621 MoneyBall

Brandon Chung, Jiaxin Zheng, Andreina Arias, and Stephanie Chiang

2025-09-27

Introduction

In this homework assignment we will explore, analyze and model a data set containing 2276 professional baseball team records from the years 1871 to 2006. Our objective is to build a multiple linear regression model on the given training data to predict the number of wins for each team in the test data.

Table 1: Variable Definitions and Theoretical Effects on Wins

Variable_Name	Definition	Theoretical_Effect
INDEX	Identification variable (do not use)	None
TARGET_WINS	Number of wins	—
TEAM_BATTING_H	Base hits (1B, 2B, 3B, HR)	Positive
TEAM_BATTING_2B	Doubles	Positive
TEAM_BATTING_3B	Triples	Positive
TEAM_BATTING_HR	Homeruns	Positive
TEAM_BATTING_BB	Walks	Positive
TEAM_BATTING_HBP	Hit by pitch	Positive
TEAM_BATTING_SO	Strikeouts	Negative
TEAM_BASERUN_SB	Stolen bases	Positive
TEAM_BASERUN_CS	Caught stealing	Negative
TEAM_FIELDING_E	Errors	Negative
TEAM_FIELDING_DP	Double plays	Positive
TEAM_PITCHING_BB	Walks allowed	Negative
TEAM_PITCHING_H	Hits allowed	Negative
TEAM_PITCHING_HR	Homeruns allowed	Negative
TEAM_PITCHING_SO	Strikeouts by pitchers	Positive

Data Exploration

Data Summary

The moneyball training data set contains 16 variables, excluding the index, and 2,276 observations. Each observational unit represents a single team's statistics for that year's performance. There are 15 predictor variables which are counts of various actions in baseball such as base hits, home runs, strikeouts, stolen bases, caught stealing, hits allows and more.

As seen below in our numerical summary, the data contains NA values in certain variables (TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_SO, and TEAM_FIELDING_DP). These NA values will be addressed in the data preparation. In addition,

TEAM_BATTING_HBP contains a large amount of NAs at a count of 2085. There is also certain variables with max values that deviate significantly from the interquartile ranges such as TEAM_PITCHING_H and TEAM_PITCHING_SO.

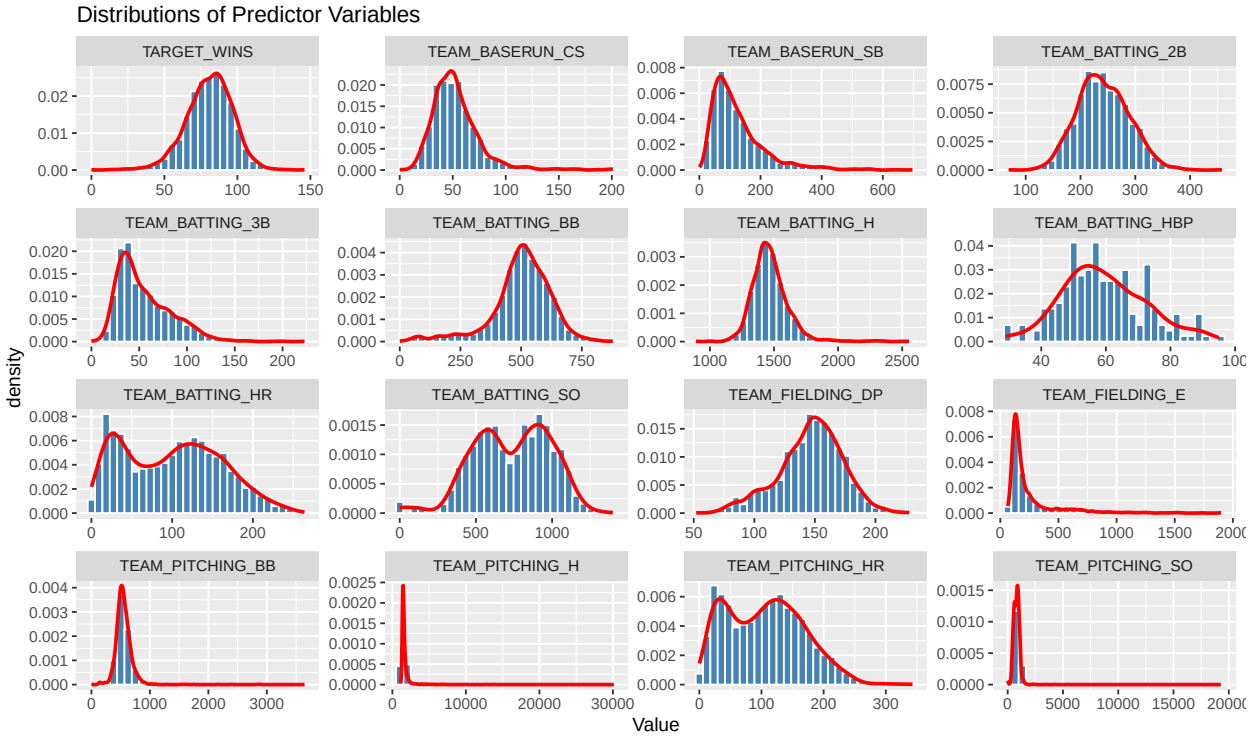
```
## Rows: 2,276
## Columns: 16
## $ TARGET_WINS      <int> 39, 70, 86, 70, 82, 75, 80, 85, 86, 76, 78, 68, 72, 7~
## $ TEAM_BATTING_H   <int> 1445, 1339, 1377, 1387, 1297, 1279, 1244, 1273, 1391,~
## $ TEAM_BATTING_2B  <int> 194, 219, 232, 209, 186, 200, 179, 171, 197, 213, 179~
## $ TEAM_BATTING_3B  <int> 39, 22, 35, 38, 27, 36, 54, 37, 40, 18, 27, 31, 41, 2~
## $ TEAM_BATTING_HR  <int> 13, 190, 137, 96, 102, 92, 122, 115, 114, 96, 82, 95,~
## $ TEAM_BATTING_BB  <int> 143, 685, 602, 451, 472, 443, 525, 456, 447, 441, 374~
## $ 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, 60, 119, 221~
## $ TEAM_BASERUN_CS  <int> NA, 28, 27, 30, 39, 59, 54, 36, 27, 34, 39, 79, 109, ~
## $ TEAM_BATTING_HBP <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ TEAM_PITCHING_H  <int> 9364, 1347, 1377, 1396, 1297, 1279, 1244, 1281, 1391,~
## $ TEAM_PITCHING_HR <int> 84, 191, 137, 97, 102, 92, 122, 116, 114, 96, 86, 95,~
## $ TEAM_PITCHING_BB <int> 927, 689, 602, 454, 472, 443, 525, 459, 447, 441, 391~
## $ 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, 11~
## $ TEAM_FIELDING_DP <int> NA, 155, 153, 156, 168, 149, 186, 136, 169, 159, 141,~
```

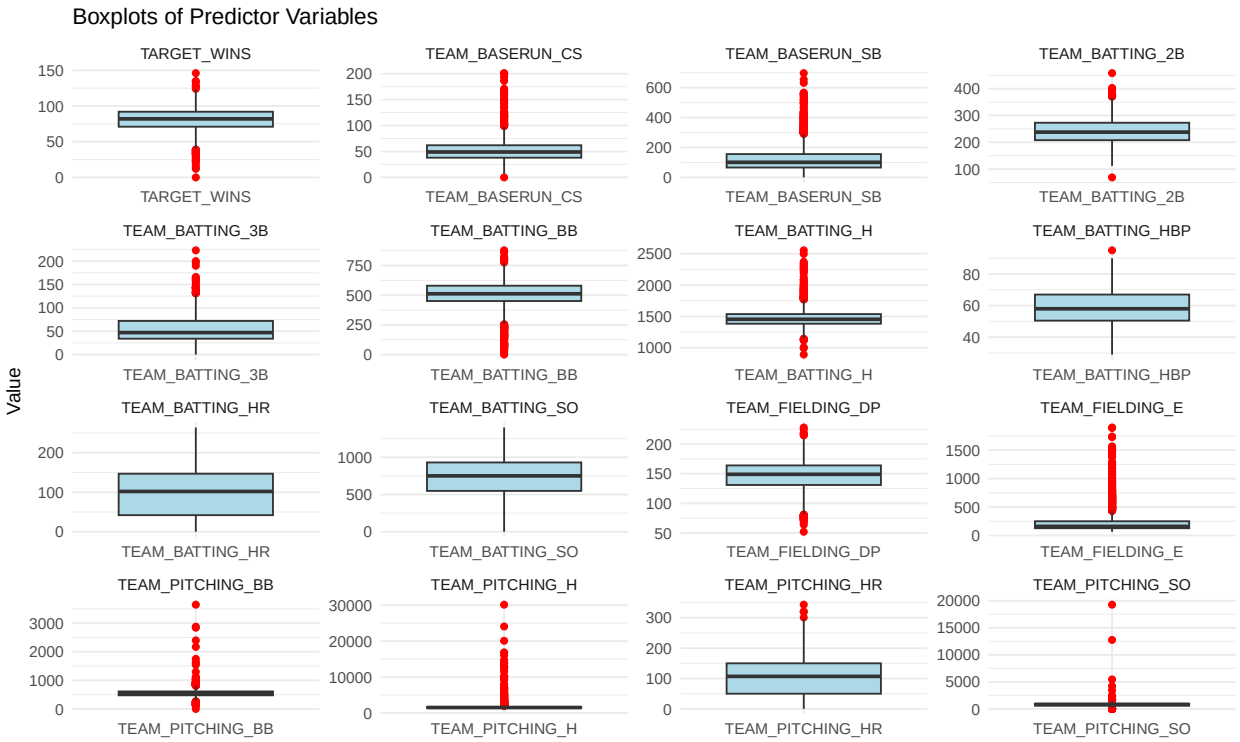
```
##      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##                0                0                0                0
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##                0                0              102              131
##  TEAM_BASERUN_CS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

```
##   TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
##  Min.   :  0.00   Min.   : 891   Min.   : 69.0   Min.   :  0.00
##  1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0   1st Qu.: 34.00
##  Median : 82.00   Median :1454   Median :238.0   Median : 47.00
##  Mean   : 80.79   Mean   :1469   Mean   :241.2   Mean   : 55.25
##  3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0   3rd Qu.: 72.00
##  Max.   :146.00   Max.   :2554   Max.   :458.0   Max.   :223.00
##
##  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB
##  Min.   :  0.00   Min.   :  0.0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0   1st Qu.: 66.0
##  Median :102.00   Median :512.0   Median : 750.0   Median :101.0
##  Mean   : 99.61   Mean   :501.6   Mean   : 735.6   Mean   :124.8
##  3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0   3rd Qu.:156.0
##  Max.   :264.00   Max.   :878.0   Max.   :1399.0   Max.   :697.0
##                                   NA's   :102      NA's   :131
##  TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
##  Min.   :  0.0   Min.   :29.00    Min.   : 1137   Min.   :  0.0
##  1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419   1st Qu.: 50.0
##  Median : 49.0   Median :58.00    Median : 1518   Median :107.0
##  Mean   : 52.8   Mean   :59.36    Mean   : 1779   Mean   :105.7
```

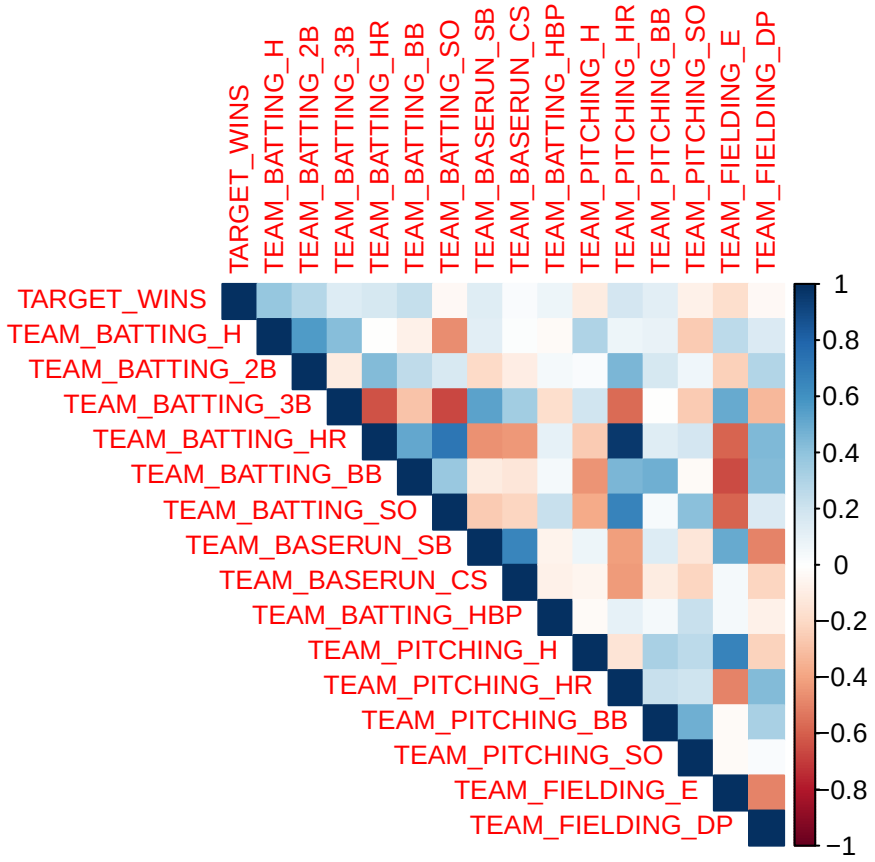
```
##  3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682   3rd Qu.:150.0
##  Max.   :201.0   Max.   :95.00    Max.   :30132   Max.   :343.0
##  NA's   :772     NA's   :2085
##  TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E  TEAM_FIELDING_DP
##  Min.   :   0.0   Min.   :    0.0   Min.   :  65.0   Min.   : 52.0
##  1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0   1st Qu.:131.0
##  Median : 536.5   Median :  813.5   Median : 159.0   Median :149.0
##  Mean   : 553.0   Mean   :  817.7   Mean   : 246.5   Mean   :146.4
##  3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2   3rd Qu.:164.0
##  Max.   :3645.0   Max.   :19278.0   Max.   :1898.0   Max.   :228.0
##                   NA's   :102                        NA's   :286
```

Data Visualizations





The histogram and box plots above provide a better understanding of the distribution of our predictor variables. Most variables have a relatively normal distribution where others show strong left and right side skewing. The box plots also clue us into possible data entry errors as may be the case for TEAM_PITCHING_SO.

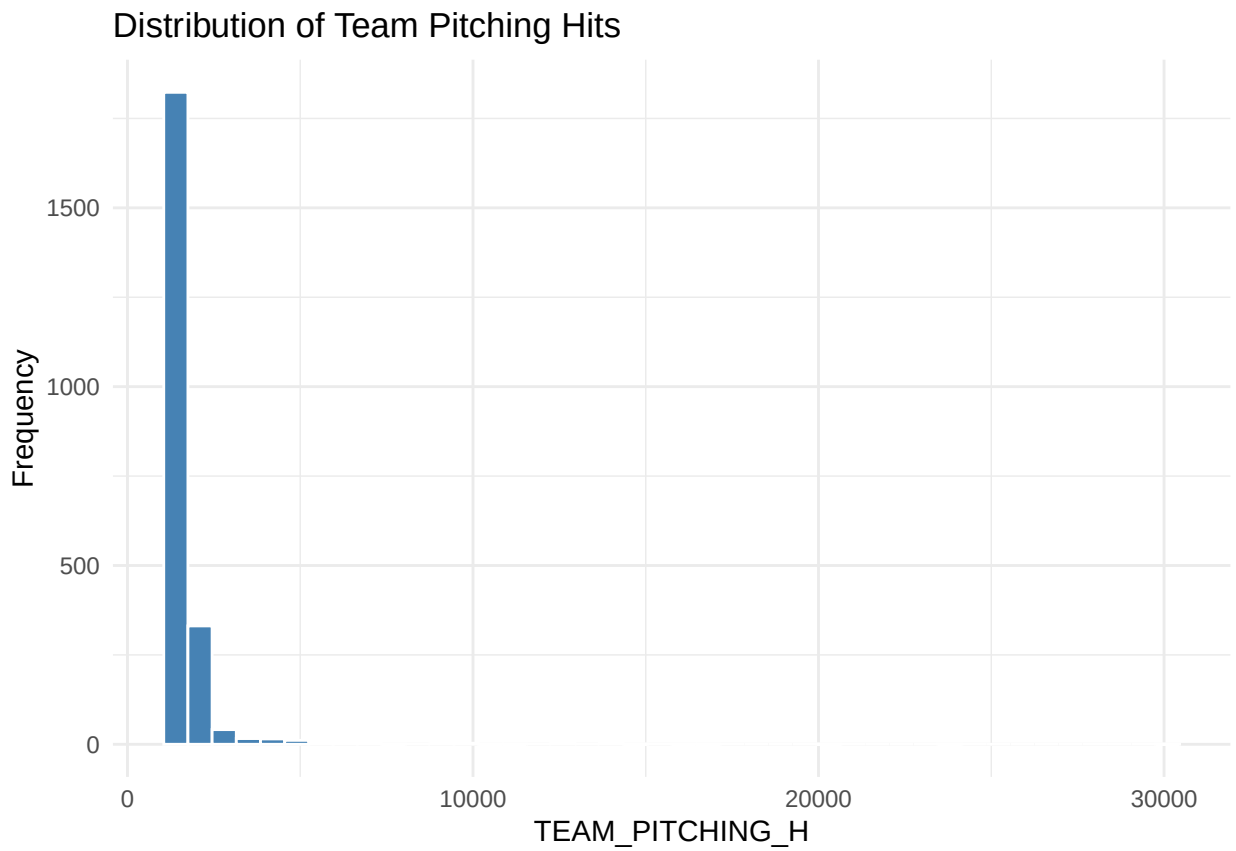


The correlation heat map helps us to see the relationship of variables against the target variable and other predictors. Correlations are mostly what was expected based on the theoretical effect given in the introduction with some exceptions. An example of this can be seen with TEAM_BASERUN_CS where the correlation is slightly positive (0.02240407) when the theoretical effect is to have a negative impact on wins.

Diving deeper into the outliers for the TEAM_PITCHING_SO (pitchers striking out the opposing team's hitter) variable we can see that the record for these teams also are paired with a 0 TEAM_PITCHING_HR (home runs allowed by the pitchers), and so it stand to reason that these outliers are not data errors.

```
##   TARGET_WINS TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## 1          41            992             263              20               0
## 2         108           1188             338               0               0
##   TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_BASERUN_CS
## 1             142             952              NA              NA
## 2             270             945              NA              NA
##   TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
## 1               NA           20088                0             2876
## 2               NA           16038                0             3645
##   TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP
## 1            19278             952              NA
## 2            12758             716              NA
```

For the outliers in TEAM_PITCHING_H (hits allowed by pitchers) our distribution shows us that the outliers are likely not data errors either. There are infrequent but other recorded values between our outliers and the IQR of our variable. Our outliers in this variable are plausible real recorded values that happen to fall far on our distribution's right sided tail.



Data Preparation

The variable TEAM_BATTING_HBP which represents a batter being hit by a pitch was removed as the influence is a factor outside of the batter's controls and it's not a repeatable skill. The variable also contained 2,085 NA values out of the total of 2,276 observations.

```
## 'data.frame':    2276 obs. of  15 variables:
##  $ 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_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 ...
```

Near zero variance variables are variables with observed values that barely change across observations.

Because of this they contribute little to analysis and introduce unnecessary complexity along with multi-collinearity risk. No variables were found to be near zero variance as seen below.

```
##                  freqRatio percentUnique zeroVar   nzv
## TARGET_WINS       1.014493      4.745167   FALSE FALSE
## TEAM_BATTING_H    1.333333     25.000000   FALSE FALSE
## TEAM_BATTING_2B   1.000000     10.544815   FALSE FALSE
## TEAM_BATTING_3B   1.074074      6.326889   FALSE FALSE
## TEAM_BATTING_HR   1.038462     10.676626   FALSE FALSE
## TEAM_BATTING_BB   1.058824     23.418278   FALSE FALSE
## TEAM_BATTING_SO   2.000000     36.115993   FALSE FALSE
## TEAM_BASERUN_SB   1.000000     15.289982   FALSE FALSE
## TEAM_BASERUN_CS   1.125000      5.623902   FALSE FALSE
## TEAM_BATTING_HBP  1.000000      2.416520   FALSE FALSE
## TEAM_PITCHING_H   1.083333     37.038664   FALSE FALSE
## TEAM_PITCHING_HR  1.038462     11.247803   FALSE FALSE
## TEAM_PITCHING_BB  1.066667     23.506151   FALSE FALSE
## TEAM_PITCHING_SO  2.222222     36.159930   FALSE FALSE
## TEAM_FIELDING_E   1.037037     24.121265   FALSE FALSE
## TEAM_FIELDING_DP  1.000000      6.326889   FALSE FALSE
```

For data imputation we looked at the columns with missing values and used imputation on those columns that have a rate 5% missing data.

```
##      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##         0.000000         0.000000         0.000000         0.000000
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##         0.000000         0.000000         4.481547         5.755712
##  TEAM_BASERUN_CS  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##        33.919156         0.000000         0.000000         0.000000
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##         4.481547         0.000000        12.565905
```

Used multiple imputation to impute the missing data using MICE predictive mean matching method.

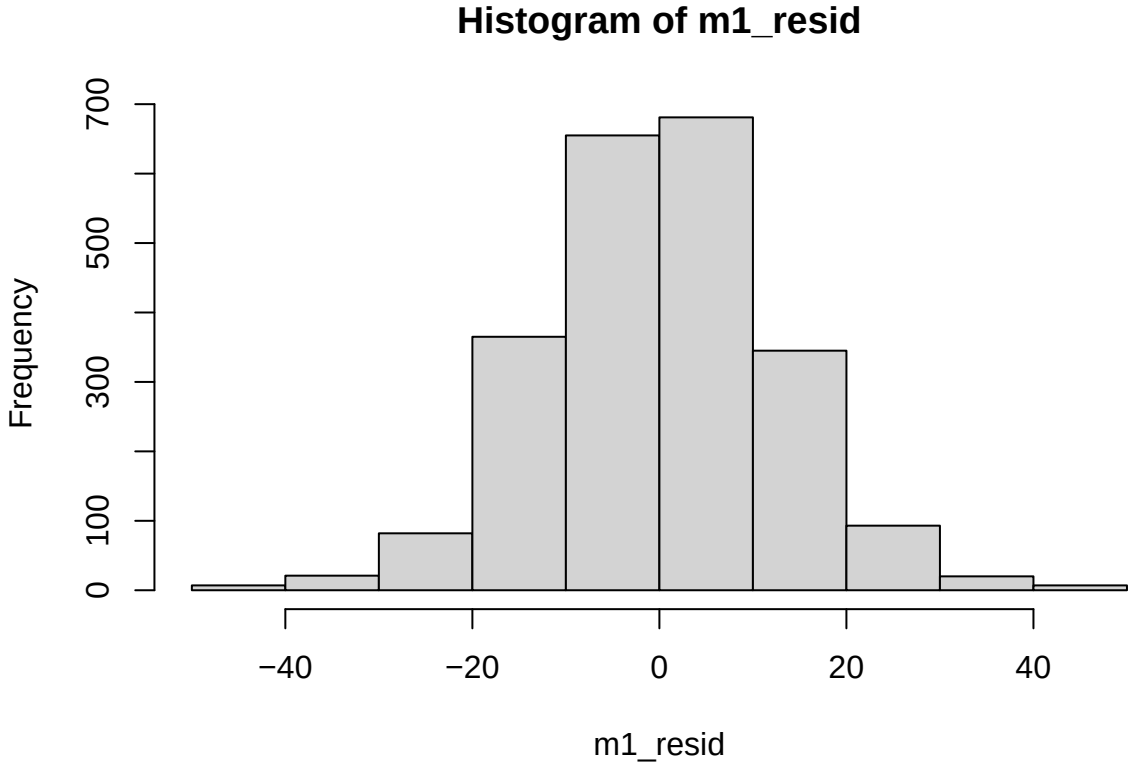
Multiple Linear Regression Models

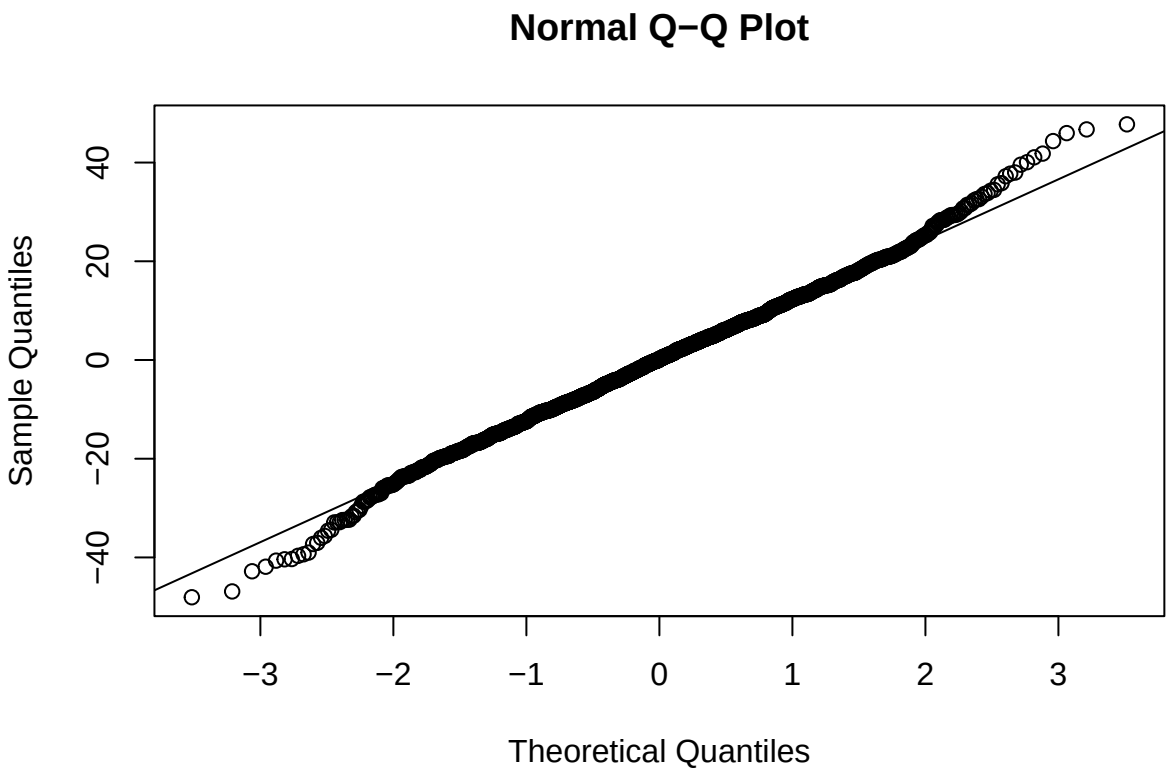
Model 1: All Features

For the first model we choose to include all the predictive variables. This will allow us to see which features have significant influence on our TARGET_WINS dependent variable.

```
##
## Call:
## lm(formula = TARGET_WINS ~ 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,
##     data = Training_imp)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -48.066  -8.413   0.173   8.114  47.738
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.6652346  5.1731357   6.508 9.37e-11 ***
## TEAM_BATTING_H    0.0431257  0.0035895  12.014  < 2e-16 ***
## TEAM_BATTING_2B  -0.0199054  0.0088954  -2.238 0.025337 *
## TEAM_BATTING_3B   0.0412403  0.0164442   2.508 0.012215 *
## TEAM_BATTING_HR   0.0576471  0.0265424   2.172 0.029968 *
## TEAM_BATTING_BB   0.0130473  0.0056243   2.320 0.020440 *
## TEAM_BATTING_SO  -0.0150600  0.0024780  -6.077 1.43e-09 ***
## TEAM_BASERUN_SB   0.0494468  0.0054066   9.146  < 2e-16 ***
## TEAM_BASERUN_CS   0.0020950  0.0110596   0.189 0.849777
## TEAM_PITCHING_H   0.0013758  0.0003859   3.566 0.000371 ***
## TEAM_PITCHING_HR  0.0236405  0.0235842   1.002 0.316263
## TEAM_PITCHING_BB -0.0036554  0.0040041  -0.913 0.361385
## TEAM_PITCHING_SO  0.0015600  0.0008943   1.744 0.081220 .
## TEAM_FIELDING_E  -0.0415048  0.0027079 -15.327  < 2e-16 ***
## TEAM_FIELDING_DP -0.1119556  0.0124114  -9.020  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.66 on 2261 degrees of freedom
## Multiple R-squared:  0.358,  Adjusted R-squared:  0.354
## F-statistic: 90.06 on 14 and 2261 DF,  p-value: < 2.2e-16
```



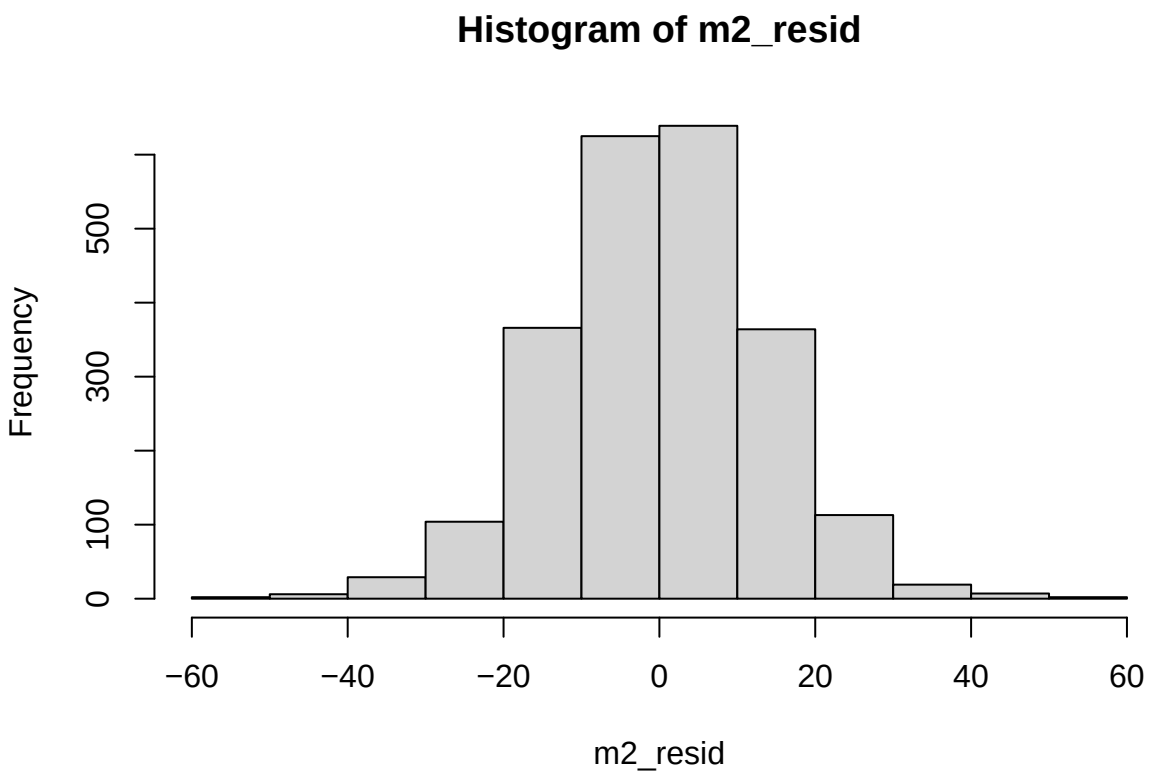


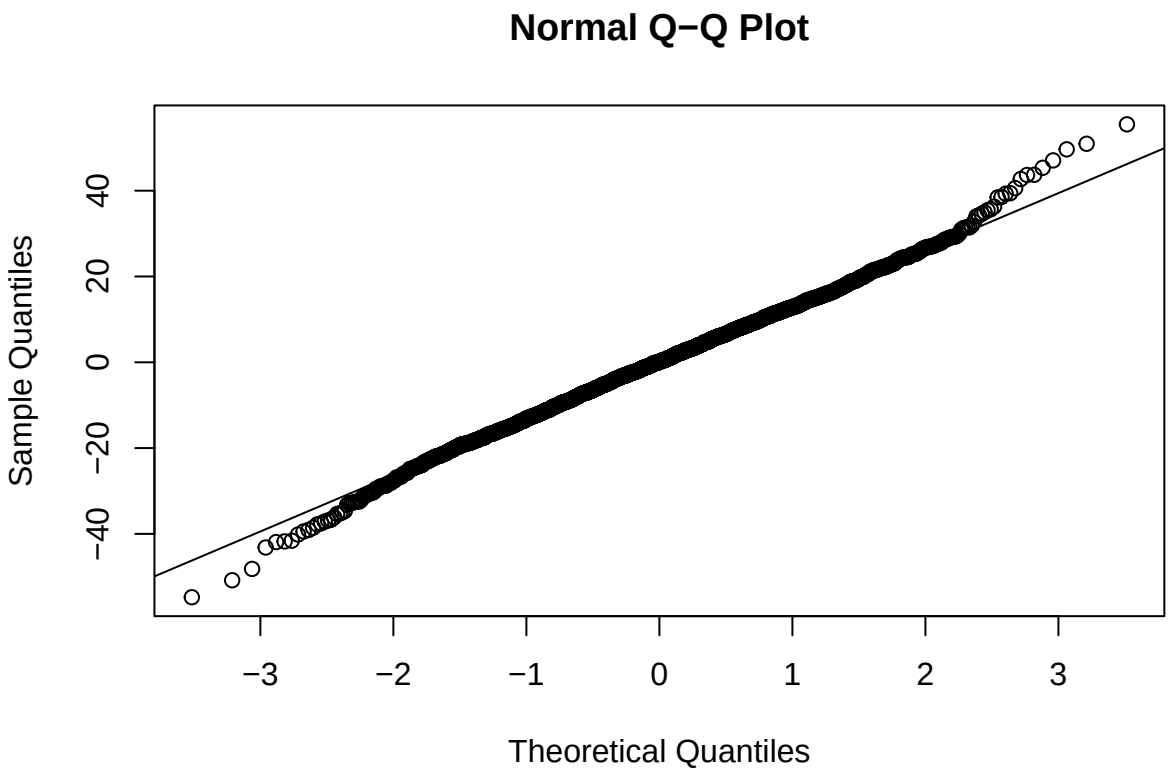
Model 2:

For the second model we narrowed down the variable selection based on our findings that TEAM_PITCHING_HR has high multicollinearity with TEAM_BATTING_HR, therefore we removed TEAM_PITCHING_HR. In addition, we removed TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_PITCHING_SO, TEAM_FIELDING_DP for missing values. Our thoughts here is that by removing these variables our model is more reliable due to removal of imputed values and reduced model complexity.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_PITCHING_H +
##     TEAM_PITCHING_BB + TEAM_FIELDING_E, data = Training_imp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.776  -8.875   0.097   8.860  55.466
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.290e+00  3.443e+00   2.117 0.034376 *
## TEAM_BATTING_H    4.848e-02  3.207e-03  15.118  < 2e-16 ***
## TEAM_BATTING_2B  -2.582e-02  9.057e-03  -2.851 0.004400 **
## TEAM_BATTING_3B   1.011e-01  1.665e-02   6.072 1.48e-09 ***
```

```
## TEAM_BATTING_HR   3.672e-02  7.749e-03   4.739 2.28e-06 ***
## TEAM_BATTING_BB  -7.926e-05  4.585e-03  -0.017 0.986208
## TEAM_PITCHING_H  -1.312e-03  3.683e-04  -3.561 0.000377 ***
## TEAM_PITCHING_BB  1.036e-02  2.802e-03   3.695 0.000225 ***
## TEAM_FIELDING_E  -1.664e-02  2.368e-03  -7.025 2.81e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.48 on 2267 degrees of freedom
## Multiple R-squared:   0.27,  Adjusted R-squared:  0.2675
## F-statistic: 104.8 on 8 and 2267 DF,  p-value: < 2.2e-16
```



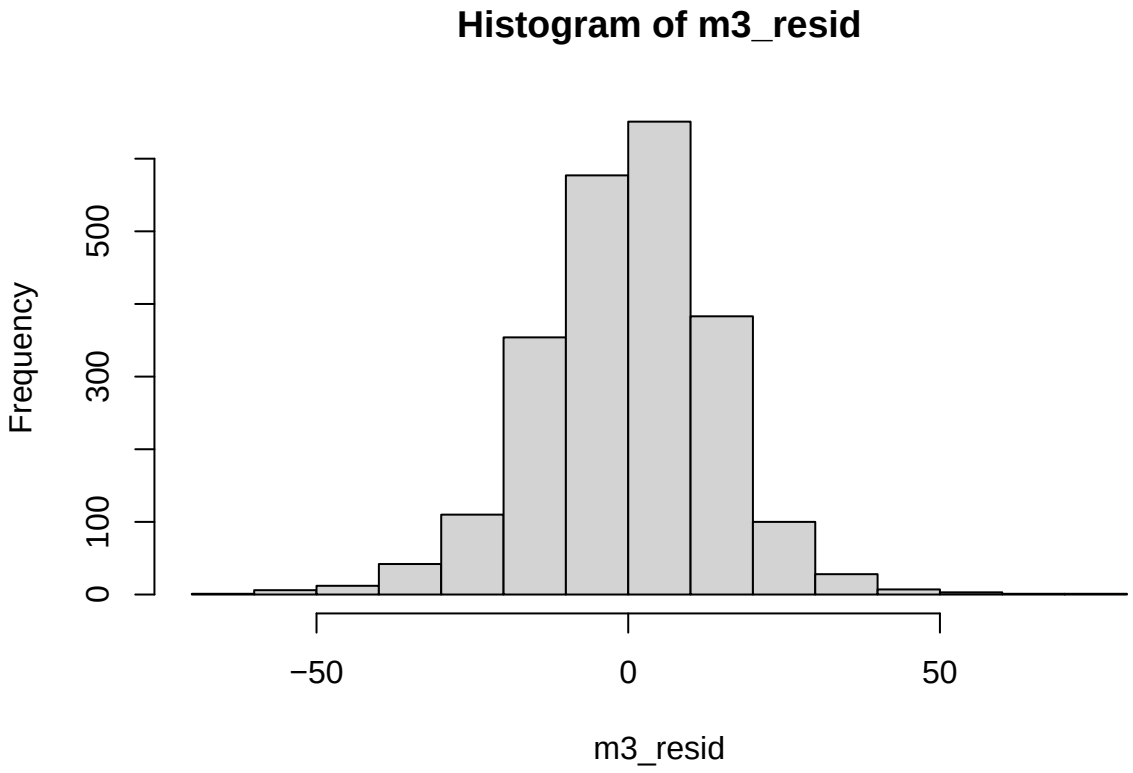


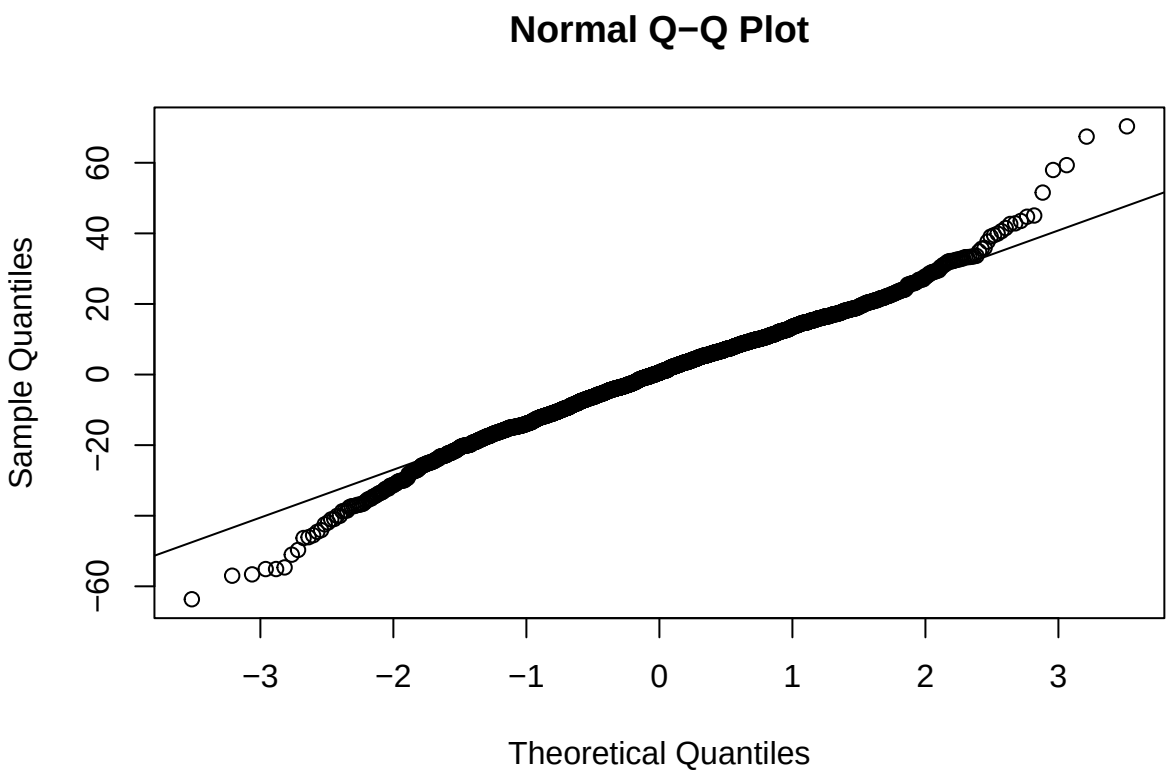
Model 3:

For our third model our group utilized the backward selection process where we removed the lowest p-value variables noted from model 1 and 2. Included in this model were only variables with p-values greater than 0.05.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_SO + TEAM_BASERUN_CS +
##     TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_BATTING_BB, data = Training_imp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -63.659  -8.994   0.549   9.297  70.322
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      63.658983   1.850740  34.397  < 2e-16 ***
## TEAM_BATTING_SO  -0.021016   0.001696 -12.388  < 2e-16 ***
## TEAM_BASERUN_CS   0.083583   0.007696  10.860  < 2e-16 ***
## TEAM_PITCHING_HR  0.116163   0.007706  15.075  < 2e-16 ***
## TEAM_PITCHING_BB -0.009051   0.002172  -4.166 3.21e-05 ***
## TEAM_BATTING_BB   0.037613   0.003223  11.669  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 14.4 on 2270 degrees of freedom
## Multiple R-squared:  0.1657, Adjusted R-squared:  0.1638
## F-statistic: 90.14 on 5 and 2270 DF,  p-value: < 2.2e-16
```





Select Models:

While Model 1 has higher multicollinearity in certain predictors, our analysis identified Model 1 as the strongest regression model. It achieved the lowest residual error (12.66) and the highest adjusted R^2 (0.354), making it the most accurate and reliable predictor of team wins.

Model 1 shows that for a baseball team to increase their amount of wins for the season they should focus on increasing their batting home runs and stolen bases. TEAM_BATTING_HR has the greatest positive impact at a coefficient of 0.05764 and TEAM_BASERUN_SB has the second greatest positive impact with a coefficient of 0.04945. Conversely, minimizing fielding errors (TEAM_FIELDING_E) as this variable has the largest negative impact on wins with a coefficient of -0.041504.

The variable TEAM_BATTING_HR is noted to be highly correlated with TEAM_PITCHING_HR, however both of these variables have large theoretical impact to the probability of winning. Hitting a home run or allowing a home run directly influences the game's score and therefore our group decided to keep these variables.

```
## # A tibble: 3 x 4
##   Model     RSE Adj.R2 F.Statistic
##   <chr>   <dbl>  <dbl>       <dbl>
## 1 Model 1  12.7  0.354        90.1
## 2 Model 2  13.5  0.267       105.
## 3 Model 3  14.4  0.164        90.1
```

Model 1 variables VIF

```
##   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR
##         3.823342         2.460052         2.995896        36.657149
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_BASERUN_CS
##         6.756380         5.274069         4.349937         4.373084
##  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
##         4.182680        29.664612         6.297724         3.336076
##  TEAM_FIELDING_E TEAM_FIELDING_DP
##         5.399699         1.872039
```

Model 2 variables VIF

```
##   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR
##         2.691190         2.248967         2.707698         2.755238
##  TEAM_BATTING_BB  TEAM_PITCHING_H TEAM_PITCHING_BB  TEAM_FIELDING_E
##         3.958646         3.361075         2.720094         3.642208
```

Model 3 variables VIF

```
##  TEAM_BATTING_SO  TEAM_BASERUN_CS TEAM_PITCHING_HR TEAM_PITCHING_BB
##         1.909613         1.635908         2.446552         1.432176
##  TEAM_BATTING_BB
##         1.714261
```

Utilizing our model 1 below we can see our predicted TARGET_WINS for the evaluation data.

```
##   [1]  61.70438  64.43788  74.03427  87.39829  58.94786  77.30199  86.13339
##   [8]  76.26872  69.82539  73.39817  68.68975  82.94084  82.04394  83.30519
##  [15]  86.00371  78.02754  73.63939  78.06545  71.50434  91.30627  81.36126
##  [22]  83.82291  79.61094  72.07780  82.58964  88.28316  48.71756  74.33875
##  [29]  82.72964  74.07607  90.01052  85.66996  81.48934  82.88474  78.94106
##  [36]  86.30069  75.49494  89.97919  86.62608  91.18688  82.82761  90.68766
##  [43]  26.96493 109.79863  97.22876  98.13209 100.82611  76.25749  68.20711
##  [50]  79.56018  76.91483  85.61544  75.67395  73.50105  74.54285  78.78853
##  [57]  92.67873  76.20721  64.58450  81.16847  88.29978  73.38585  88.15314
##  [64]  86.27224  85.34943 108.55313  73.01577  79.03907  78.59596  88.13572
##  [71]  84.77313  70.74176  77.95723  90.39901  80.00471  83.91870  82.31571
##  [78]  83.67792  72.69503  77.56226  84.84680  87.35468  96.60434  74.03809
##  [85]  84.48714  81.67617  83.82346  83.89574  89.98801  90.31530  83.03474
##  [92]  83.68749  73.71849  87.69547  86.27199  85.21599  87.84104 101.48732
##  [99]  85.53824  86.51020  78.84594  74.09628  83.65425  84.05378  78.11537
## [106]  63.05545  57.92238  76.62968  86.48213  57.39852  85.01666  86.85096
## [113]  94.61449  91.90134  81.10868  77.98767  85.54428  81.09600  73.48884
## [120]  77.50156  99.09390  69.19853  69.67346  68.15842  68.00319  88.09358
## [127]  90.02270  76.59586  92.76469  91.37175  85.09122  79.84423  79.90539
## [134]  85.03472  87.59056  71.73025  74.05494  77.55132  89.23137  81.18155
## [141]  63.94014  73.66388  90.29776  71.64263  71.34484  71.42443  76.51099
## [148]  78.86705  78.93489  82.97546  82.38224  80.33354  53.00456  68.93829
## [155]  76.46388  70.76381  89.54568  68.43599  90.84364  75.78387 102.72792
## [162] 107.37037  93.87661 103.47792  97.22779  89.54158  81.77328  82.51216
## [169]  73.62276  80.78200  89.72416  89.20888  80.09017  93.92141  82.66462
## [176]  72.87966  77.64884  70.23489  73.58130  79.10529  90.23682  88.50916
## [183]  86.02718  84.53059  84.86146  99.18920  87.99015  65.03207  64.47068
```

```
## [190] 115.06085  70.88524  84.05417  76.60068  77.27529  79.08308  67.76610
## [197]  78.06713  84.38418  79.32341  82.97360  73.77323  78.59396  72.37519
## [204]  91.71000  81.53258  83.28816  77.10427  76.87136  82.76409  72.50850
## [211] 104.82097  89.74709  81.07565  64.70332  67.65049  82.83278  78.40176
## [218]  94.62109  77.53758  78.18899  77.52839  74.00609  80.55173  72.57908
## [225]  70.83159  75.07318  81.50358  78.42806  81.18371  84.41642  81.95687
## [232]  93.48219  78.70169  89.32369  79.60386  74.66871  82.09326  77.39837
## [239]  88.68597  72.03140  88.47934  86.42141  83.39604  81.54812  60.92730
## [246]  88.06432  81.04736  85.19076  72.97059  84.39924  79.99281  62.77491
## [253]  95.70136  33.87954  69.47688  76.60465  82.90241  84.59043  76.51166
```