DATA621: Business Analytics and Data Mining

Assignment 1: Linear Regression on MoneyBall Dataset

Irene Jacob

Overview

In this homework assignment, you will explore, analyze and model a data set containing approximately 2200 records. Each record represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all the statistics adjusted to match the performance of a 162-game season.

Your objective is to build a multiple linear regression model on the training data to predict the number of wins for the team. You can only use the variables given to you (or variables that you derive from the variables provided). Below is a short description of the variables of interest in the data set:

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT
INDEX	Identification Variable (do not use)	None
TARGET_WINS	Number of 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

1. Data Exploration

First the training and evaluation datasets are loaded into train and test respectively from github. There are 2276 entries and 17 variables in the train dataset. The Index is removed while loading the dataset. There are 6 variables with missing values (NAs).

The summary statistics is as follows:

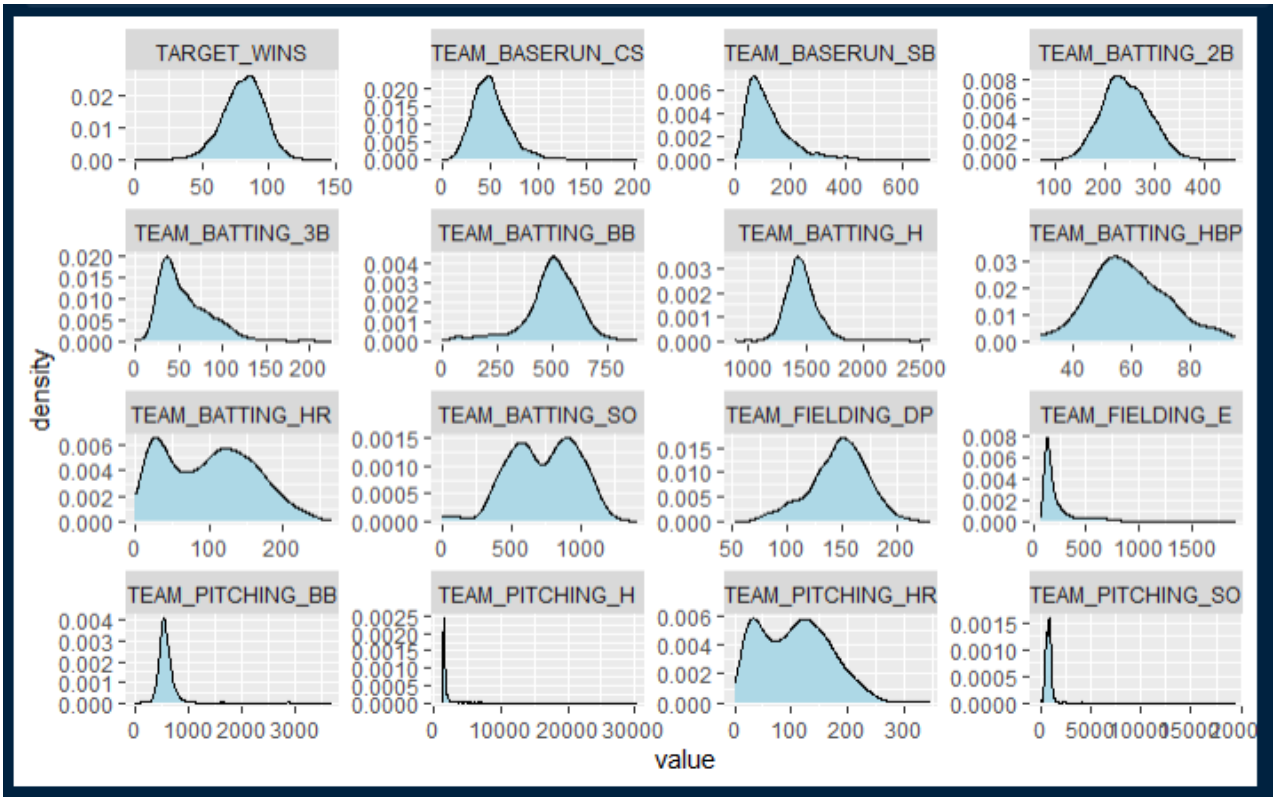
```
 TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB
Min.   :  0.00   Min.   : 891   Min.   : 69.0   Min.   :  0.00   Min.   :  0.00   Min.   :  0.0
1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0   1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0
Median : 82.00   Median :1454   Median :238.0   Median : 47.00   Median :102.00   Median :512.0
Mean   : 80.79   Mean   :1469   Mean   :241.2   Mean   : 55.25   Mean   : 99.61   Mean   :501.6
3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0   3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0
Max.   :146.00   Max.   :2554   Max.   :458.0   Max.   :223.00   Max.   :264.00   Max.   :878.0

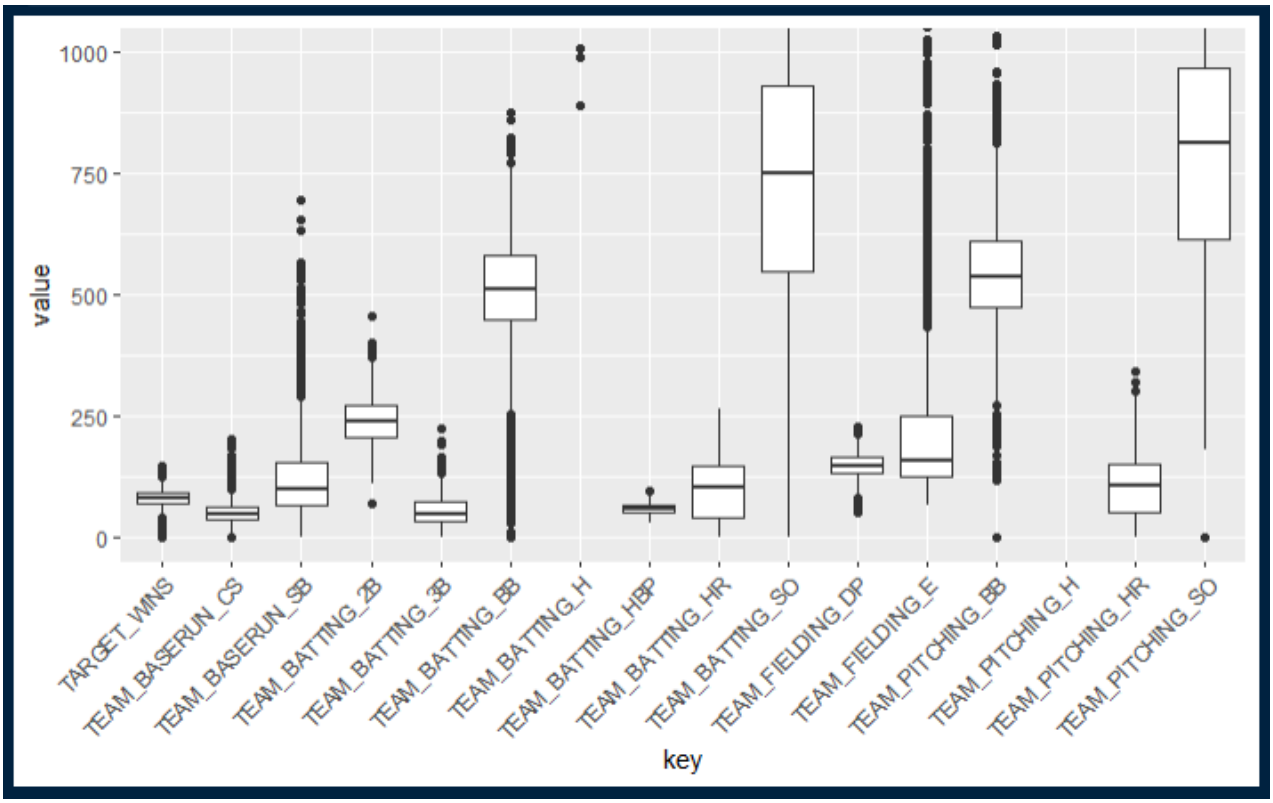
TEAM_BATTING_SO  TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
Min.   :   0.0   Min.   :  0.0   Min.   :  0.0   Min.   :29.00    Min.   : 1137   Min.   :  0.0
1st Qu.: 548.0   1st Qu.: 66.0   1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419   1st Qu.: 50.0
Median : 750.0   Median :101.0   Median : 49.0   Median :58.00    Median : 1518   Median :107.0
Mean   : 735.6   Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779   Mean   :105.7
3rd Qu.: 930.0   3rd Qu.:156.0   3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682   3rd Qu.:150.0
Max.   :1399.0   Max.   :697.0   Max.   :201.0   Max.   :95.00    Max.   :30132   Max.   :343.0
NA's   :102      NA's   :131     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
```

Finding the skewness of the dataset

While examining the skewness of the dataset some variables show almost normal distribution while some show extreme skewness suggesting outliers.

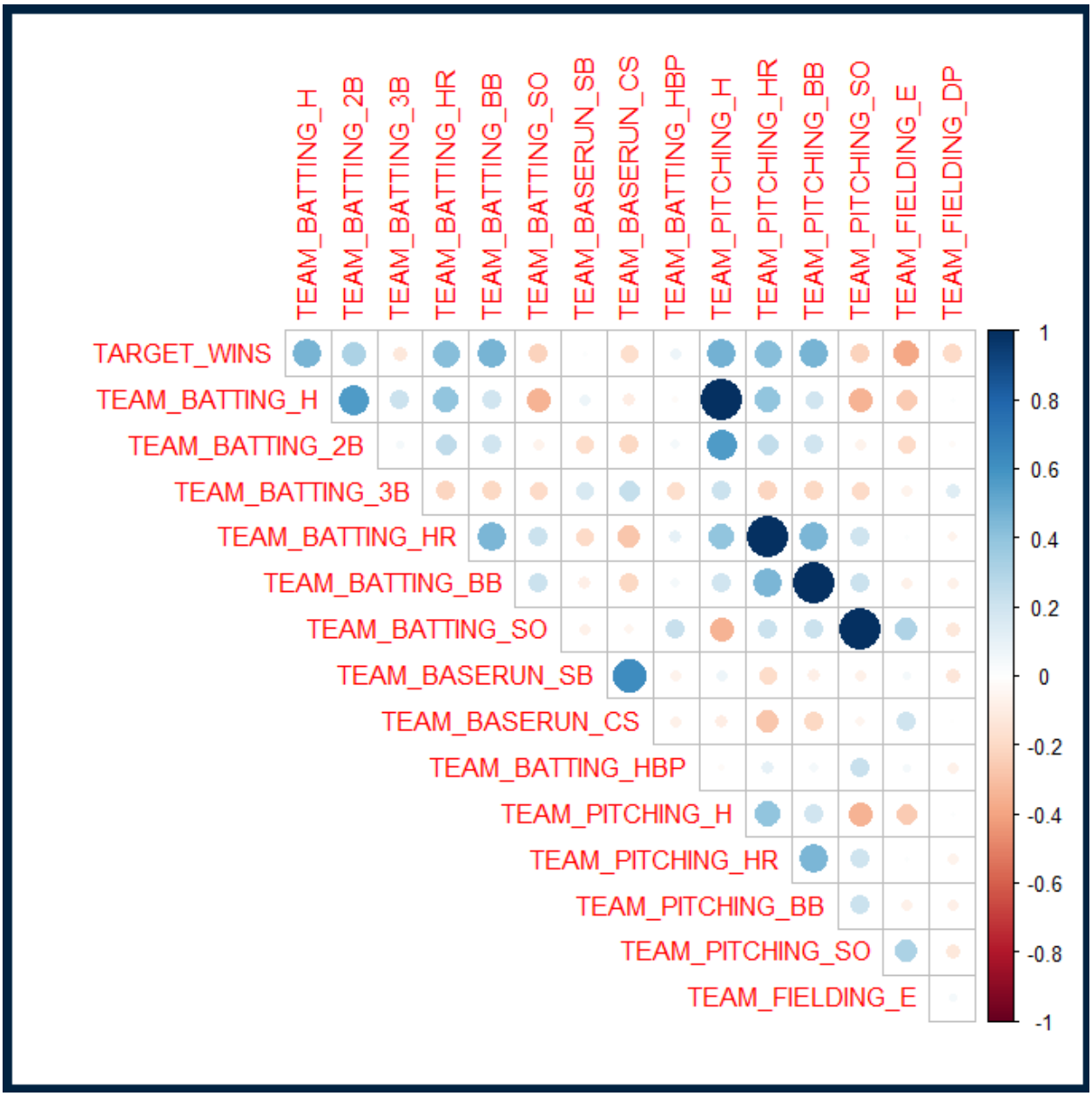
The outliers can be visualized more clearly using a boxplot.


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

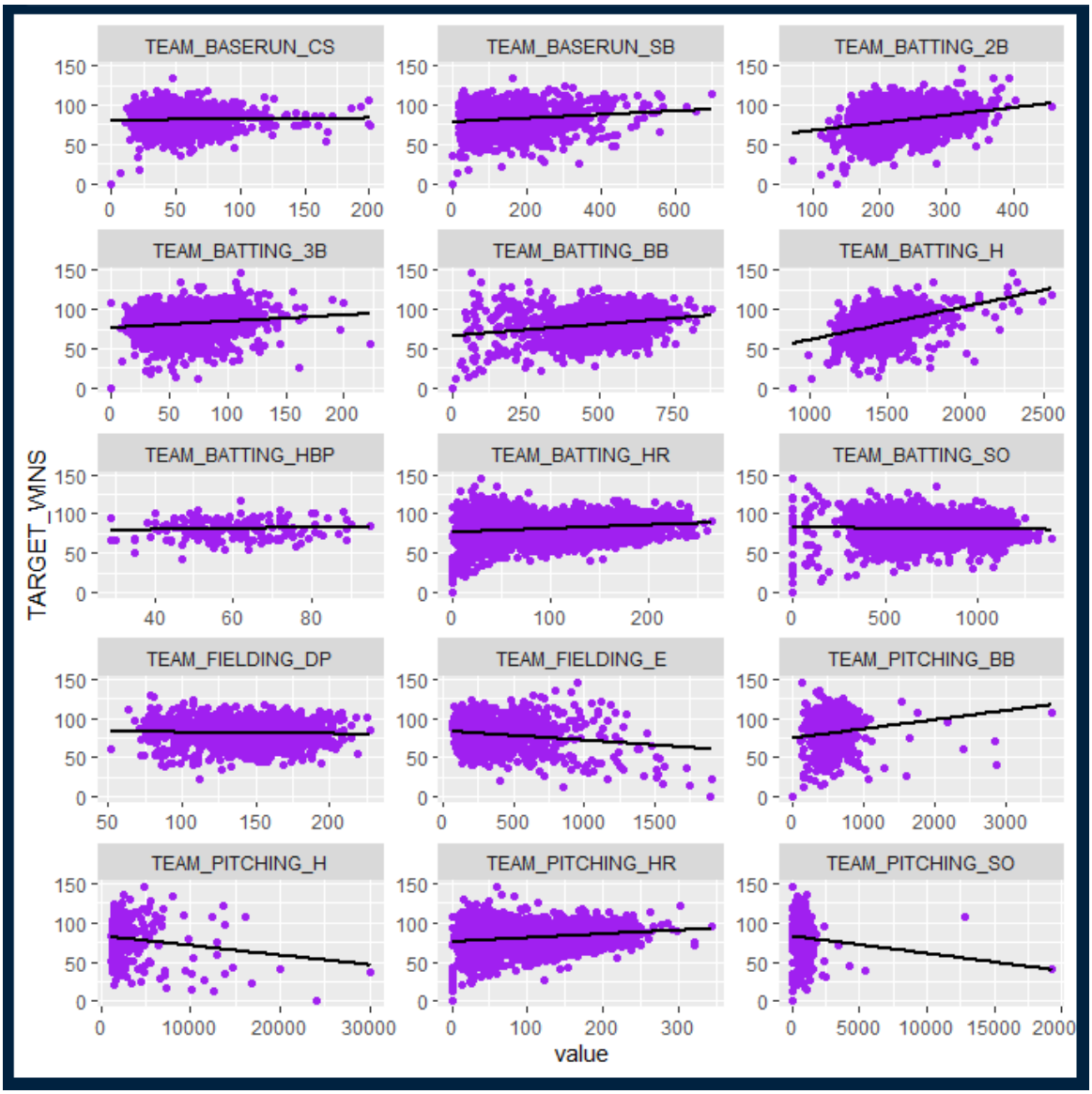

value
key
TARGET_WINS
TEAM_BASERUN_CS
TEAM_BASERUN_SB
TEAM_BATTING_2B
TEAM_BATTING_3B
TEAM_BATTING_BB
TEAM_BATTING_H
TEAM_BATTING_HBP
TEAM_BATTING_HR
TEAM_BATTING_SO
TEAM_FIELDING_DP
TEAM_FIELDING_E
TEAM_PITCHING_BB
TEAM_PITCHING_H
TEAM_PITCHING_HR
TEAM_PITCHING_SO

Correlation

The variables show positive and negative correlation between each other. The correlation plot is shown below.



Correlation with target variable:



Missing Data

TEAM_BATTING_HBP is missing 92% of the data so I will be excluding that from the model.

Variable	Count	Percentage
<chr>	<int>	<chr>
TEAM_BATTING_HBP	2085	92%
TEAM_BASERUN_CS	772	34%
TEAM_FIELDING_DP	286	13%
TEAM_BASERUN_SB	131	5.8%
TEAM_BATTING_SO	102	4.5%
TEAM_PITCHING_SO	102	4.5%

6 rows

2. Data Preparation

Handling the missing data

The data has some missing values, and certain variables that I removed and also some outliers.

I removed TEAM_BATTING_HBP as more than 92% of the values are missing. I will be handling the missing values by imputing the mean value. TEAM_PITCHING_SO, TEAM_PITCHING_BB, TEAM_PITCHING_H, TEAM_FIELDING_E have a lot of outliers so those are also imputed with the mean value.

I noticed that for the team with 0 wins has 0 in many of the fields so I am filtering out the non zero wins.

Below is the summary after the above changes were made.

```
##   TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
##  Min.   : 12.00   Min.   : 992   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   : 80.83   Mean   :1470   Mean   :241.3   Mean   : 55.27
##  3rd Qu.: 92.00   3rd Qu.:1538   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.   : 12.0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 557.5   1st Qu.: 67.0
##  Median :102.00   Median :512.0   Median : 735.6   Median :106.0
##  Mean   : 99.66   Mean   :501.8   Mean   : 735.9   Mean   :124.8
##  3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 925.0   3rd Qu.:151.0
##  Max.   :264.00   Max.   :878.0   Max.   :1399.0   Max.   :697.0
##  TEAM_BASERUN_CS  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##  Min.   :  7.00   Min.   :1137    Min.   :  0.0    Min.   : 119.0
##  1st Qu.: 44.00   1st Qu.:1419    1st Qu.: 50.0    1st Qu.: 476.0
##  Median : 52.80   Median :1518    Median :107.0    Median : 537.0
##  Mean   : 52.83   Mean   :1626    Mean   :105.7    Mean   : 548.3
##  3rd Qu.: 54.50   3rd Qu.:1664    3rd Qu.:150.0    3rd Qu.: 610.0
##  Max.   :201.00   Max.   :4969    Max.   :343.0    Max.   :1750.0
##  TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP
##  Min.   :   0.0   Min.   : 65.0   Min.   : 52.0
##  1st Qu.: 626.0   1st Qu.:127.0   1st Qu.:134.0
##  Median : 800.0   Median :159.0   Median :146.4
##  Mean   : 800.4   Mean   :175.6   Mean   :146.4
##  3rd Qu.: 956.0   3rd Qu.:191.0   3rd Qu.:161.5
##  Max.   :3450.0   Max.   :479.0   Max.   :228.0
```

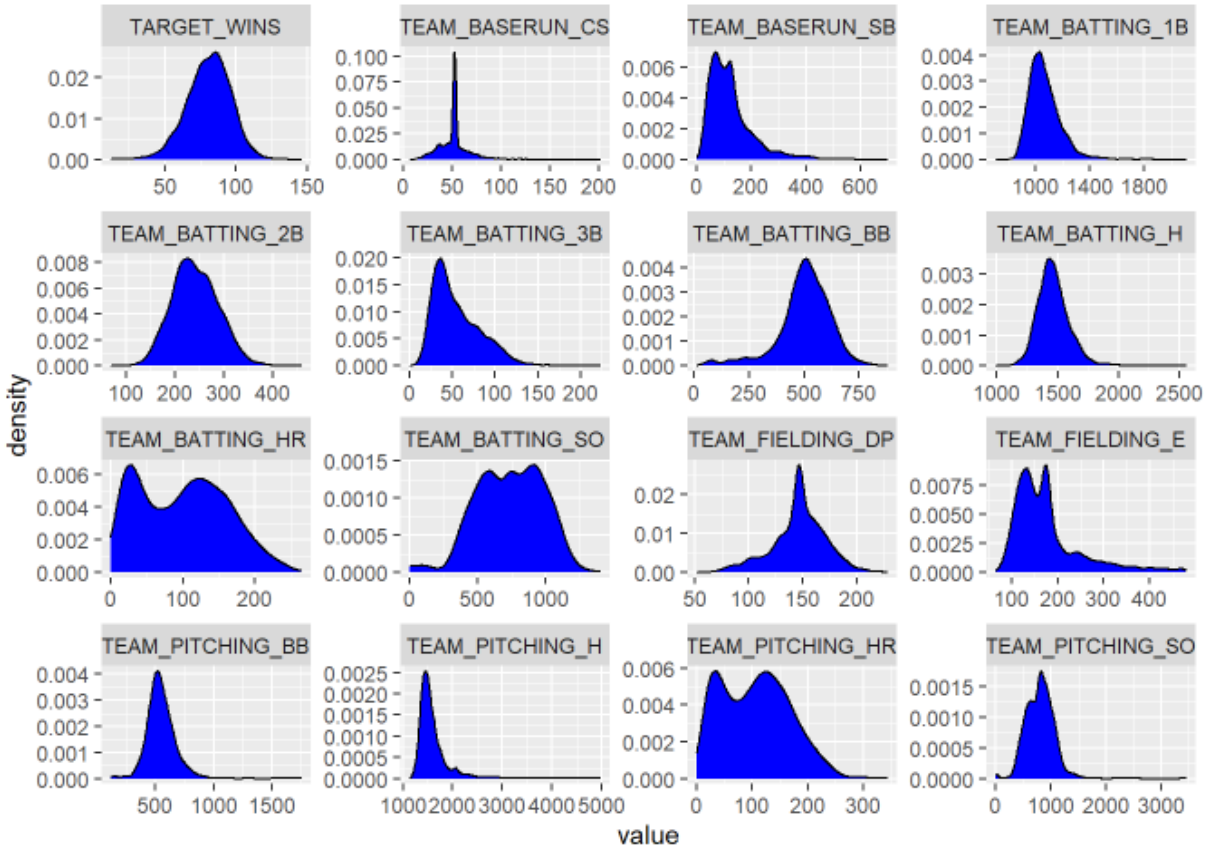
Feature Engineering

I noticed that there were entries for doubles and triples by batters, but singles were not recorded. To get the singles value I used the below equation

TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR

I added the singles value to the train and test set.

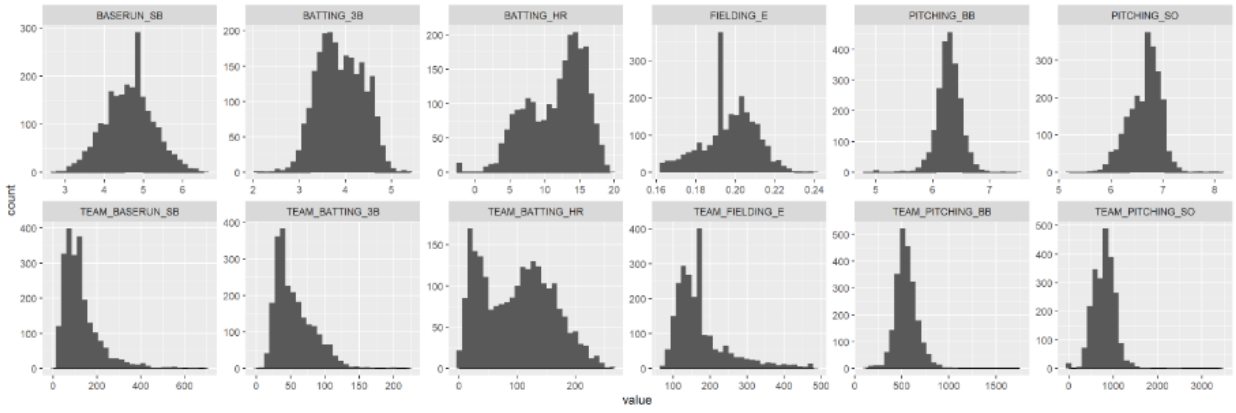
Below is the density plot and the summary of the data so far:



```
##   TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
##  Min.   : 12.00   Min.   : 992   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   : 80.83   Mean   :1470   Mean   :241.3   Mean   : 55.27
##  3rd Qu.: 92.00   3rd Qu.:1538   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.   : 12.0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 557.5   1st Qu.: 67.0
##  Median :102.00   Median :512.0   Median : 735.6   Median :106.0
##  Mean   : 99.66   Mean   :501.8   Mean   : 735.9   Mean   :124.8
##  3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 925.0   3rd Qu.:151.0
##  Max.   :264.00   Max.   :878.0   Max.   :1399.0   Max.   :697.0
##  TEAM_BASERUN_CS  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
##  Min.   :  7.00   Min.   :1137    Min.   :  0.0    Min.   : 119.0
##  1st Qu.: 44.00   1st Qu.:1419    1st Qu.: 50.0    1st Qu.: 476.0
##  Median : 52.80   Median :1518    Median :107.0    Median : 537.0
##  Mean   : 52.83   Mean   :1626    Mean   :105.7    Mean   : 548.3
##  3rd Qu.: 54.50   3rd Qu.:1664    3rd Qu.:150.0    3rd Qu.: 610.0
##  Max.   :201.00   Max.   :4969    Max.   :343.0    Max.   :1750.0
##  TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP TEAM_BATTING_1B
##  Min.   :   0.0   Min.   : 65.0   Min.   : 52.0    Min.   : 709
##  1st Qu.: 626.0   1st Qu.:127.0   1st Qu.:134.0    1st Qu.: 991
##  Median : 800.0   Median :159.0   Median :146.4    Median :1050
##  Mean   : 800.4   Mean   :175.6   Mean   :146.4    Mean   :1073
##  3rd Qu.: 956.0   3rd Qu.:191.0   3rd Qu.:161.5    3rd Qu.:1129
##  Max.   :3450.0   Max.   :479.0   Max.   :228.0    Max.   :2112
```

Transformation

I found 6 variables that are heavily skewed so I did some boxcox transformations on them and the comparison density plots are as follows. The variables on the first row are the transformed ones and they look more normal than the old ones without transformations.



I applied the mean value imputation on the NA values of the transformed fields.

3. Build models

1. Simple model using the transformed data
2. Simple model
3. Full model
4. Polynomial Regression
5. Excluding variables with Multicollinearity
6. Excluding variables having insignificant p values

1. Simple model using the transformed data

For this model I am selecting some of the variables having significant correlation.

I will be choosing the below variables:

TEAM_BATTING_HR

TEAM_PITCHING_BB

TEAM_FIELDING_E

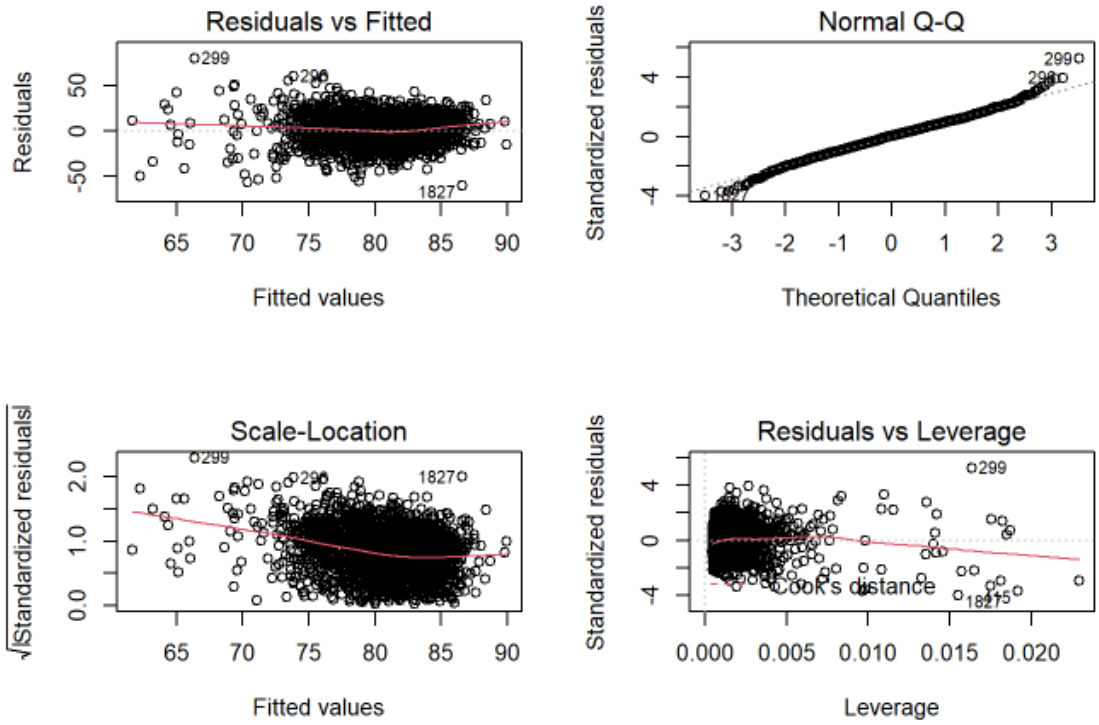
The linear regression summary for this model is as follows:

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_HR + TEAM_PITCHING_BB +
##     TEAM_FIELDING_E, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.556  -9.901   0.616  10.193  79.631
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       22.3652    11.0563   2.023   0.0432 *
## TEAM_BATTING_HR    0.5566     0.1140   4.881 1.13e-06 ***
## TEAM_PITCHING_BB   8.3934     1.4427   5.818 6.80e-09 ***
## TEAM_FIELDING_E   -3.7810    34.0849  -0.111   0.9117
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.28 on 2271 degrees of freedom
## Multiple R-squared:  0.04918,    Adjusted R-squared:  0.04793
## F-statistic: 39.16 on 3 and 2271 DF,  p-value: < 2.2e-16
```

For this it can be seen that the Adjusted R^2 is 0.04793 which is not that great as it is less than 0.4.

The p value is very less for this model.

The plots for this model are as follows:



2. Simple Model

For this model I will use the data on which transformations are not made. I am using the same variables as the previous model.

TEAM_BATTING_HR

TEAM_PITCHING_BB

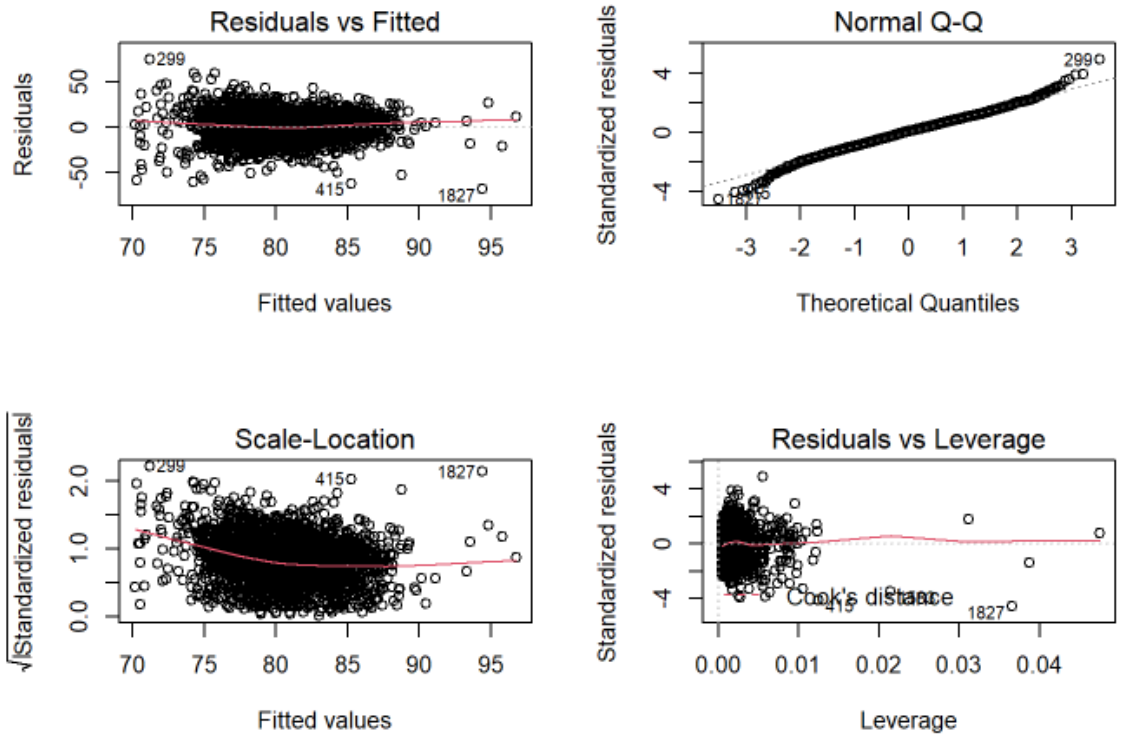
TEAM_FIELDING_E

The linear regression summary for this model is as follows:

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_HR + TEAM_PITCHING_BB +
##     TEAM_FIELDING_E, data = train_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -68.377  -9.930   0.786  10.054  74.808
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      66.984768   1.989178  33.675  < 2e-16 ***
## TEAM_BATTING_HR   0.041095   0.007094   5.793 7.88e-09 ***
## TEAM_PITCHING_BB  0.016128   0.002612   6.174 7.87e-10 ***
## TEAM_FIELDING_E   0.005141   0.005715   0.900    0.368
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.3 on 2271 degrees of freedom
## Multiple R-squared:  0.0468, Adjusted R-squared:  0.04554
## F-statistic: 37.17 on 3 and 2271 DF,  p-value: < 2.2e-16
```

For this it can be seen that the Adjusted R^2 is 0.04554 which is not that great as it is less than 0.4. The p value is very less for this model similar to the previous model.

The plots for this model are as follows:



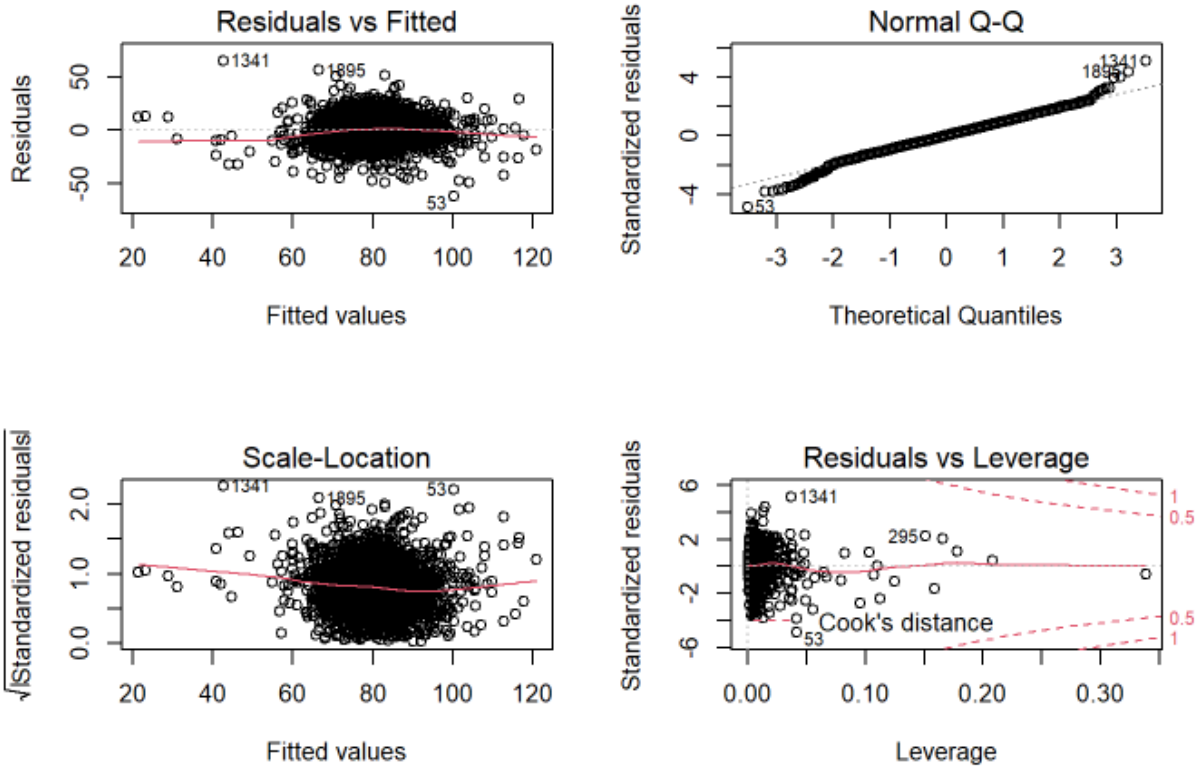
3. Full model

For this model I will use all the variables in the train dataset. The linear regression summary for this model is as follows:

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = train_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.212  -8.029   0.171   8.440  65.249
##
## Coefficients: (1 not defined because of singularities)
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      22.073680   5.583949   3.953 7.95e-05 ***
## TEAM_BATTING_H    0.043162   0.003722  11.597  < 2e-16 ***
## TEAM_BATTING_2B  -0.018652   0.009097  -2.050  0.04045 *
## TEAM_BATTING_3B   0.090657   0.016626   5.453 5.50e-08 ***
## TEAM_BATTING_HR  -0.048907   0.030706  -1.593  0.11135
## TEAM_BATTING_BB   0.066961   0.005095  13.143  < 2e-16 ***
## TEAM_BATTING_SO   0.003144   0.003956   0.795  0.42688
## TEAM_BASERUN_SB   0.023170   0.004193   5.526 3.66e-08 ***
## TEAM_BASERUN_CS   0.005847   0.015797   0.370  0.71131
## TEAM_PITCHING_H   0.003070   0.000935   3.284  0.00104 **
## TEAM_PITCHING_HR  0.086333   0.027145   3.180  0.00149 **
## TEAM_PITCHING_BB -0.034795   0.004370  -7.962 2.67e-15 ***
## TEAM_PITCHING_SO -0.005554   0.002884  -1.926  0.05422 .
## TEAM_FIELDING_E  -0.049204   0.005488  -8.966  < 2e-16 ***
## TEAM_FIELDING_DP -0.146089   0.013343 -10.949  < 2e-16 ***
## TEAM_BATTING_1B         NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.97 on 2260 degrees of freedom
## Multiple R-squared:  0.3181, Adjusted R-squared:  0.3139
## F-statistic: 75.31 on 14 and 2260 DF,  p-value: < 2.2e-16
```

For this it can be seen that the Adjusted R^2 is 0.3139 which better than the previous models but still less than 0.4. The p value is very less for this model similar to the previous model.

The plots for this model are as follows:



4. Polynomial Regression

Polynomial Regression is a regression algorithm that models the relationship between y and x as nth degree polynomial.

Here I am considering all the variables and their nth degree variables. (Assuming n as 4).

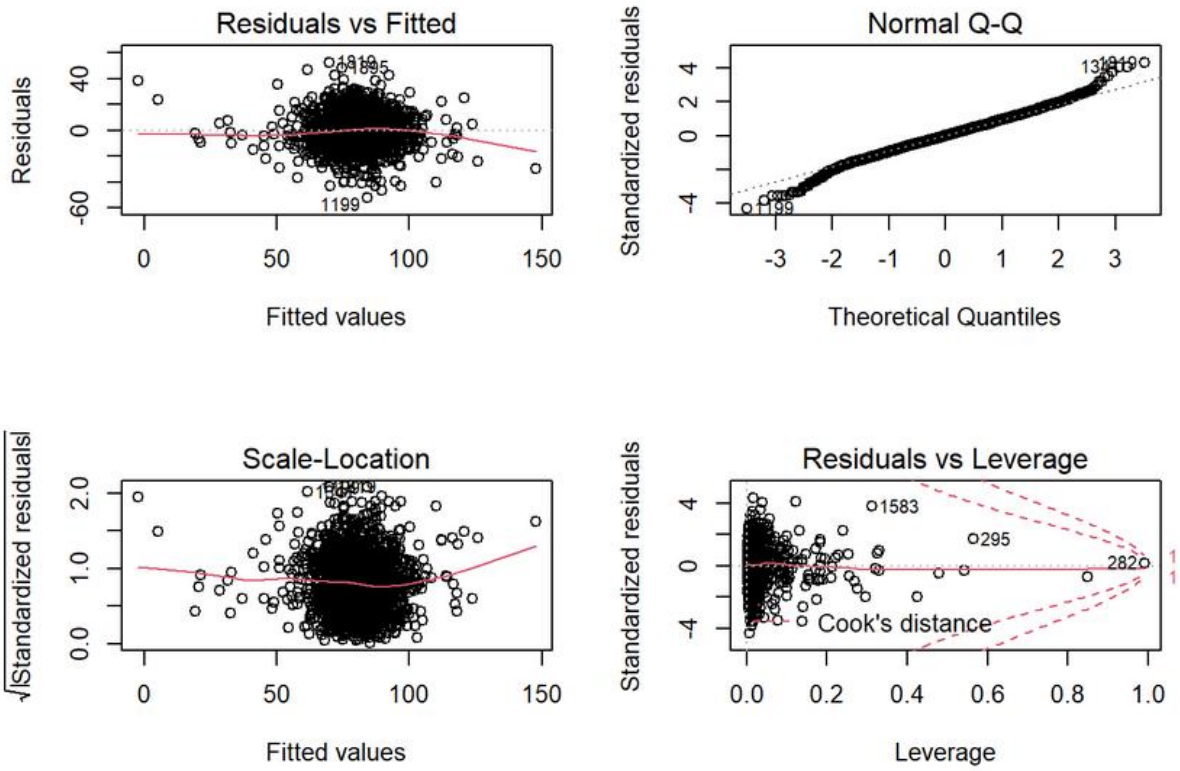
The linear regression summary is as follows:

```
##
## Call:
## lm(formula = train_poly_lm_call[2], data = train_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.291  -7.246   0.158   7.658  52.304
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -1.670e+01  2.017e+01  -0.828 0.407654
## TEAM_BATTING_2B         1.263e+00  1.863e-01   6.776 1.57e-11 ***
## TEAM_BATTING_3B        -3.581e-01  1.867e-01  -1.918 0.055194 .
## TEAM_BATTING_HR         6.536e-01  2.431e-01   2.688 0.007231 **
## TEAM_BATTING_BB         3.697e-01  7.163e-02   5.161 2.67e-07 ***
## TEAM_BASERUN_SB         5.272e-02  1.111e-02   4.743 2.23e-06 ***
## TEAM_PITCHING_H        -3.976e-02  1.674e-02  -2.376 0.017590 *
## TEAM_PITCHING_HR       -7.726e-01  2.059e-01  -3.753 0.000179 ***
## TEAM_PITCHING_SO       -6.595e-02  1.581e-02  -4.172 3.14e-05 ***
## TEAM_FIELDING_E        -1.803e-01  2.364e-02  -7.628 3.51e-14 ***
## I(TEAM_BATTING_2B^2)   -4.771e-03  7.432e-04  -6.419 1.67e-10 ***
## I(TEAM_BATTING_3B^2)    1.065e-02  3.682e-03   2.892 0.003866 **
## I(TEAM_BATTING_HR^2)   -7.648e-03  2.927e-03  -2.613 0.009042 **
## I(TEAM_BATTING_BB^2)   -1.206e-03  2.832e-04  -4.259 2.14e-05 ***
## I(TEAM_BATTING_SO^2)    1.428e-04  3.043e-05   4.694 2.85e-06 ***
## I(TEAM_BASERUN_CS^2)   -5.503e-04  3.604e-04  -1.527 0.126919
## I(TEAM_PITCHING_H^2)    2.021e-05  6.340e-06   3.188 0.001452 **
## I(TEAM_PITCHING_HR^2)   9.262e-03  2.272e-03   4.076 4.75e-05 ***
## I(TEAM_PITCHING_SO^2)   5.738e-05  2.259e-05   2.540 0.011160 *
## I(TEAM_FIELDING_E^2)    2.295e-04  4.246e-05   5.405 7.17e-08 ***
## I(TEAM_FIELDING_DP^2) -2.476e-03  3.470e-04  -7.137 1.29e-12 ***
## I(TEAM_BATTING_1B^3)    1.075e-08  7.740e-10  13.888  < 2e-16 ***
## I(TEAM_BATTING_2B^3)    5.920e-06  9.696e-07   6.106 1.20e-09 ***
## I(TEAM_BATTING_3B^3)   -7.825e-05  2.806e-05  -2.789 0.005331 **
## I(TEAM_BATTING_HR^3)    3.197e-05  1.467e-05   2.179 0.029458 *
## I(TEAM_BATTING_BB^3)    1.778e-06  4.539e-07   3.918 9.20e-05 ***
## I(TEAM_BATTING_SO^3)   -1.834e-07  4.686e-08  -3.915 9.33e-05 ***
## I(TEAM_BASERUN_SB^3)   -3.601e-07  1.410e-07  -2.555 0.010698 *
## I(TEAM_BASERUN_CS^3)    3.260e-06  2.113e-06   1.543 0.122946
## I(TEAM_PITCHING_H^3)   -2.656e-09  7.386e-10  -3.596 0.000330 ***
## I(TEAM_PITCHING_HR^3)  -3.574e-05  1.003e-05  -3.564 0.000373 ***
## I(TEAM_PITCHING_BB^3)  -7.060e-08  1.022e-08  -6.908 6.39e-12 ***
## I(TEAM_PITCHING_SO^3)  -2.523e-08  1.145e-08  -2.204 0.027640 *
## I(TEAM_FIELDING_DP^3)   8.972e-06  1.495e-06   6.003 2.26e-09 ***
## I(TEAM_BATTING_3B^4)    1.684e-07  7.006e-08   2.404 0.016294 *
## I(TEAM_BATTING_HR^4)   -4.607e-08  2.575e-08  -1.789 0.073787 .
## I(TEAM_BATTING_BB^4)   -8.716e-10  2.523e-10  -3.455 0.000561 ***
## I(TEAM_BATTING_SO^4)    6.131e-11  2.042e-11   3.003 0.002703 **
## I(TEAM_BASERUN_SB^4)    4.673e-10  2.024e-10   2.309 0.021042 *
## I(TEAM_PITCHING_HR^4)   4.609e-08  1.489e-08   3.095 0.001992 **
## I(TEAM_PITCHING_BB^4)   3.980e-11  6.218e-12   6.401 1.87e-10 ***
## I(TEAM_PITCHING_SO^4)   3.872e-12  1.819e-12   2.129 0.033389 *
## ---
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.19 on 2233 degrees of freedom
## Multiple R-squared:  0.4052, Adjusted R-squared:  0.3943
## F-statistic:  37.1 on 41 and 2233 DF,  p-value: < 2.2e-16
```

For this it can be seen that the Adjusted R^2 is 0.3943 which better than all the previous models. The Adjusted R^2 value is almost 0.4 which is good. The p value is very less for this model similar to the previous model.

The plots for this model are as follows:



5. **Excluding variables with Multicollinearity**

For this model I decided to ignore the variables that showed multicollinearity. I removed the below variables:

TEAM_BATTING_SO

TEAM_PITCHING_BB

TEAM_PITCHING_H

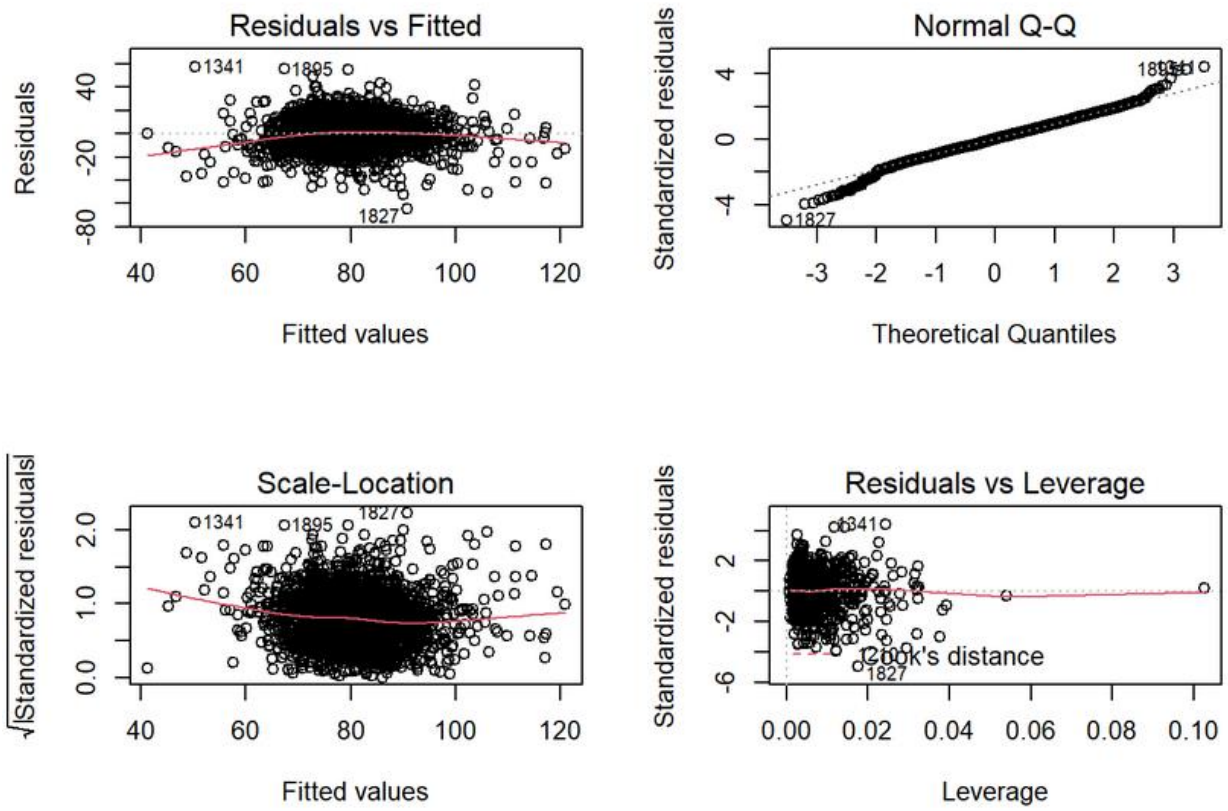
TEAM_PITCHING_HR

The linear regression summary is as follows:

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - TEAM_BATTING_SO - TEAM_PITCHING_BB -
##     TEAM_PITCHING_H - TEAM_PITCHING_HR, data = train_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.834  -8.089   0.262   8.451  57.598
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      30.782093   4.876182   6.313 3.29e-10 ***
## TEAM_BATTING_H    0.039246   0.003284  11.951  < 2e-16 ***
## TEAM_BATTING_2B  -0.012053   0.009020  -1.336  0.18158
## TEAM_BATTING_3B   0.089511   0.016668   5.370 8.67e-08 ***
## TEAM_BATTING_HR   0.057120   0.008850   6.454 1.33e-10 ***
## TEAM_BATTING_BB   0.032662   0.002953  11.059  < 2e-16 ***
## TEAM_BASERUN_SB   0.020187   0.004164   4.847 1.34e-06 ***
## TEAM_BASERUN_CS   0.018095   0.015907   1.138  0.25542
## TEAM_PITCHING_SO -0.005450   0.001516  -3.596  0.00033 ***
## TEAM_FIELDING_E  -0.045863   0.005517  -8.313  < 2e-16 ***
## TEAM_FIELDING_DP -0.155824   0.013415 -11.615  < 2e-16 ***
## TEAM_BATTING_1B         NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.19 on 2264 degrees of freedom
## Multiple R-squared:  0.2942, Adjusted R-squared:  0.2911
## F-statistic: 94.38 on 10 and 2264 DF,  p-value: < 2.2e-16
```

For this it can be seen that the Adjusted R^2 is 0.2911 which is not that great as it less than 0.4. The p value is very less for this model similar to the previous model.

The plots for this model are as follows:



6. **Excluding variables having insignificant p values**

For this model I decided to ignore the variables that showed insignificant p values. I removed the below variables:

TEAM_BATTING_SO

TEAM_PITCHING_BB

TEAM_PITCHING_H

TEAM_PITCHING_HR

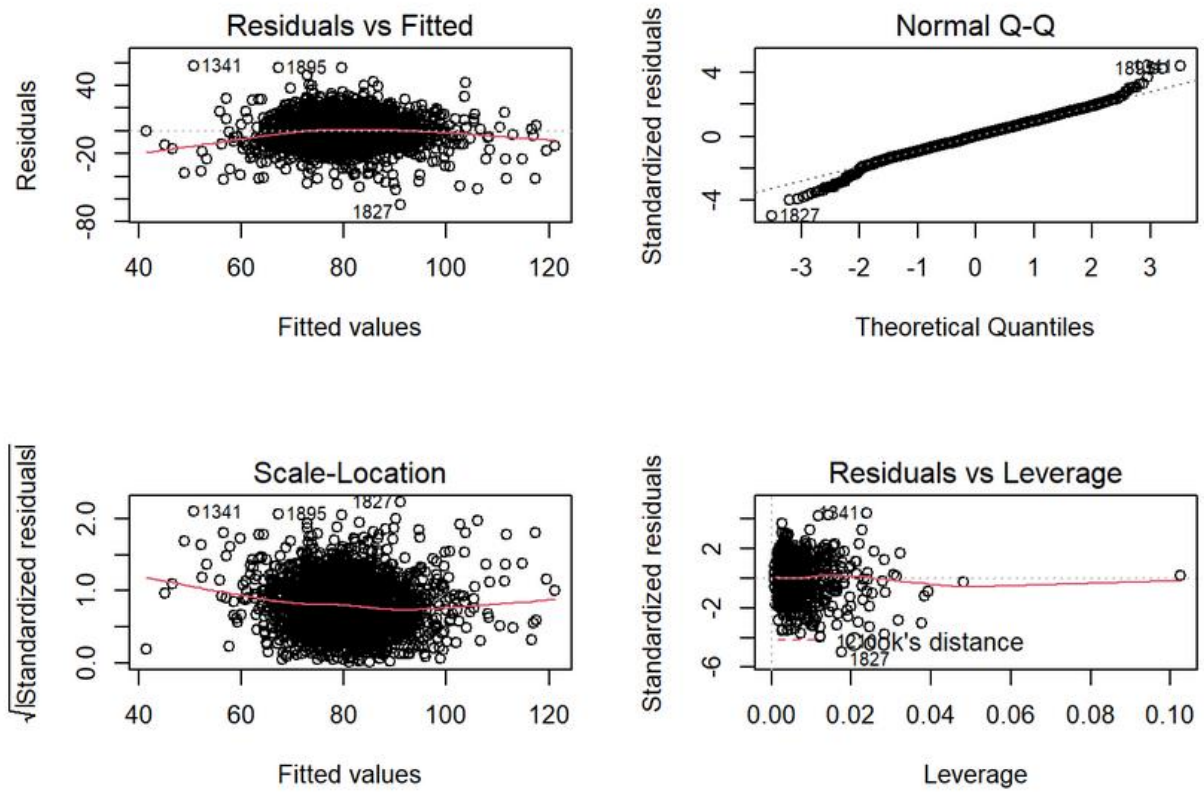
TEAM_BASERUN_CS

The linear regression summary is as follows:

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - TEAM_BATTING_SO - TEAM_PITCHING_BB -
##     TEAM_PITCHING_H - TEAM_PITCHING_HR - TEAM_BASERUN_CS, data = train_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.994  -8.015   0.249   8.425  57.277
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      32.044558   4.748519   6.748 1.89e-11 ***
## TEAM_BATTING_H    0.039052   0.003280  11.907  < 2e-16 ***
## TEAM_BATTING_2B  -0.011197   0.008989  -1.246 0.213044
## TEAM_BATTING_3B   0.088809   0.016658   5.331 1.07e-07 ***
## TEAM_BATTING_HR   0.055487   0.008734   6.353 2.54e-10 ***
## TEAM_BATTING_BB   0.032743   0.002953  11.089  < 2e-16 ***
## TEAM_BASERUN_SB   0.021076   0.004090   5.153 2.79e-07 ***
## TEAM_PITCHING_SO -0.005558   0.001513  -3.674 0.000244 ***
## TEAM_FIELDING_E  -0.046147   0.005511  -8.373  < 2e-16 ***
## TEAM_FIELDING_DP -0.156115   0.013414 -11.639  < 2e-16 ***
## TEAM_BATTING_1B         NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.19 on 2265 degrees of freedom
## Multiple R-squared:  0.2938, Adjusted R-squared:  0.291
## F-statistic: 104.7 on 9 and 2265 DF,  p-value: < 2.2e-16
```

For this it can be seen that the Adjusted R^2 is 0.291 which is not that great as it less than 0.4. The p value is very less for this model similar to the previous model.

The plots for this model are as follows:



4. Select model

From the 6 models I build it was clear that model 4 using the polynomial regression had the best Adjusted R^2 value. The p values for all the models remained similar despite excluding multicollinearity and p values in the last 2 models.

So I select model 4 and use it to predict on the test dataset.

The final output is exported to a .csv file. Below is the output file.

https://github.com/irene908/DATA621/blob/main/DATA621_Assignment1.csv

5. Annexure

R code:

```
---
title: "DATA621 Assignment 1"
author: "Irene Jacob"
date: "9/25/2021"
output:
```

```
  html_document:
    df_print: paged
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
library(corrplot)
library("base")
library(MASS)
library(rpart.plot)
library(forecast)
library(GGally)
library(tibble)
library(tidyr)
library(tidyverse)
library(dplyr)
library(reshape2)
library(tidymodels)
```

Assignment 1

1. Data Exploration

```{r}
train <- read.csv("https://raw.githubusercontent.com/irene908/DATA621/main/moneyball-training-data.csv") %>%select(-INDEX)

test <- read.csv("https://raw.githubusercontent.com/irene908/DATA621/main/moneyball-evaluation-data.csv") %>%select(-INDEX)
```

```{r}
dim(train)
```

```{r}
summary(train)
```

```{r}
train %>% gather() %>% ggplot(aes(x= value)) + geom_density(fill='light blue') + facet_wrap(~key, scales = 'free')
```

```{r}
```
```

```
train_new <- train %>% gather(key = 'key', value = 'value')

ggplot(train_new,aes(x = key, y = value)) +geom_boxplot()+coord_cartesian(ylim = c(0, 1000))+
theme(axis.text.x=element_text(angle=45, hjust=1))
```

```{r,fig.height = 7, fig.width = 7}
train %>% cor(., use = "complete.obs") %>% corrplot(., type = "upper", diag = FALSE)
```

```{r,fig.height = 7, fig.width = 7}

train %>% gather(key, value, -TARGET_WINS) %>% ggplot(., aes(value, TARGET_WINS)) +  geom_point(
color="purple") + geom_smooth(method = "lm", se = FALSE, color = "black") + facet_wrap(~key, scales
="free", ncol = 3)
```

```{r}

train %>% gather(key, value) %>% filter(is.na(value)) %>% group_by(key) %>% tally() %>% mutate(p = n /
nrow(train) * 100) %>% mutate(p = paste0(round(p, ifelse(p < 10, 1, 0)), "%")) %>% arrange(desc(n))
%>% rename(`Variable` = key, `Count` = n, `Percentage` = p)

```
2. Data Preparation

Handling missing data

```{r}
# Drop the BATTING_HBP field
train <- train %>% select(-TEAM_BATTING_HBP)

train_new <- train

train_new$TEAM_PITCHING_SO <- ifelse(train_new$TEAM_PITCHING_SO > 4000, NA,
train_new$TEAM_PITCHING_SO)

train_new$TEAM_PITCHING_H <- ifelse(train_new$TEAM_PITCHING_H > 5000, NA,
train_new$TEAM_PITCHING_H)

train_new$TEAM_PITCHING_BB <- ifelse(train_new$TEAM_PITCHING_BB > 2000, NA,
train_new$TEAM_PITCHING_BB)

train_new$TEAM_FIELDING_E <- ifelse(train_new$TEAM_FIELDING_E > 480, NA,
train_new$TEAM_FIELDING_E)
```

```

```{r}

for(i in 1:ncol(train_new)){
  train_new[is.na(train_new[,i]), i] <- mean(train_new[,i], na.rm = TRUE)
}

train_new <- train_new %>%
  filter(TARGET_WINS != 0)

```


```{r}

summary(train_new)
```


Feature Engineering

```{r}

single_Feature <- function(df){ df %>% mutate(TEAM_BATTING_1B = TEAM_BATTING_H -
TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) }

train_new <- single_Feature(train_new)
test <- single_Feature(test)
```

View the final prepared data

```{r}

train_new %>% gather(key, value) %>% ggplot(., aes(value)) + geom_density(fill='blue') +
facet_wrap(~key, scales ="free")

```

```{r}

#summary of the prepared train data

summary(train_new)

```
```

Transformation

```{r echo=FALSE, fig.width=15, message=FALSE, warning=FALSE}

# created empty data frame to store transformed variables
train_temp <- data.frame(matrix(ncol = 1, nrow = length(train_new$TARGET_WINS)))

# performed boxcox transformation after identifying proper lambda
train_temp$TEAM_BATTING_3B <- train_new$TEAM_BATTING_3B
BATTING_3B_Lambda <- BoxCox.lambda(train_new$TEAM_BATTING_3B)
train_temp$BATTING_3B <- log(train_new$TEAM_BATTING_3B)

# performed boxcox transformation after identifying proper lambda
train_temp$TEAM_BATTING_HR <- train_new$TEAM_BATTING_HR
BATTING_HR_Lambda <- BoxCox.lambda(train_new$TEAM_BATTING_HR)
train_temp$BATTING_HR <- BoxCox(train_new$TEAM_BATTING_HR, BATTING_HR_Lambda)

# performed a log transformation
train_temp$TEAM_PITCHING_BB <- train_new$TEAM_PITCHING_BB
train_temp$PITCHING_BB <- log(train_new$TEAM_PITCHING_BB)

# performed a log transformation
train_temp$TEAM_PITCHING_SO <- train_new$TEAM_PITCHING_SO
train_temp$PITCHING_SO <- log(train_new$TEAM_PITCHING_SO)

# performed an inverse log transformation
train_temp$TEAM_FIELDING_E <- train_new$TEAM_FIELDING_E
train_temp$FIELDING_E <- 1/log(train_new$TEAM_FIELDING_E)

# performed a log transformation
train_temp$TEAM_BASERUN_SB <- train_new$TEAM_BASERUN_SB
train_temp$BASERUN_SB <- log(train_new$TEAM_BASERUN_SB)

train_temp <- train_temp[, 2:13]
train_tmp <- train_temp %>% gather(key = 'key', value = 'value')


ggplot(train_tmp, aes(x=value)) + geom_density() + geom_histogram() + facet_wrap(~key, scales
="free", ncol = 6)

#hist(train_temp)
```

Finalizing the dataset for model building

```{r}
# Build clean dataframe with transformation

train_new <- data.frame(cbind(train_new, BATTING_3B = train_temp$BATTING_3B, BATTING_HR =
train_temp$BATTING_HR,BASERUN_SB = train_temp$BASERUN_SB, PITCHING_BB =
train_temp$PITCHING_BB, PITCHING_SO = train_temp$PITCHING_SO, FIELDING_E =
train_temp$FIELDING_E))

is.na(train_new) <- sapply(train_new, is.infinite)

# Impute missing value with the mean

train_new$BATTING_3B[is.na(train_new$BATTING_3B)] <- mean(train_new$BATTING_3B, na.rm =
TRUE)
train_new$BASERUN_SB[is.na(train_new$BASERUN_SB)] <- mean(train_new$BASERUN_SB, na.rm =
TRUE)
train_new$PITCHING_SO[is.na(train_new$PITCHING_SO)] <- mean(train_new$PITCHING_SO, na.rm =
TRUE)
```

3. Build models

```{r}
x<-c(1,17,18,19,20,21,22)
train_df <- train_new[,x]
train_new <- train_new[,1:16]

```

Simple model using the transformed data

selecting a few high correlation variables

```{r}
colnames(train_df)<- c('TARGET_WINS','TEAM_BATTING_3B','TEAM_BATTING_HR',
'TEAM_BASERUN_SB', 'TEAM_PITCHING_BB', 'TEAM_PITCHING_SO', 'TEAM_FIELDING_E')
train_simple <- lm(TARGET_WINS ~ TEAM_BATTING_HR + TEAM_PITCHING_BB + TEAM_FIELDING_E,
data = train_df)
summary(train_simple)
par(mfrow = c(2, 2))
plot(train_simple)
```

Simple model without the transformed data

```{r}

train_simple_t <- lm(TARGET_WINS ~ TEAM_BATTING_HR + TEAM_PITCHING_BB + TEAM_FIELDING_E,
data = train_new)
summary(train_simple_t)
par(mfrow = c(2, 2))
plot(train_simple_t)

````

Full model without the transformed data

```{r}
train_full <- lm(TARGET_WINS ~., data = train_new)
summary(train_full)
par(mfrow = c(2, 2))
plot(train_full)
```


Polynomial Regression without the transformed data

```{r}

train_poly <- "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_H + TEAM_PITCHING_HR + TEAM_PITCHING_BB +
TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP + I(TEAM_BATTING_1B^2)+
I(TEAM_BATTING_2B^2) + I(TEAM_BATTING_3B^2) + I(TEAM_BATTING_HR^2) +
I(TEAM_BATTING_BB^2) + I(TEAM_BATTING_SO^2) + I(TEAM_BASERUN_SB^2) +
I(TEAM_BASERUN_CS^2) + I(TEAM_PITCHING_H^2) + I(TEAM_PITCHING_HR^2) +
I(TEAM_PITCHING_BB^2) + I(TEAM_PITCHING_SO^2) + I(TEAM_FIELDING_E^2) +
I(TEAM_FIELDING_DP^2)  + I(TEAM_BATTING_1B^3)+ I(TEAM_BATTING_2B^3) +
I(TEAM_BATTING_3B^3) + I(TEAM_BATTING_HR^3) + I(TEAM_BATTING_BB^3) +
I(TEAM_BATTING_SO^3) + I(TEAM_BASERUN_SB^3) + I(TEAM_BASERUN_CS^3) +
I(TEAM_PITCHING_H^3) + I(TEAM_PITCHING_HR^3) + I(TEAM_PITCHING_BB^3) +
I(TEAM_PITCHING_SO^3) + I(TEAM_FIELDING_E^3) + I(TEAM_FIELDING_DP^3)
+I(TEAM_BATTING_1B^4) + I(TEAM_BATTING_2B^4) + I(TEAM_BATTING_3B^4) +
I(TEAM_BATTING_HR^4) + I(TEAM_BATTING_BB^4) + I(TEAM_BATTING_SO^4) +
I(TEAM_BASERUN_SB^4) + I(TEAM_BASERUN_CS^4) + I(TEAM_PITCHING_H^4) +
I(TEAM_PITCHING_HR^4) + I(TEAM_PITCHING_BB^4) + I(TEAM_PITCHING_SO^4) +
I(TEAM_FIELDING_E^4) + I(TEAM_FIELDING_DP^4) "
train_poly_lm <- lm(train_poly, train_new)
train_poly_lm_stepback <- MASS::stepAIC(train_poly_lm, direction="backward", trace = F)
train_poly_lm_call <- summary(train_poly_lm_stepback)$call
train_poly_lm_stepback <- lm(train_poly_lm_call[2], train_new)
summary(train_poly_lm_stepback)
```

```
par(mfrow = c(2, 2))
plot(train_poly_lm_stepback)
```

excluding variables with Multicollinearity

```{r}
train_multi <- lm(TARGET_WINS ~.- TEAM_BATTING_SO- TEAM_PITCHING_BB- TEAM_PITCHING_H-
TEAM_PITCHING_HR, data = train_new)
summary(train_multi)
par(mfrow = c(2, 2))
plot(train_multi)

```

Excluding variables having insignificant p values

```{r}

train_p <- lm(TARGET_WINS ~.- TEAM_BATTING_SO - TEAM_PITCHING_BB - TEAM_PITCHING_H -
TEAM_PITCHING_HR - TEAM_BASERUN_CS, data = train_new)
summary(train_p)
par(mfrow = c(2, 2))
plot(train_p)

```

4. Select Model

```{r}
test$TARGET_WINS <- round(predict(train_poly_lm_stepback, test), 0)
```

```{r}
write.csv(test,"DATA621_Assignment1.csv")
```