot(IMP_TEAM_BATTING_H ,TARGET_WINS,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_BATTING_BB ,TARGET_WINS,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_BATTING_SO,TARGET_WINS ,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_BASERUN_SB,TARGET_WINS ,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_BASERUN_CS,TARGET_WINS ,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_PITCHING_H ,TARGET_WINS,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_PITCHING_BB,TARGET_WINS ,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_PITCHING_SO,TARGET_WINS ,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_FIELDING_E,TARGET_WINS,
                     col = "dark orange"))
with(MoneyBall, plot(IMP_TEAM_FIELDING_DP ,TARGET_WINS,
                     col = "dark orange"))
par(mfrow = c(1,1))

#----------------------------
##creating a drop list
#----------------------------

#creating a drop list to remove not required variables.
drop.list <- c('INDEX','TEAM_BATTING_HBP','TEAM_BATTING_SO',
               'TEAM_BATTING_BB','TEAM_BASERUN_SB','TEAM_BASERUN_CS',
               'TEAM_PITCHING_SO','TEAM_FIELDING_DP','TEAM_BATTING_H',
               'TEAM_BATTING_2B','TEAM_BATTING_3B', 'TEAM_BATTING_HR',
               'TEAM_PITCHING_H','TEAM_PITCHING_HR','TEAM_PITCHING_BB'
              ,'TEAM_FIELDING_E')

#droping the variables
MoneyBall <- MoneyBall[,!(names(MoneyBall) %in% drop.list )]

View(t(basicStats(MoneyBall)))
```

```
#-------------------------------------------------------------------------------
## 3- Build Models
#-------------------------------------------------------------------------------

#------------------------------
##Model_1_lm
#------------------------------
names(MoneyBall)

##Adjusted R-squared:     0.3151
Model_1_lm <- lm(TARGET_WINS ~ IMP_TEAM_BATTING_H  +
                               IMP_TEAM_BATTING_BB +
                               IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB +
                               IMP_TEAM_FIELDING_E +
                               IMP_TEAM_PITCHING_BB +
                               IMP_TEAM_PITCHING_H
                             ,data=MoneyBall)

summary(Model_1_lm)

#------------------------------
##Model_1_lm - Assessing the Goodness-Of-Fit in OLS Regression
#------------------------------

# Validating the normality assumption:
par(mfrow = c(1,2))
#Creating 2 Q-Q plots to evaluate the distribution of
# SalePrice and L_SalePrice
qqnorm(Model_1_lm$residuals, main = "Q-Q plot, Rediduals",
       xlab="Theoretical Quantiles", col = "black",
       ylab="Standardized residuals",datax=FALSE)

qqline(Model_1_lm$residuals, datax=FALSE, distribution=qnorm,
       probs=c(0.25,0.75),qtype=7, col = "red")

hist(Model_1_lm$residuals, breaks = "FD", col = "violet"); box();
par(mfrow = c(1,1))

#Validating the homoscedasticity assumption (equal variance):
residualPlots(Model_1_lm)
    # par(mfrow = c(1,1))
    # residualPlot(Model_1_lm)
```

```
#------------------------------
##Model_2_lm
#------------------------------
names(MoneyBall)

##Adjusted R-squared:     0.3148
Model_2_lm <- lm(TARGET_WINS ~ IMP_TEAM_BATTING_H  +
                               IMP_TEAM_BATTING_BB +
                               IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB +
                               IMP_TEAM_FIELDING_E +
                               IMP_TEAM_PITCHING_BB +
                               IMP_TEAM_PITCHING_H +
                               IMP_TEAM_PITCHING_HR
                             ,data=MoneyBall)

summary(Model_2_lm)

#------------------------------
##Model_2_lm - Assessing the Goodness-Of-Fit in OLS Regression
#------------------------------

# Validating the normality assumption:
par(mfrow = c(1,2))
#Creating 2 Q-Q plots to evaluate the distribution of
# SalePrice and L_SalePrice
qqnorm(Model_2_lm$residuals, main = "Q-Q plot, Rediduals",
       xlab="Theoretical Quantiles", col = "black",
       ylab="Standardized residuals",datax=FALSE)

qqline(Model_2_lm$residuals, datax=FALSE, distribution=qnorm,
       probs=c(0.25,0.75),qtype=7, col = "red")

hist(Model_2_lm$residuals, breaks = "FD", col = "violet"); box();
par(mfrow = c(1,1))

#Validating the homoscedasticity assumption (equal variance):
residualPlots(Model_2_lm)
par(mfrow = c(1,1))
residualPlot(Model_2_lm)
```

```
#------------------------------
##Model_3_lm
#------------------------------
names(MoneyBall)

##Adjusted R-squared:     0.3344
Model_3_lm <- lm(TARGET_WINS ~ IMP_TEAM_BATTING_H  +
                               IMP_TEAM_BATTING_BB +
                               IMP_TEAM_BASERUN_SB + M_TEAM_BASERUN_SB +
                               IMP_TEAM_BASERUN_CS + M_TEAM_BASERUN_CS +
                               IMP_TEAM_PITCHING_H +
                               IMP_TEAM_PITCHING_BB +
                               IMP_TEAM_FIELDING_E
                             ,data=MoneyBall)

summary(Model_3_lm)

#------------------------------
##Model_3_lm - Assessing the Goodness-Of-Fit in OLS Regression
#------------------------------

# Validating the normality assumption:
par(mfrow = c(1,2))
#Creating 2 Q-Q plots to evaluate the distribution of
# SalePrice and L_SalePrice
qqnorm(Model_3_lm$residuals, main = "Q-Q plot, Rediduals",
       xlab="Theoretical Quantiles", col = "black",
       ylab="Standardized residuals",datax=FALSE)

qqline(Model_3_lm$residuals, datax=FALSE, distribution=qnorm,
       probs=c(0.25,0.75),qtype=7, col = "red")

hist(Model_3_lm$residuals, breaks = "FD", col = "violet"); box();
par(mfrow = c(1,1))

#Validating the homoscedasticity assumption (equal variance):
residualPlots(Model_3_lm)
    # par(mfrow = c(1,1))
    # residualPlot(Model_3_lm)
```

```
#-------------------------------------------------------------------------
-
## 4- SELECT MODELS - Predictive Accuracy
#-------------------------------------------------------------------------
-

#extract the model information from summary output
Model_1_call <- Model_1_lm$call
Model_2_call <- Model_2_lm$call
Model_3_call <- Model_3_lm$call

#Printing the each model.
Model_1_call

Model_2_call

Model_3_call


# Compute the VIF values
library(car)
Model_1_lm.VIF <- as.matrix(sort(vif(Model_1_lm),decreasing=TRUE))
Model_2_lm.VIF <- as.matrix(sort(vif(Model_2_lm),decreasing=TRUE))
Model_3_lm.VIF <- as.matrix(sort(vif(Model_3_lm),decreasing=TRUE))


colnames(Model_1_lm.VIF) <- "VIF_Values"
colnames(Model_2_lm.VIF) <- "VIF_Values"
colnames(Model_3_lm.VIF) <- "VIF_Values"


View(Model_1_lm.VIF)
View(Model_2_lm.VIF)
View(Model_3_lm.VIF)

##MSE
mse.Model_1_lm <- mean(Model_1_lm$residuals^2)
mse.Model_2_lm <- mean(Model_2_lm$residuals^2)
mse.Model_3_lm <- mean(Model_3_lm$residuals^2)


##MAE
mae.Model_1_lm <- mean(abs(Model_1_lm$residuals))
mae.Model_2_lm <- mean(abs(Model_2_lm$residuals))
mae.Model_3_lm <- mean(abs(Model_3_lm$residuals))



##Creating a Table to include all the metrics
rsqrd.Mat <- matrix(c(summary(Model_1_lm)$adj.r.squared,
                      summary(Model_2_lm)$adj.r.squared,
                      summary(Model_3_lm)$adj.r.squared),
                    ncol = 1)

rownames(rsqrd.Mat) <- c("Model_1_lm", "Model_2_lm", "Model_3_lm")
colnames(rsqrd.Mat) <- "Adjusted_R_Squared"
```

```
AIC.Mat <- matrix(c(AIC(Model_1_lm),
                    AIC(Model_2_lm),
                    AIC(Model_3_lm)),
                  ncol = 1)
rownames(AIC.Mat) <- c("Model_1_lm", "Model_2_lm", "Model_3_lm")
colnames(AIC.Mat) <- "AIC_Values"


BIC.Mat <- matrix(c(BIC(Model_1_lm),
                    BIC(Model_2_lm),
                    BIC(Model_3_lm)),
                  ncol = 1)
rownames(BIC.Mat) <-  c("Model_1_lm", "Model_2_lm", "Model_3_lm")
colnames(BIC.Mat) <- "BIC_Values"


MSE.Mat <- matrix(c(mse.Model_1_lm,
                    mse.Model_2_lm,
                    mse.Model_3_lm),
                  ncol = 1)
rownames(MSE.Mat) <-  c("Model_1_lm", "Model_2_lm", "Model_3_lm")
colnames(MSE.Mat) <- "MSE_Values"

MAE.Mat <- matrix(c(mae.Model_1_lm,
                    mae.Model_2_lm,
                    mae.Model_3_lm),
                  ncol = 1)
rownames(MAE.Mat) <-  c("Model_1_lm", "Model_2_lm", "Model_3_lm")
colnames(MAE.Mat) <- "MAE_Values"


final.table  <- cbind( rsqrd.Mat,
                       AIC.Mat,
                       BIC.Mat,
                       MSE.Mat,
                       MAE.Mat)

View(final.table)
```

Appendix II: Stand-Alone R Code

```
#-----------------------------------------------------------------------
# Predict 411 - MONEYBALL Data Analysis Project
# Singh, Gurjeet
# 10/14/2017
# Stand-Alone program
#-----------------------------------------------------------------------

library(readr)
library(car)
library(fBasics)
library(ggplot2)
library(corrplot)


#-------------------------------------------------------------------------------
## 1 - Importing a Test File and check import
#-------------------------------------------------------------------------------

summary(MoneyBall_Test)

colnames(MoneyBall_Test)[1] <- "INDEX"

#-------------------------------------------------------------------------------
## 2 - DATA PREPARATION
#-------------------------------------------------------------------------------

#----------------------------------
##clean missing values with median values
#----------------------------------

MoneyBall_Test$IMP_TEAM_BATTING_SO <-
ifelse(is.na(MoneyBall_Test$TEAM_BATTING_SO),
                                          752,
                                          MoneyBall_Test$TEAM_BATTING_SO)
MoneyBall_Test$M_TEAM_BATTING_SO <-
ifelse(is.na(MoneyBall_Test$TEAM_BATTING_SO),
                                        1, 0)

MoneyBall_Test$IMP_TEAM_BASERUN_SB <-
ifelse(is.na(MoneyBall_Test$TEAM_BASERUN_SB),
                                           101,
                                           MoneyBall_Test$TEAM_BASERUN_SB)
MoneyBall_Test$M_TEAM_BASERUN_SB <-
ifelse(is.na(MoneyBall_Test$TEAM_BASERUN_SB),
                                        1, 0)

MoneyBall_Test$IMP_TEAM_BASERUN_CS <-
ifelse(is.na(MoneyBall_Test$TEAM_BASERUN_CS),
                                           49,
                                           MoneyBall_Test$TEAM_BASERUN_CS)
MoneyBall_Test$M_TEAM_BASERUN_CS <-
ifelse(is.na(MoneyBall_Test$TEAM_BASERUN_CS),
                                        1, 0)
```

```
MoneyBall_Test$IMP_TEAM_PITCHING_SO <-
ifelse(is.na(MoneyBall_Test$TEAM_PITCHING_SO),
                                          814,

MoneyBall_Test$TEAM_PITCHING_SO)
MoneyBall_Test$M_TEAM_PITCHING_SO <-
ifelse(is.na(MoneyBall_Test$TEAM_PITCHING_SO),
                                        1, 0)

MoneyBall_Test$IMP_TEAM_FIELDING_DP <-
ifelse(is.na(MoneyBall_Test$TEAM_FIELDING_DP),
                                          149,

MoneyBall_Test$TEAM_FIELDING_DP)
MoneyBall_Test$M_TEAM_FIELDING_DP <-
ifelse(is.na(MoneyBall_Test$TEAM_FIELDING_DP),
                                        1, 0)

#----------------------------------
##Cleaning outliers
#----------------------------------

#--1
MoneyBall_Test$IMP_TEAM_BATTING_H <- with(MoneyBall_Test,
                                  ifelse(TEAM_BATTING_H < 1100, 1100,
                                         ifelse(TEAM_BATTING_H > 2000, 2000,
                                                TEAM_BATTING_H)));

#--2
MoneyBall_Test$IMP_TEAM_BATTING_BB <- with(MoneyBall_Test,
ifelse(TEAM_BATTING_BB < 150, 150,
                                                 ifelse(TEAM_BATTING_BB >
800, 800,

TEAM_BATTING_BB)));

#--3
MoneyBall_Test$IMP_TEAM_BATTING_SO <- with(MoneyBall_Test,
ifelse(IMP_TEAM_BATTING_SO < 72, 72,
                                                ifelse(IMP_TEAM_BATTING_SO
> 1350, 1350,

IMP_TEAM_BATTING_SO)));

#--4
MoneyBall_Test$IMP_TEAM_BASERUN_SB <- with(MoneyBall_Test,
ifelse(IMP_TEAM_BASERUN_SB < 14, 14,
                                                ifelse(IMP_TEAM_BASERUN_SB
> 500, 500,

IMP_TEAM_BASERUN_SB)));

#--5
MoneyBall_Test$IMP_TEAM_BASERUN_CS <- with(MoneyBall_Test,
ifelse(IMP_TEAM_BASERUN_CS < 10, 10,

ifelse(IMP_TEAM_BASERUN_CS > 123, 123,
```

```
IMP_TEAM_BASERUN_CS)));

#--6
MoneyBall_Test$IMP_TEAM_PITCHING_H <- with(MoneyBall_Test,
ifelse(TEAM_PITCHING_H < 1200, 1200,
                                                    ifelse(TEAM_PITCHING_H
> 2200, 2200,

TEAM_PITCHING_H)));
#--7
MoneyBall_Test$IMP_TEAM_PITCHING_HR <- with(MoneyBall_Test,
ifelse(TEAM_PITCHING_HR < 8, 8,

ifelse(TEAM_PITCHING_HR > 260, 260,

TEAM_PITCHING_HR)));
#--8
MoneyBall_Test$IMP_TEAM_PITCHING_BB <- with(MoneyBall_Test,
ifelse(TEAM_PITCHING_BB < 119, 119,

ifelse(TEAM_PITCHING_BB > 1000, 1000,

TEAM_PITCHING_BB)));
#--9
MoneyBall_Test$IMP_TEAM_PITCHING_SO <- with(MoneyBall_Test,
ifelse(IMP_TEAM_PITCHING_SO < 241, 241,

ifelse(IMP_TEAM_PITCHING_SO > 1700, 1700,

IMP_TEAM_PITCHING_SO)));
#--10
MoneyBall_Test$IMP_TEAM_FIELDING_E <- with(MoneyBall_Test,
ifelse(TEAM_FIELDING_E < 65, 65,
                                                  ifelse(TEAM_FIELDING_E
> 500, 500,
                                                    TEAM_FIELDING_E)));
#--11
MoneyBall_Test$IMP_TEAM_FIELDING_DP <- with(MoneyBall_Test,
ifelse(IMP_TEAM_FIELDING_DP < 71, 71,

ifelse(IMP_TEAM_FIELDING_DP > 220, 220,

IMP_TEAM_FIELDING_DP)));
```

```
#-------------------------------------------------------------------------
## 3- MODEL Deployment
#-------------------------------------------------------------------------

View(t(basicStats(MoneyBall_Test)))

P_TARGET_WINS <- with(MoneyBall_Test,  - 3.960326
                      + 0.060744 * IMP_TEAM_BATTING_H
                      + 0.031502 * IMP_TEAM_BATTING_BB
                      + 0.075213 * IMP_TEAM_BASERUN_SB
                      + 30.282484 * M_TEAM_BASERUN_SB
                      - 0.034563 * IMP_TEAM_BASERUN_CS
                      + 6.428316 * M_TEAM_BASERUN_CS
                      - 0.009303 * IMP_TEAM_PITCHING_H
                      - 0.001774 * IMP_TEAM_PITCHING_BB
                      - 0.073967 * IMP_TEAM_FIELDING_E)

FINAL_Submission <- with(MoneyBall_Test, cbind.data.frame(INDEX,
round(P_TARGET_WINS)))

colnames(FINAL_Submission) <- c("INDEX", "P_TARGET_WINS")
View(FINAL_Submission)
View(t(basicStats(FINAL_Submission[-1])))

write.csv(FINAL_Submission, "Singh_Gurjeet_MoneyBall_Test_Score.csv")
```