# Homework 1

*Group 2*

*02/27/2019*

## Part 1: Overview

The purpose of this assignment is to explore, analyze and model professional baseball team performance from the years 1871 to 2006. Our objective is to build a multiple linear regression model on the provided data to predict the number of wins for the team.**

### Dependencies

Replication of our work requires the following dependencies:

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(car)
library(corrplot)
library(Hmisc)
library(psych)
library (MASS)
library(lmtest)
library(faraway)
library(knitr)
library(kableExtra)
```

### Data Preparation

We first read the training and test data from the csv files located in our repository.

```
train_data <- "moneyball-training-data.csv"
test_data <- "moneyball-evaluation-data.csv"
moneyball_data <- read.csv(train_data, header=TRUE, stringsAsFactors=FALSE, fileEncoding="latin1")
test_data <- read.csv(test_data, header = TRUE, stringsAsFactors = FALSE)
```

## Part 2: Data Exploration

Upon review of the dataset, we found large amounts of missing variables. We choose to replace the empty data points with the mean of that data column. The method was preferable to removing the data with omitted values, because that would have removed 90% of the provided data.

We calculated the appropriate means to compensate for the incomplete data below:

```
sapply(moneyball_data, function(y) sum(length(which(is.na(y)))))/nrow(moneyball_data)*100
```

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

```
sapply(test_data, function(y) sum(length(which(is.na(y)))))/nrow(moneyball_data)*100
```

```
##            INDEX   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##        0.0000000        0.0000000        0.0000000        0.0000000
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##        0.0000000        0.0000000        0.7908612        0.5711775
##  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H TEAM_PITCHING_HR
##        3.8224956       10.5448155        0.0000000        0.0000000
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##        0.0000000        0.7908612        0.0000000        1.3620387
```

We also choose to removed “index” and “TEAM_BATTING_HBP” columns as “TEAM_BATTING_HBP” has 92% of missing values" and “index” was just a counter.

```
moneyball_data<-subset(moneyball_data, select = -c(INDEX))
moneyball<-subset(moneyball_data, select = -c(TEAM_BATTING_HBP))

test_data <- subset(test_data, select = -c(INDEX))
test_data <- subset(test_data, select = -c(TEAM_BATTING_HBP))
```

## Summary Statistics

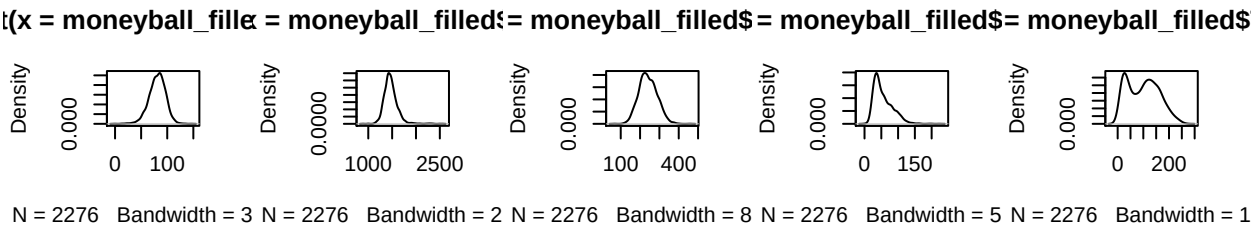
Through these steps, we replaced the missing data with the appropriate mean data.

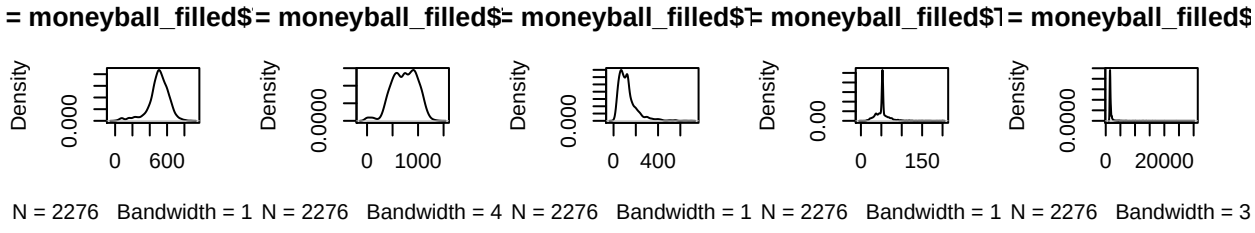
```
replace_mean <- function(x){
  x <- as.numeric(as.character(x))
  x[is.na(x)] = mean(x, na.rm=TRUE)
  return(x)
}

moneyball_filled <- apply(moneyball, 2, replace_mean)
moneyball_filled <- as.data.frame(moneyball_filled)

test_filled <- apply(test_data, 2, replace_mean)
test_filled <- as.data.frame(test_data)
```

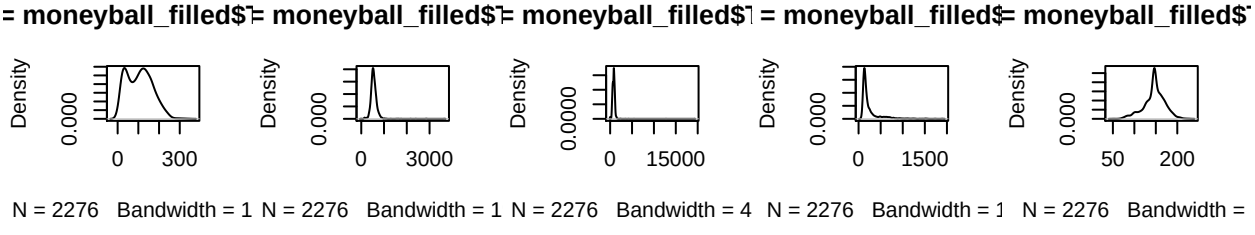
| | |
|---|---|
| TEAM_BATTING_H | 1469.3900 |
| TEAM_BATTING_2B | 241.3205 |
| TEAM_BATTING_3B | 55.9112 |
| TEAM_BATTING_HR | 95.6332 |
| TEAM_BATTING_BB | 498.9575 |
| TEAM_BATTING_SO | NA |
| TEAM_BASERUN_SB | NA |
| TEAM_BASERUN_CS | NA |
| TEAM_PITCHING_H | 1813.4633 |
| TEAM_PITCHING_HR | 102.1467 |
| TEAM_PITCHING_BB | 552.4170 |
| TEAM_PITCHING_SO | NA |
| TEAM_FIELDING_E | 249.7490 |
| TEAM_FIELDING_DP | NA |

### Histogram

Now that we have a 'good' dataset, we can look at some histograms for each data vector. Our output suggests that most variables are fairly normally distributed and span many orders of magnitude. This tells us that our model will have some kind scaling factor between our data vectors.

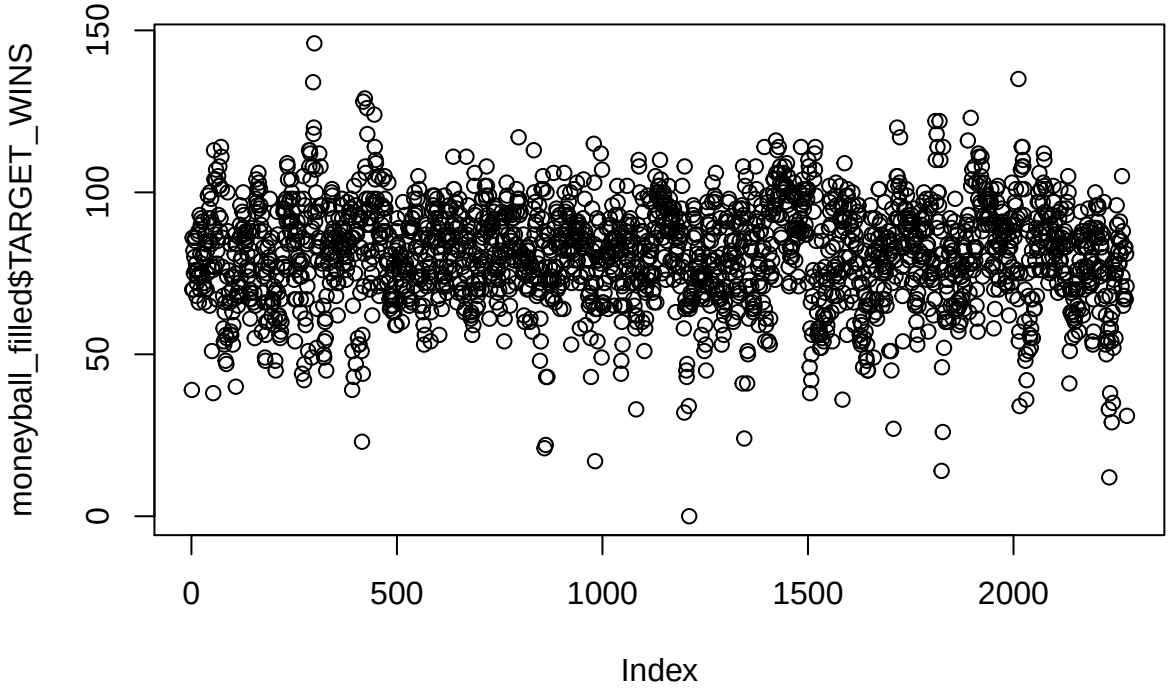


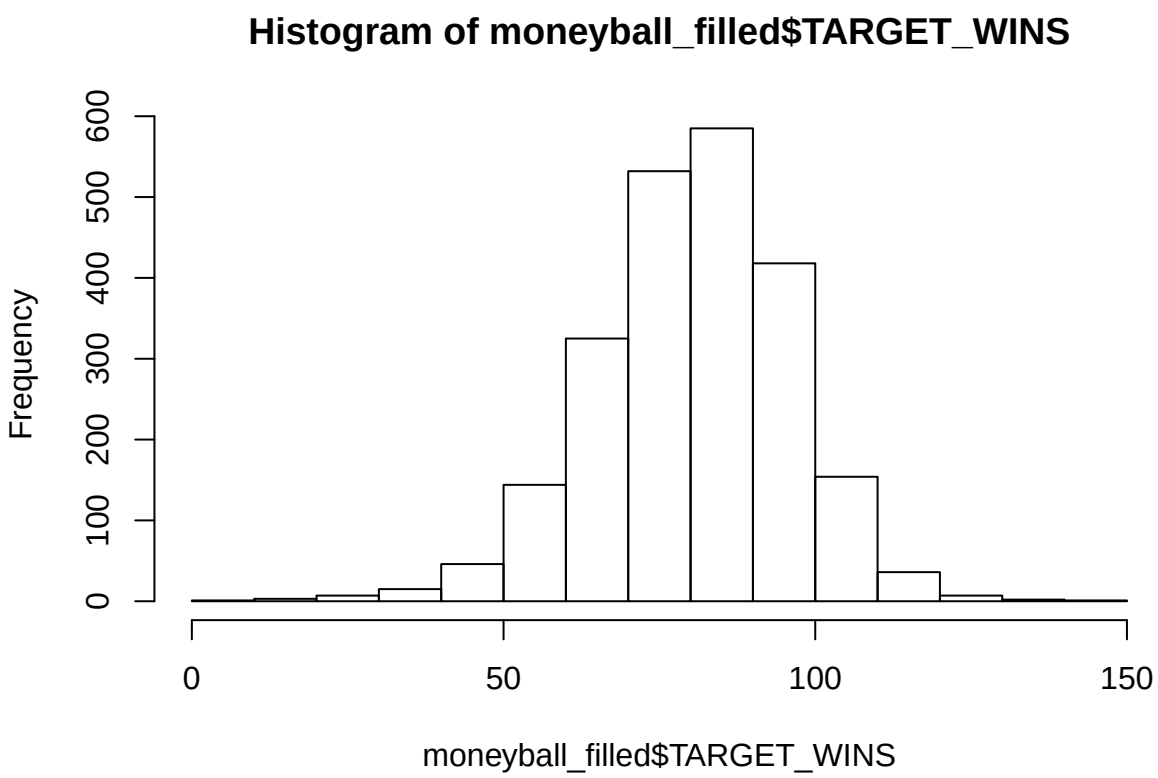



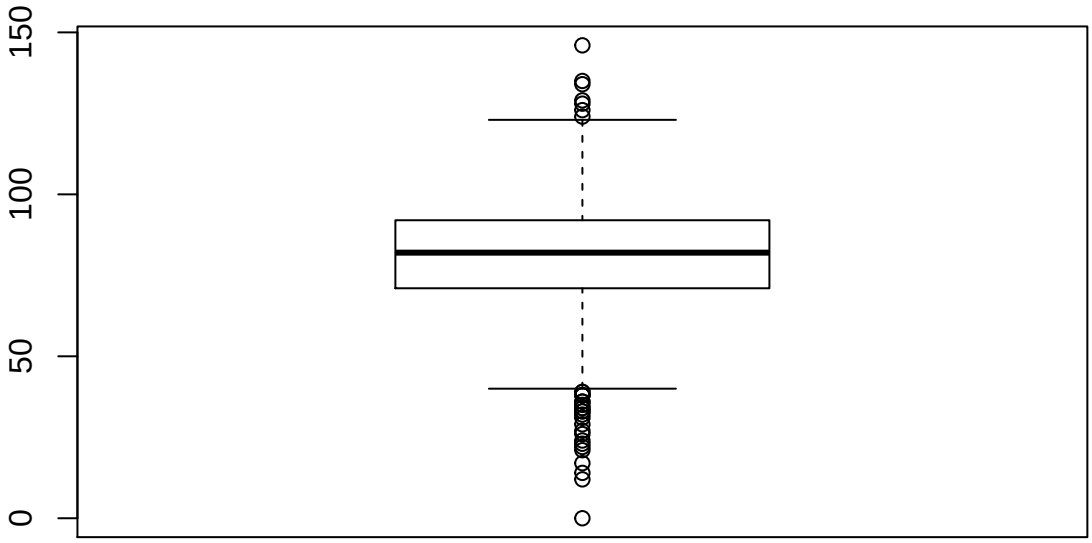




### Target Wins Variable

To better understand the goal of our model, we examined the targeted wins variable. Below are the plots that show this variable follows a normal, unimodal distribution that is slightly skewed to the left.


moneyball_filled$TARGET_WINS
0
50
100
150
0
500
1000
1500
2000
Index

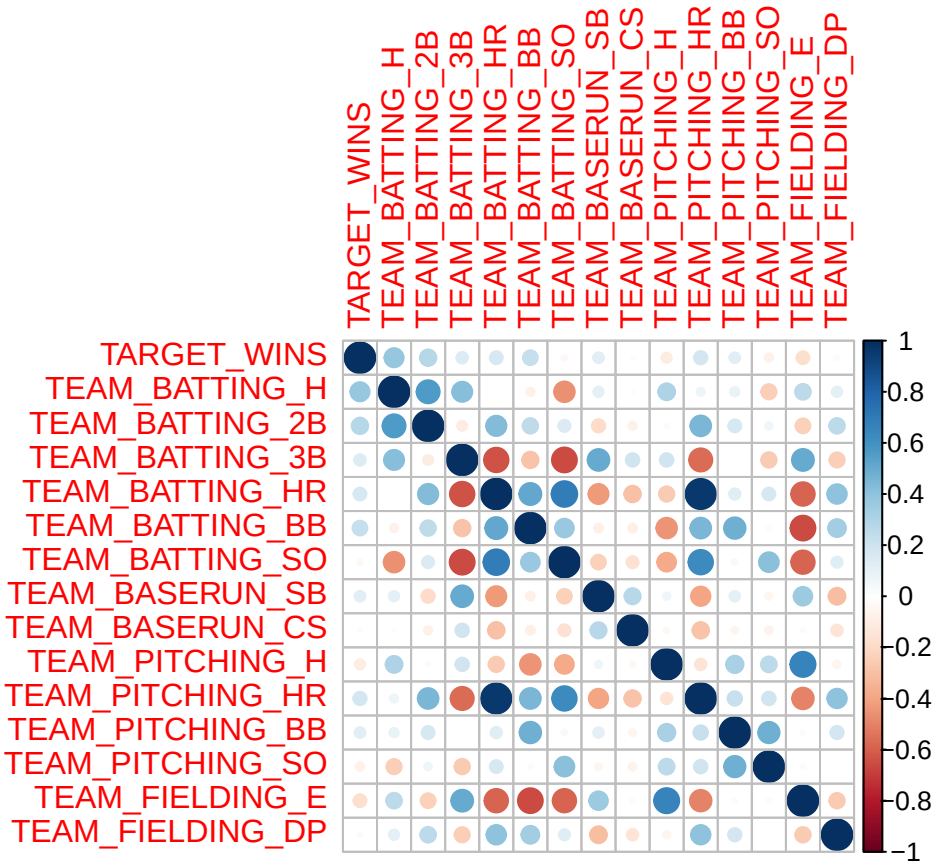

Histogram of moneyball_filled$TARGET_WINS
Frequency
0
100
200
300
400
500
600
0
50
100
150
moneyball_filled$TARGET_WINS




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   71.00   82.00   80.79   92.00  146.00
```

## Correlation

We then checked for correlation among our dependent variables, as all variables are numeric we will rely on correlation. Below is a correlation plot that highlights the correlation between various data vectors. Dark blue is a high, positive correlation and dark red is a large negative correlation.




Here, we notice several variables have poor correlation with the target variable (p<0.1):
- TEAM_FIELDING_E
- TEAM_BASERUN_CS
- TEAM_BATTING_SO
- TEAM_BATTING_3B

However, others have strong correlation between each others (>0.6):
- TEAM_PITCHING_HR vs TEAM_BATTING_HR (0.969)
- TEAM_BATTING_HR vs TEAM_BATTING_SO (0.693)
- TEAM_BATTING_3B vs TEAM_BATTING_SO (-0.656)

Due to co-linearity or statistical irrelevance, we can remove: TEAM_FIELDING_E, TEAM_BASERUN_CS, TEAM_BATTING_SO, TEAM_BATTING_3B, and TEAM_BATTING_HR.

# Part 3: Modeling

## Model 1

We started with a naive model that uses all of the data vectors. We got an adjusted $R^2$ value of 31.4%.

```
##
## 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 = moneyball_filled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.994  -8.576   0.136   8.345  58.628
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.502e+01  5.397e+00   4.636 3.75e-06 ***
## TEAM_BATTING_H    4.824e-02  3.687e-03  13.085  < 2e-16 ***
## TEAM_BATTING_2B  -2.006e-02  9.152e-03  -2.192 0.028486 *
## TEAM_BATTING_3B   6.047e-02  1.676e-02   3.608 0.000315 ***
## TEAM_BATTING_HR   5.299e-02  2.743e-02   1.932 0.053488 .
## TEAM_BATTING_BB   1.042e-02  5.818e-03   1.790 0.073544 .
## TEAM_BATTING_SO  -9.349e-03  2.551e-03  -3.665 0.000253 ***
## TEAM_BASERUN_SB   2.949e-02  4.462e-03   6.610 4.78e-11 ***
## TEAM_BASERUN_CS  -1.188e-02  1.614e-02  -0.736 0.461905
## TEAM_PITCHING_H  -7.342e-04  3.676e-04  -1.997 0.045946 *
## TEAM_PITCHING_HR  1.480e-02  2.432e-02   0.609 0.542877
## TEAM_PITCHING_BB  8.891e-05  4.145e-03   0.021 0.982891
## TEAM_PITCHING_SO  2.843e-03  9.187e-04   3.095 0.001994 **
## TEAM_FIELDING_E  -2.112e-02  2.480e-03  -8.516  < 2e-16 ***
## TEAM_FIELDING_DP -1.210e-01  1.302e-02  -9.297  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.04 on 2261 degrees of freedom
## Multiple R-squared:  0.3189, Adjusted R-squared:  0.3147
## F-statistic: 75.63 on 14 and 2261 DF,  p-value: < 2.2e-16
```

## Model 2

Then, we removed the least significant variable, `TEAM_PITCHING_BB`. This yielded a slight increase in our $R^2$ score at 31.5%.

```
##
## 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_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP, data = moneyball_filled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.994  -8.576   0.136   8.345  58.626
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      25.0145796  5.3904993   4.640 3.67e-06 ***
## TEAM_BATTING_H    0.0482393  0.0036807  13.106  < 2e-16 ***
## TEAM_BATTING_2B  -0.0200575  0.0091490  -2.192 0.028457 *
## TEAM_BATTING_3B   0.0604730  0.0167556   3.609 0.000314 ***
```

```
## TEAM_BATTING_HR   0.0527106  0.0240710   2.190 0.028641 *
## TEAM_BATTING_BB   0.0105175  0.0033664   3.124 0.001805 **
## TEAM_BATTING_SO  -0.0093631  0.0024585  -3.809 0.000144 ***
## TEAM_BASERUN_SB   0.0295055  0.0044087   6.693 2.76e-11 ***
## TEAM_BASERUN_CS  -0.0118872  0.0161276  -0.737 0.461155
## TEAM_PITCHING_H  -0.0007306  0.0003283  -2.225 0.026147 *
## TEAM_PITCHING_HR  0.0150659  0.0209923   0.718 0.473025
## TEAM_PITCHING_SO  0.0028567  0.0006717   4.253 2.20e-05 ***
## TEAM_FIELDING_E  -0.0211192  0.0024784  -8.521  < 2e-16 ***
## TEAM_FIELDING_DP -0.1210298  0.0130139  -9.300  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.04 on 2262 degrees of freedom
## Multiple R-squared:  0.3189, Adjusted R-squared:  0.315
## F-statistic: 81.49 on 13 and 2262 DF,  p-value: < 2.2e-16
```

## Model 3

We repeated the above procedure for TEAM_BASERUN_CS, netting us a score of 31.52%. By removing two variables we were able to oh-so-slightly increase our $R^2$ value while reducing the amount of data we have to track and the processing time for tracking it.

```
##
## 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_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_SO +
##     TEAM_FIELDING_E + TEAM_FIELDING_DP, data = moneyball_filled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.905  -8.584   0.124   8.406  58.593
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      24.2348098  5.2851330   4.585 4.78e-06 ***
## TEAM_BATTING_H    0.0482055  0.0036800  13.099  < 2e-16 ***
## TEAM_BATTING_2B  -0.0203302  0.0091405  -2.224 0.026235 *
## TEAM_BATTING_3B   0.0608466  0.0167463   3.633 0.000286 ***
## TEAM_BATTING_HR   0.0543985  0.0239594   2.270 0.023274 *
## TEAM_BATTING_BB   0.0107643  0.0033494   3.214 0.001328 **
## TEAM_BATTING_SO  -0.0093418  0.0024580  -3.800 0.000148 ***
## TEAM_BASERUN_SB   0.0287600  0.0042906   6.703 2.57e-11 ***
## TEAM_PITCHING_H  -0.0007390  0.0003281  -2.253 0.024372 *
## TEAM_PITCHING_HR  0.0147103  0.0209846   0.701 0.483372
## TEAM_PITCHING_SO  0.0028640  0.0006716   4.265 2.08e-05 ***
## TEAM_FIELDING_E  -0.0207217  0.0024188  -8.567  < 2e-16 ***
## TEAM_FIELDING_DP -0.1211603  0.0130114  -9.312  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.04 on 2263 degrees of freedom
```

```
## Multiple R-squared:  0.3188, Adjusted R-squared:  0.3152
## F-statistic: 88.25 on 12 and 2263 DF,  p-value: < 2.2e-16
```

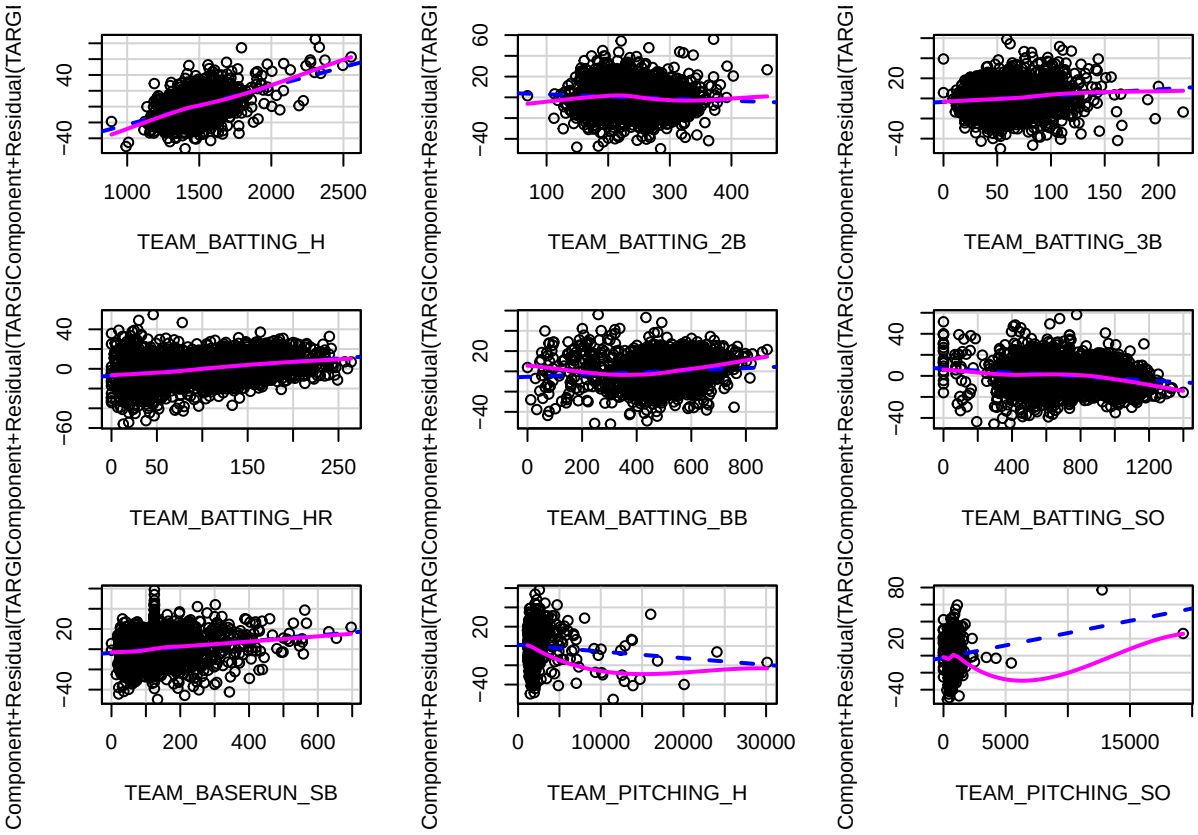
## Model 4

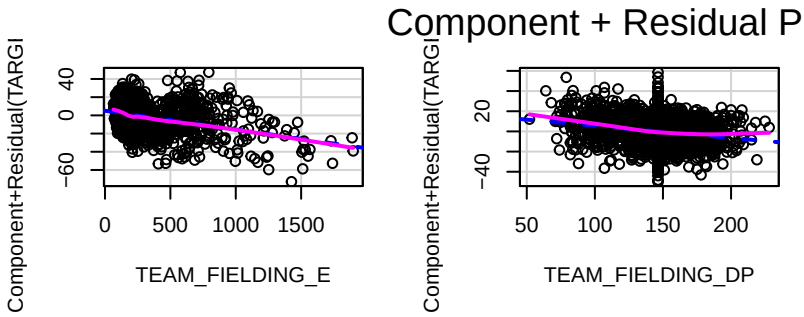
By removing TEAM_PITCHING_HR, we increase our $R^2$ value one last time to 31.53%.

```
##
## 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_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_E +
##     TEAM_FIELDING_DP, data = moneyball_filled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.899  -8.568   0.091   8.397  58.651
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      23.6666983  5.2220414   4.532 6.14e-06 ***
## TEAM_BATTING_H    0.0484570  0.0036621  13.232  < 2e-16 ***
## TEAM_BATTING_2B  -0.0205123  0.0091358  -2.245 0.024847 *
## TEAM_BATTING_3B   0.0624661  0.0165843   3.767 0.000170 ***
## TEAM_BATTING_HR   0.0697785  0.0096266   7.249 5.75e-13 ***
## TEAM_BATTING_BB   0.0107446  0.0033489   3.208 0.001354 **
## TEAM_BATTING_SO  -0.0093019  0.0024571  -3.786 0.000157 ***
## TEAM_BASERUN_SB   0.0287708  0.0042901   6.706 2.51e-11 ***
## TEAM_PITCHING_H  -0.0006920  0.0003211  -2.155 0.031253 *
## TEAM_PITCHING_SO  0.0028867  0.0006707   4.304 1.75e-05 ***
## TEAM_FIELDING_E  -0.0205973  0.0024120  -8.540  < 2e-16 ***
## TEAM_FIELDING_DP -0.1210083  0.0130082  -9.302  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.03 on 2264 degrees of freedom
## Multiple R-squared:  0.3186, Adjusted R-squared:  0.3153
## F-statistic: 96.25 on 11 and 2264 DF,  p-value: < 2.2e-16
```

## Evaluate Non-linearity

Below we use the `crPlots()` function to check for non-linearity.

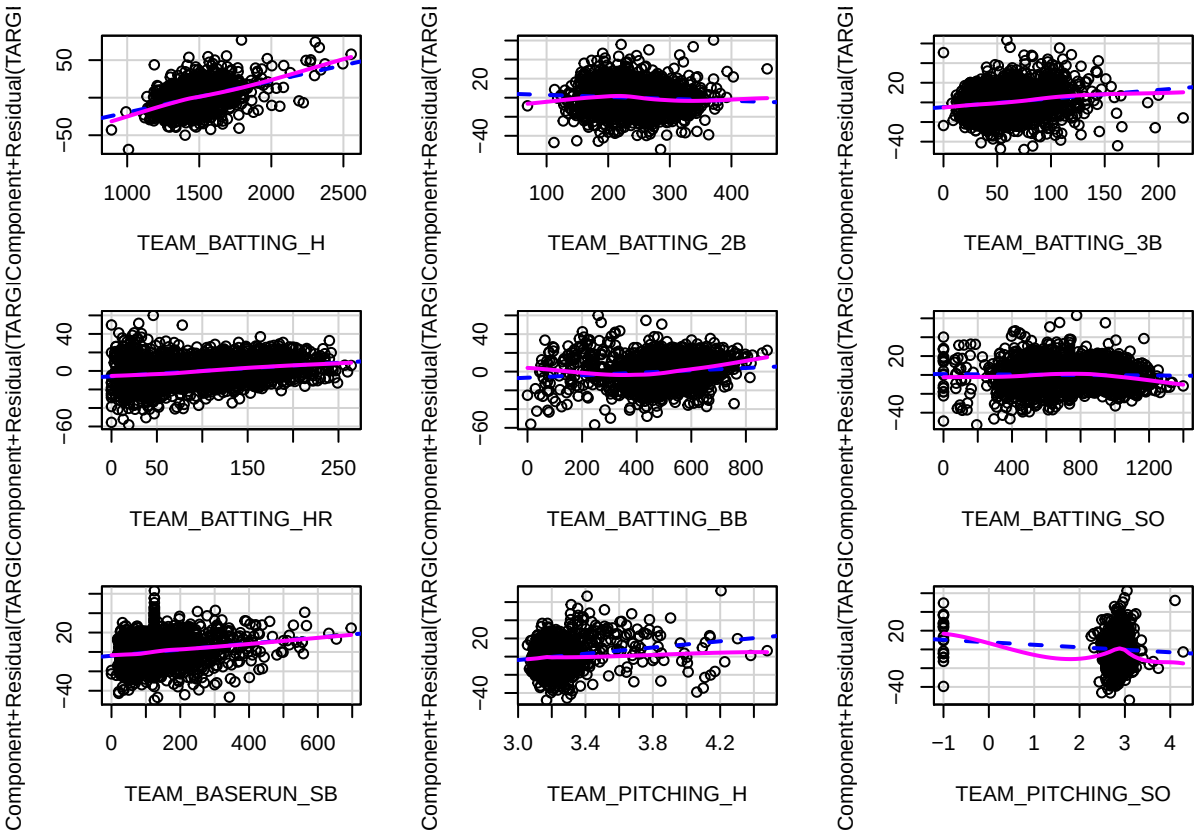

Component+Residual(TARGIComponent+Residual(TARGI
TEAM_BATTING_H
TEAM_BATTING_2B
TEAM_BATTING_3B
TEAM_BATTING_HR
TEAM_BATTING_BB
TEAM_BATTING_SO
TEAM_BASERUN_SB
TEAM_PITCHING_H
TEAM_PITCHING_SO

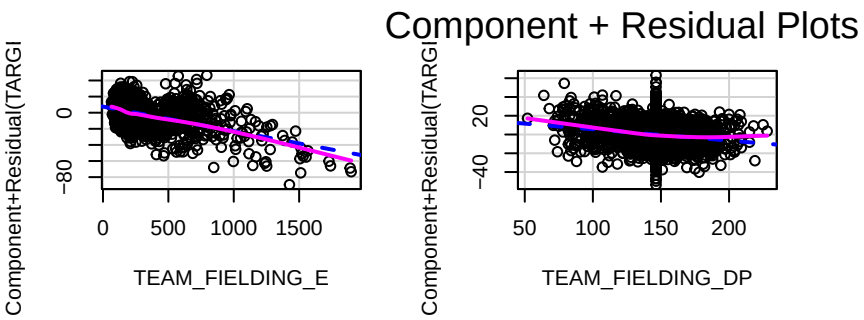



TEAM_PITCHING_H, TEAM_PITCHING_SO did not pass the check for non-linearity. So, we will transform them and refit the model. We are using a log10 transform because these numbers span many orders of magnitude.

```
##
## 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_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_E +
##     TEAM_FIELDING_DP, data = moneyball_filled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.500  -8.353   0.050   8.276  63.152
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -18.385299  13.648651  -1.347 0.178102
## TEAM_BATTING_H     0.041874   0.003784  11.065  < 2e-16 ***
## TEAM_BATTING_2B   -0.020476   0.009106  -2.249 0.024630 *
## TEAM_BATTING_3B    0.087638   0.016862   5.197 2.20e-07 ***
## TEAM_BATTING_HR    0.058540   0.009697   6.037 1.83e-09 ***
## TEAM_BATTING_BB    0.012944   0.003388   3.821 0.000137 ***
## TEAM_BATTING_SO   -0.001186   0.002534  -0.468 0.639742
## TEAM_BASERUN_SB    0.031437   0.004300   7.311 3.65e-13 ***
## TEAM_PITCHING_H   17.140905   4.594173   3.731 0.000195 ***
## TEAM_PITCHING_SO  -2.656620   0.914734  -2.904 0.003717 **
```

```
## TEAM_FIELDING_E   -0.030455   0.002954 -10.309  < 2e-16 ***
## TEAM_FIELDING_DP  -0.120505   0.013004  -9.267  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.03 on 2264 degrees of freedom
## Multiple R-squared:  0.319,  Adjusted R-squared:  0.3157
## F-statistic: 96.42 on 11 and 2264 DF,  p-value: < 2.2e-16
```



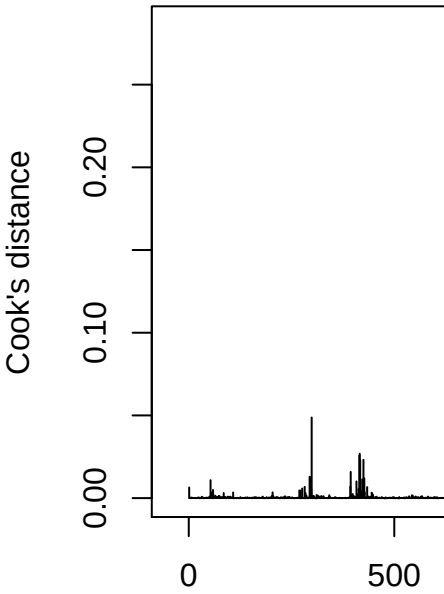



Now we remove `TEAM_BATTING_SO` because it has a p-value $> 0.05$.

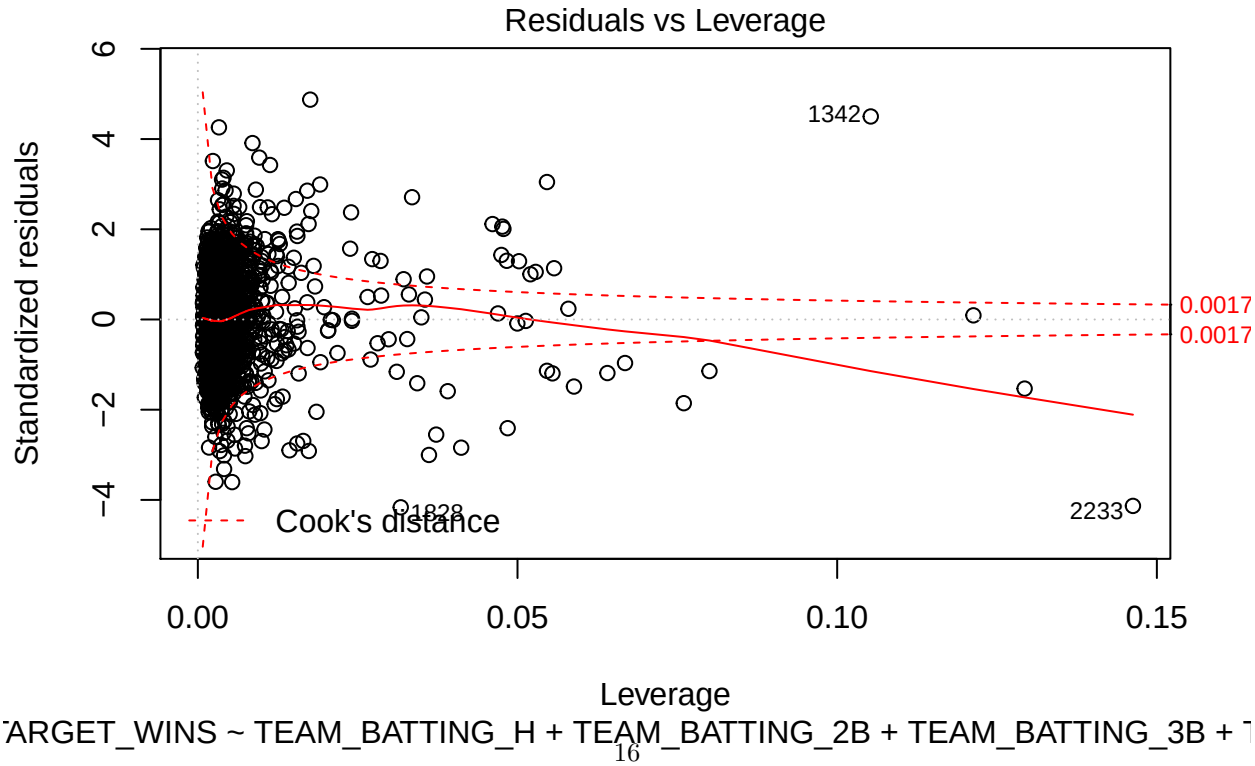
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BASERUN_SB +
##     TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP,
##     data = moneyball_filled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.382  -8.328   0.025   8.211  62.933
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -20.516367  12.864856  -1.595 0.110905
## TEAM_BATTING_H     0.042628   0.003424  12.448  < 2e-16 ***
## TEAM_BATTING_2B   -0.021583   0.008792  -2.455 0.014167 *
## TEAM_BATTING_3B    0.089227   0.016513   5.403 7.23e-08 ***
## TEAM_BATTING_HR    0.055774   0.007688   7.255 5.50e-13 ***
## TEAM_BATTING_BB    0.013293   0.003304   4.023 5.93e-05 ***
## TEAM_BASERUN_SB    0.030879   0.004130   7.476 1.09e-13 ***
## TEAM_PITCHING_H   17.440250   4.548668   3.834 0.000129 ***
## TEAM_PITCHING_SO  -2.847123   0.819083  -3.476 0.000519 ***
## TEAM_FIELDING_E   -0.030494   0.002953 -10.328  < 2e-16 ***
## TEAM_FIELDING_DP  -0.119878   0.012932  -9.270  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.03 on 2265 degrees of freedom
## Multiple R-squared:  0.319,  Adjusted R-squared:  0.316
## F-statistic: 106.1 on 10 and 2265 DF,  p-value: < 2.2e-16
```

### Eliminating Outliers




Then, we used Cook's distance to identify extreme values, removing them as necessary.

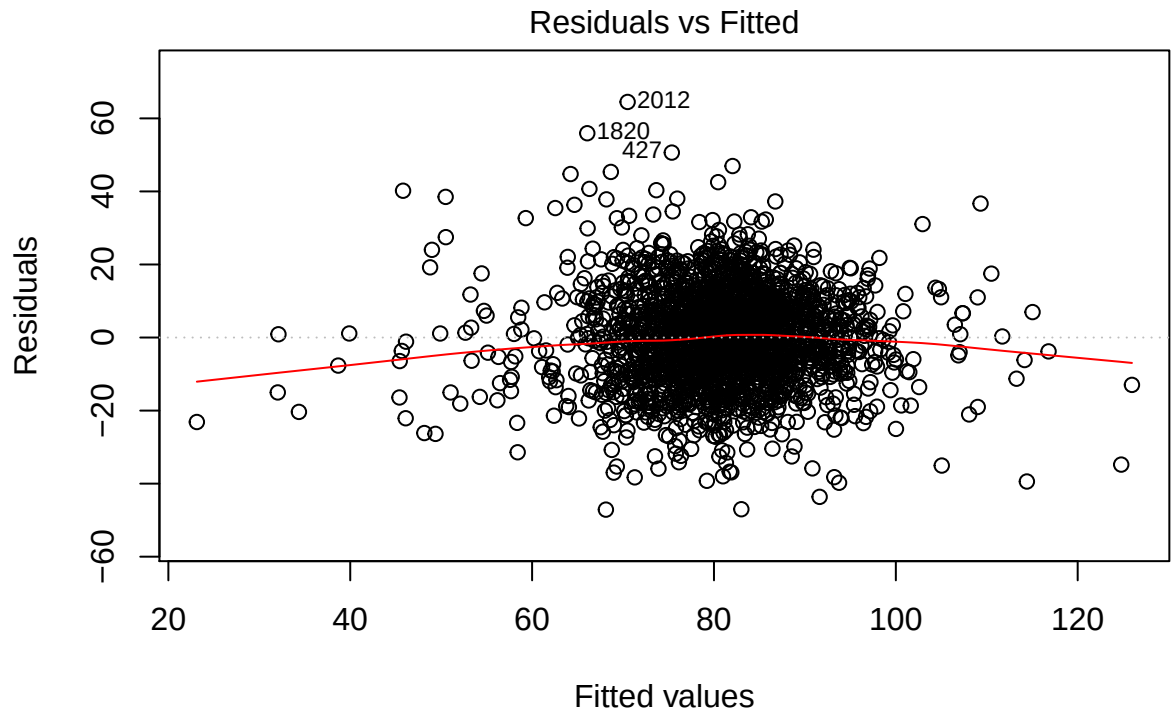


Then, we re-fit the model to the new data, yielding our highest $R^2$ value of 31.57%.

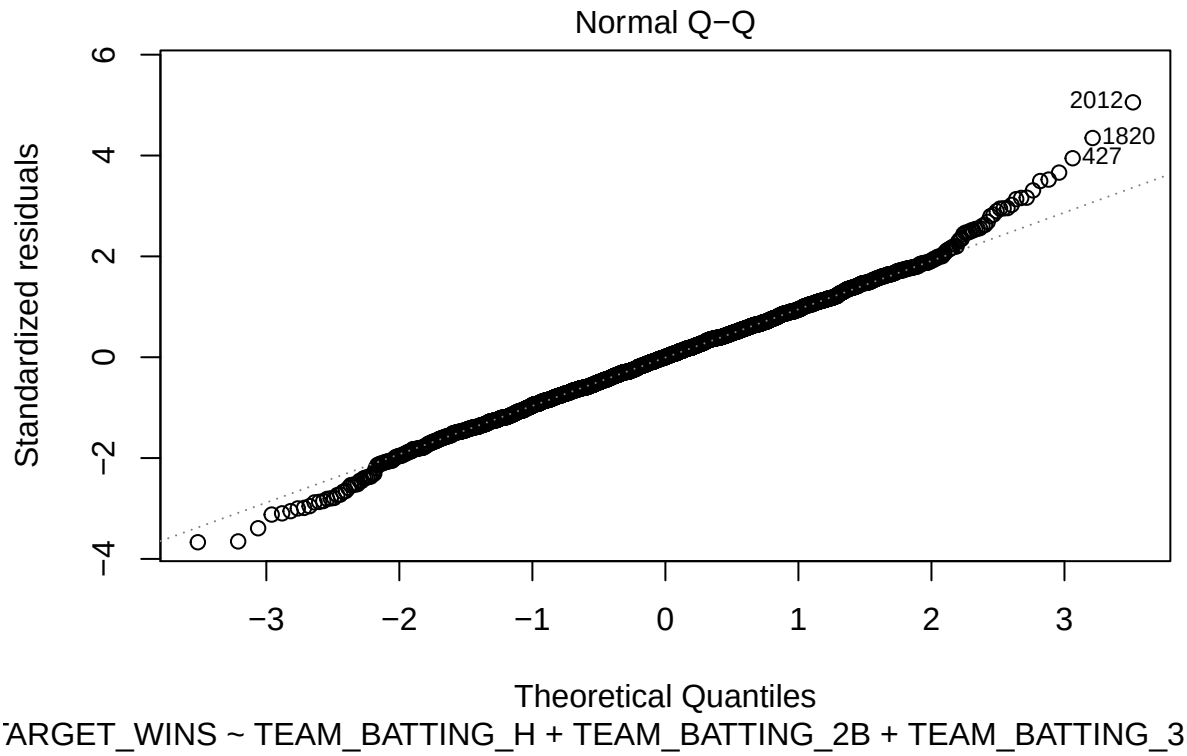
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
##     TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_E +
##     TEAM_FIELDING_DP, data = moneyball_filled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.117  -8.396   0.026   8.238  64.496
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -18.936246  14.300248  -1.324 0.185574
## TEAM_BATTING_H     0.039743   0.003901  10.187  < 2e-16 ***
## TEAM_BATTING_2B   -0.021808   0.009043  -2.412 0.015957 *
## TEAM_BATTING_3B    0.101540   0.016844   6.028 1.93e-09 ***
## TEAM_BATTING_HR    0.065471   0.009640   6.791 1.41e-11 ***
## TEAM_BATTING_BB    0.012322   0.003354   3.674 0.000245 ***
## TEAM_BATTING_SO   -0.001465   0.002507  -0.584 0.559106
## TEAM_BASERUN_SB    0.032302   0.004275   7.556 5.99e-14 ***
## TEAM_PITCHING_H   18.926032   4.961698   3.814 0.000140 ***
## TEAM_PITCHING_SO  -3.560746   0.932952  -3.817 0.000139 ***
## TEAM_FIELDING_E   -0.031367   0.003048 -10.291  < 2e-16 ***
## TEAM_FIELDING_DP  -0.120207   0.012872  -9.339  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.89 on 2261 degrees of freedom
## Multiple R-squared:  0.3245, Adjusted R-squared:  0.3212
## F-statistic: 98.72 on 11 and 2261 DF,  p-value: < 2.2e-16
```

## Checking for Colinearity

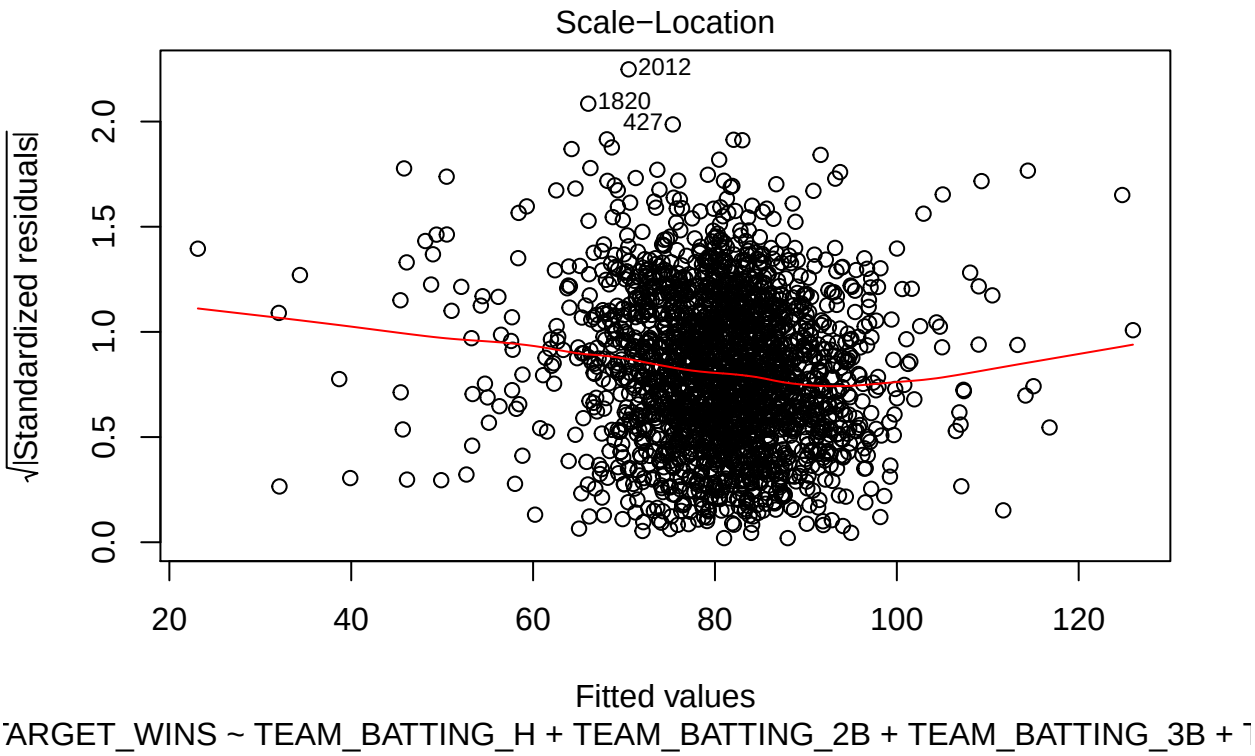
```
##   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B  TEAM_BATTING_HR
##         4.322088         2.438948         3.007291         4.650721
##  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_H
##         2.294133         5.045176         1.812464         5.617980
## TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##         1.722976         6.490171         1.364366
```

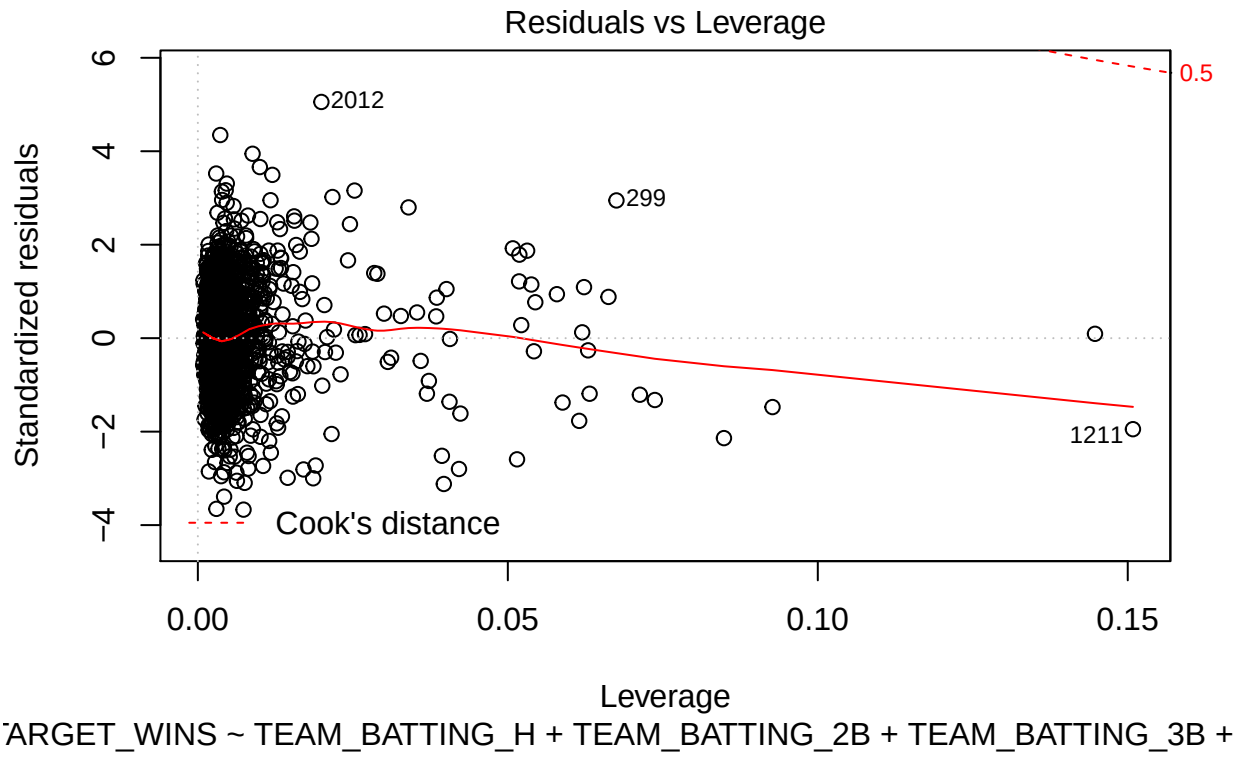



TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + T




TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + T


Scale−Location
2012
1820
427
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
20
40
60
80
100
120
Fitted values
TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + T




TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + T

## Model 5

TEAM_FIELDING_E is withing the range 5-10 (suggesting co linearity with other variables), but eliminating TEAM_FIELDING_E does not improve the model. This yields our highest $R^2$ value with 40% of the variance explained by our model.

```
##
## 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_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_E +
##     TEAM_FIELDING_DP, data = moneyball_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.317  -7.199   0.121   7.045  29.766
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      58.312951   6.019406   9.687  < 2e-16 ***
## TEAM_BATTING_H   -0.010007   0.010615  -0.943  0.34594
## TEAM_BATTING_2B  -0.049989   0.008875  -5.633 2.05e-08 ***
## TEAM_BATTING_3B   0.181788   0.018982   9.577  < 2e-16 ***
## TEAM_BATTING_HR   0.100845   0.009158  11.012  < 2e-16 ***
## TEAM_BATTING_BB   0.034055   0.003133  10.870  < 2e-16 ***
```

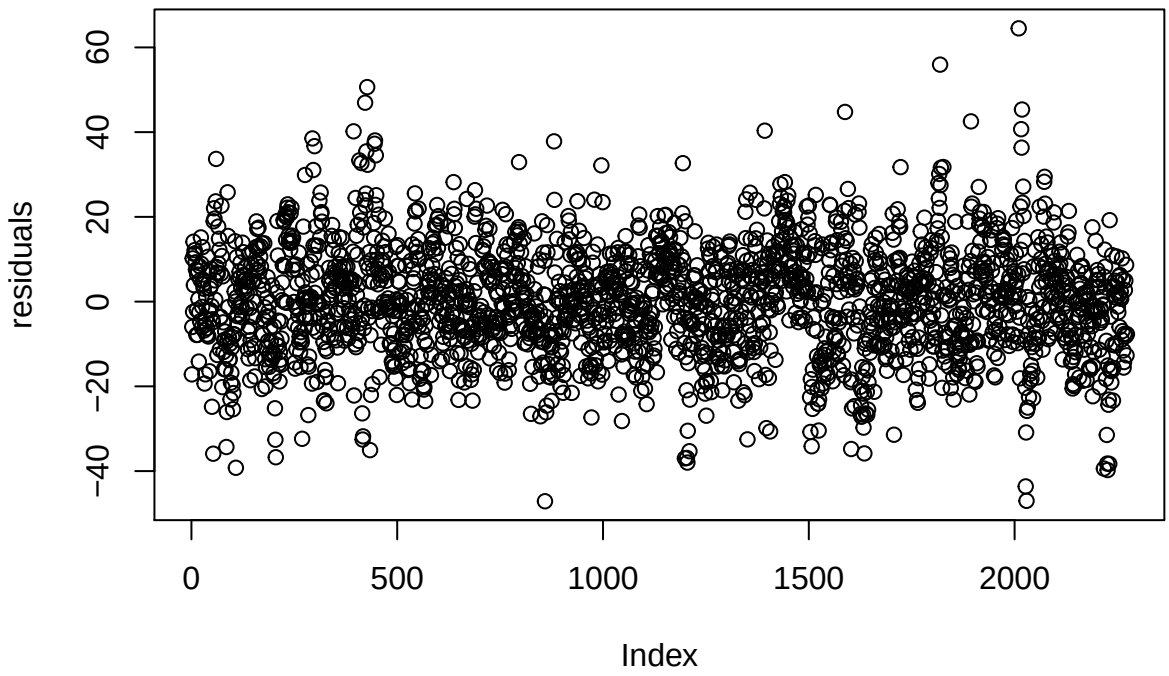
```
## TEAM_BATTING_SO   0.045928   0.016420   2.797  0.00521 **
## TEAM_BASERUN_SB   0.069889   0.005535  12.626  < 2e-16 ***
## TEAM_PITCHING_H   0.037438   0.009239   4.052 5.29e-05 ***
## TEAM_PITCHING_SO -0.065427   0.015514  -4.217 2.59e-05 ***
## TEAM_FIELDING_E  -0.116444   0.007029 -16.566  < 2e-16 ***
## TEAM_FIELDING_DP -0.112850   0.012279  -9.190  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.19 on 1823 degrees of freedom
##   (441 observations deleted due to missingness)
## Multiple R-squared:  0.4045, Adjusted R-squared:  0.4009
## F-statistic: 112.6 on 11 and 1823 DF,  p-value: < 2.2e-16
```

# Part 4: Model Evaluation

Using our finished model above, we can predict the number of wins for each team. We rounded to a whole number so that the finished values have some real world analogue. The F-statistics has a p value of basically 0, so we can determine that our model is statistically significant.

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  65  66  74  86  NA  NA  NA  76  72  73  69  82  82  81  84  76  75  79
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  NA  90  82  83  82  71  81  86  NA  74  84  74  90  85  82  84  81  86
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  76  91  86  92  81  90  NA  NA  NA  NA  NA  78  70  80  76  83  79  74
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  75  79  94  76  NA  NA  88  75  88  85  83  NA  77  82  NA  91  88  69
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  77  89  81  86  83  83  NA  NA  83  89  97  74  86  78  82  83  87  90
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##  NA  NA  75  NA  NA  NA  90 104  88  88  80  73  83  84  80  NA  NA  78
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##  87  NA  84  84  93  91  81  78  86  81  75  NA  NA  NA  NA  NA  71  88
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##  92  78  93  92  86  78  79  87  88  NA  75  77  86  80  67  NA  91  74
## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
##  72  72  78  78  78  82  82  81  NA  70  77  71  90  NA  96  NA 106 107
## 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##  95 104  98  90  82  80  73  79  NA  90  81  93  83  74  77  71  74  79
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##  86  89  85  85  NA  NA  NA  NA  NA  NA  NA  NA  76  76  79  67  78  85
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
##  80  86  78  81  76  90  80  84  79  78  NA  NA  NA  NA  85  65  69  84
## 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
##  80  93  77  80  79  75  81  74  NA  76  81  81  82  NA  NA  93  79  88
## 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
##  80  76  83  78  NA  74  90  86  82  81  61  85  80  85  72  84  82  NA
## 253 254 255 256 257 258 259
##  NA  NA  69  76  83  82  78
```

Additionally, we can use a residual plot to verify our model. We can see that our model's residuals are fairly normal and randomly distributed. They also are centered and zero.




```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -47.11665  -8.39565   0.02552   0.00000   8.23798  64.49592
```