# DATA 621 Assignment 1

*Kai Lukowiak*

*2018-02-08*

## 1. Data Exploration:

Describe the size and the variables in the moneyball training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job. Some suggestions are given below. Please do NOT treat this as a check list of things to do to complete the assignment. You should have your own thoughts on what to tell the boss. These are just ideas.

a. Mean / Standard Deviation / Median
b. Bar Chart or Box Plot of the data
c. Is the data correlated to the target variable (or to other variables?)
d. Are any of the variables missing and need to be imputed "fixed"?

### Loading the data:

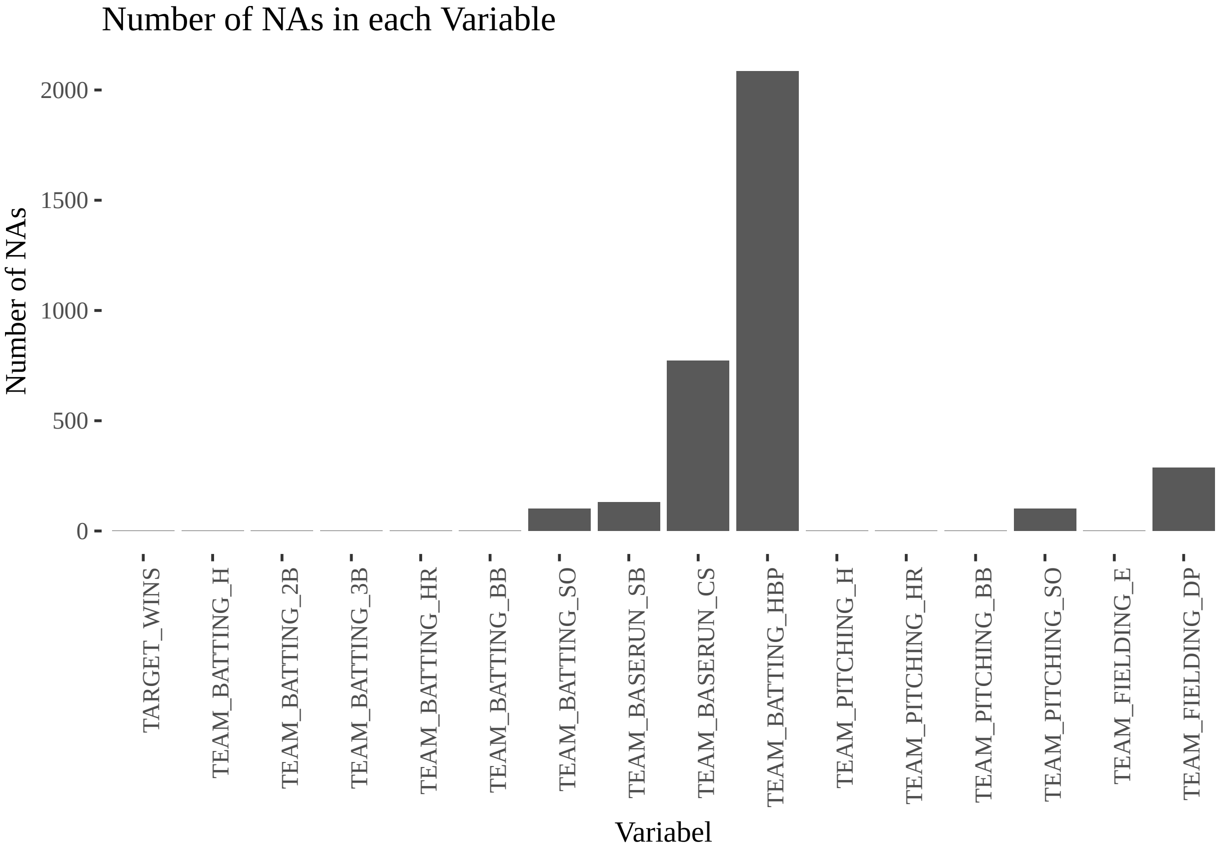
The data looks like this. We can see that all observations are integers and that their scale varies wildly.

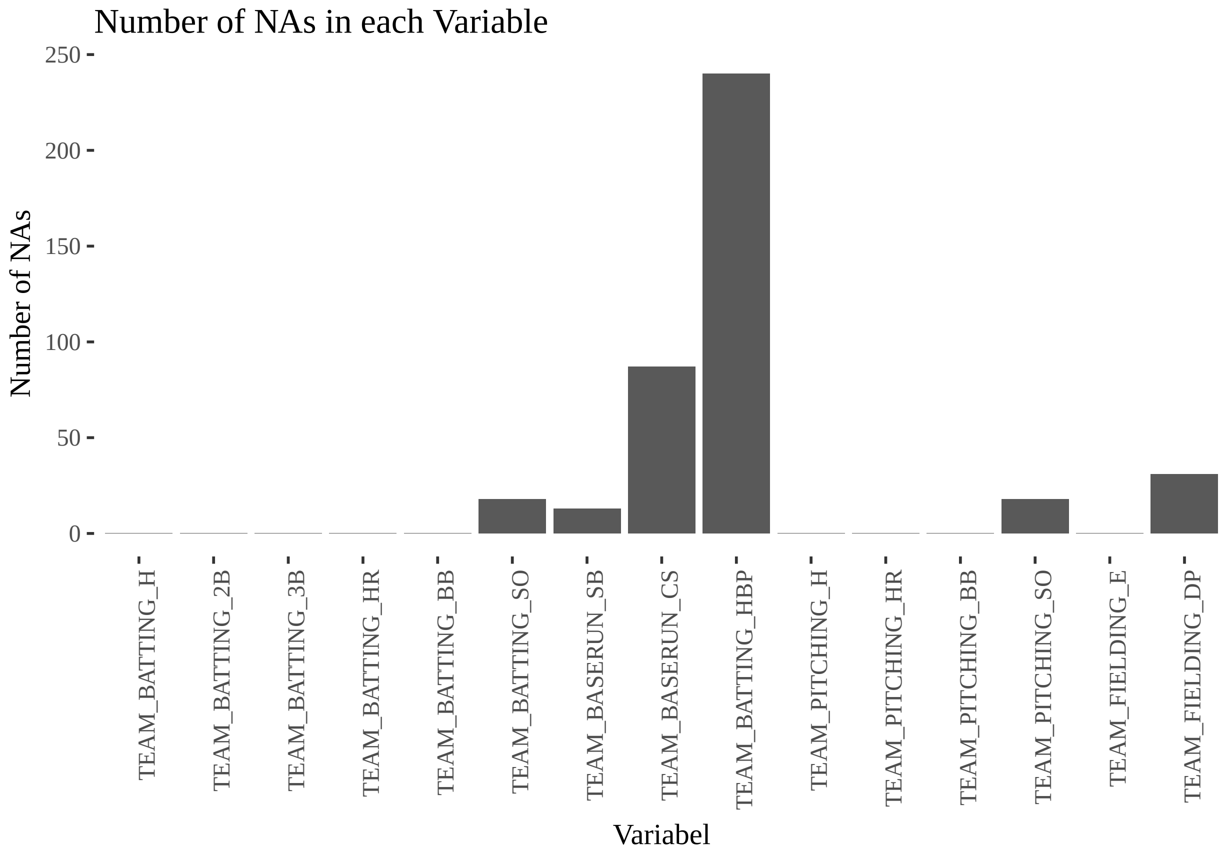
```
## Observations: 2,276
## Variables: 16
## $ TARGET_WINS      <int> 39, 70, 86, 70, 82, 75, 80, 85, 86, 76, 78, 6...
## $ TEAM_BATTING_H   <int> 1445, 1339, 1377, 1387, 1297, 1279, 1244, 127...
## $ TEAM_BATTING_2B  <int> 194, 219, 232, 209, 186, 200, 179, 171, 197, ...
## $ TEAM_BATTING_3B  <int> 39, 22, 35, 38, 27, 36, 54, 37, 40, 18, 27, 3...
## $ 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, ...
## $ TEAM_BATTING_SO  <int> 842, 1075, 917, 922, 920, 973, 1062, 1027, 92...
## $ TEAM_BASERUN_SB  <int> NA, 37, 46, 43, 49, 107, 80, 40, 69, 72, 60, ...
## $ TEAM_BASERUN_CS  <int> NA, 28, 27, 30, 39, 59, 54, 36, 27, 34, 39, 7...
## $ TEAM_BATTING_HBP <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ TEAM_PITCHING_H  <int> 9364, 1347, 1377, 1396, 1297, 1279, 1244, 128...
## $ 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, ...
## $ TEAM_PITCHING_SO <int> 5456, 1082, 917, 928, 920, 973, 1062, 1033, 9...
## $ TEAM_FIELDING_E  <int> 1011, 193, 175, 164, 138, 123, 136, 112, 127,...
## $ TEAM_FIELDING_DP <int> NA, 155, 153, 156, 168, 149, 186, 136, 169, 1...
```

```
## Observations: 259
## Variables: 15
## $ TEAM_BATTING_H   <int> 1209, 1221, 1395, 1539, 1445, 1431, 1430, 138...
## $ TEAM_BATTING_2B  <int> 170, 151, 183, 309, 203, 236, 219, 158, 177, ...
## $ TEAM_BATTING_3B  <int> 33, 29, 29, 29, 68, 53, 55, 42, 78, 42, 40, 5...
## $ TEAM_BATTING_HR  <int> 83, 88, 93, 159, 5, 10, 37, 33, 23, 58, 50, 1...
## $ TEAM_BATTING_BB  <int> 447, 516, 509, 486, 95, 215, 568, 356, 466, 4...
## $ TEAM_BATTING_SO  <int> 1080, 929, 816, 914, 416, 377, 527, 609, 689,...
## $ TEAM_BASERUN_SB  <int> 62, 54, 59, 148, NA, NA, 365, 185, 150, 52, 6...
## $ TEAM_BASERUN_CS  <int> 50, 39, 47, 57, NA, NA, NA, NA, NA, NA, NA, 2...
## $ TEAM_BATTING_HBP <int> NA, NA, NA, 42, NA, NA, NA, NA, NA, NA, NA, N...
## $ TEAM_PITCHING_H  <int> 1209, 1221, 1395, 1539, 3902, 2793, 1544, 162...
## $ TEAM_PITCHING_HR <int> 83, 88, 93, 159, 14, 20, 40, 39, 25, 62, 53, ...
## $ TEAM_PITCHING_BB <int> 447, 516, 509, 486, 257, 420, 613, 418, 497, ...
## $ TEAM_PITCHING_SO <int> 1080, 929, 816, 914, 1123, 736, 569, 715, 734...
## $ TEAM_FIELDING_E  <int> 140, 135, 156, 124, 616, 572, 490, 328, 226, ...
## $ TEAM_FIELDING_DP <int> 156, 164, 153, 154, 130, 105, NA, 104, 132, 1...
```

### Initial Vizualizaton:

The variable `TEAM_BATTING_HPB` is almost completly filled with `NA` values. This, along with the additional information that hits by ball do not occure as much as they used too means we should probably delete it.

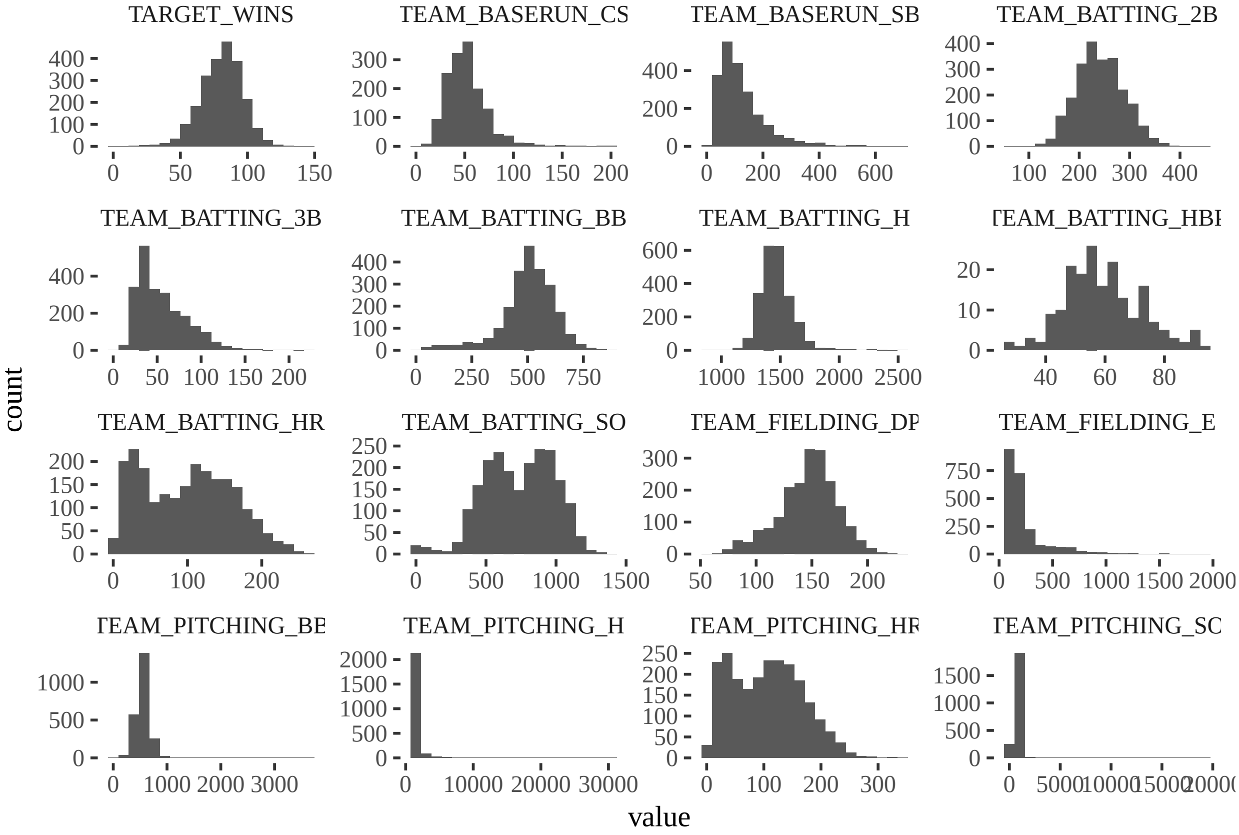




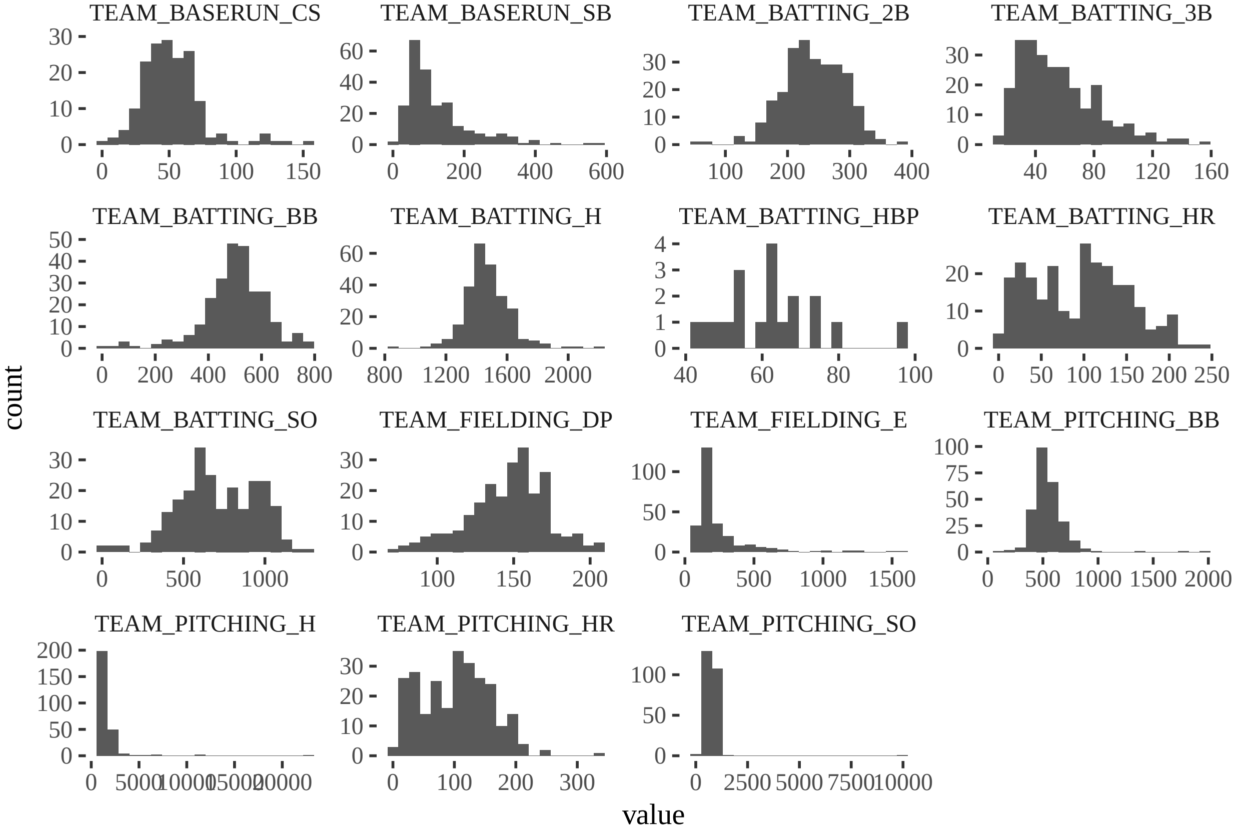
Special thanks to this Stack Overflow question

Next is a visualization of the histograms check this distrubution of the data. While there are some variables that are skewness might be an issue, we really must check for normallacy of the erros later on. All in all, the data looks good.

```
## Warning: Removed 3478 rows containing non-finite values (stat_bin).
```

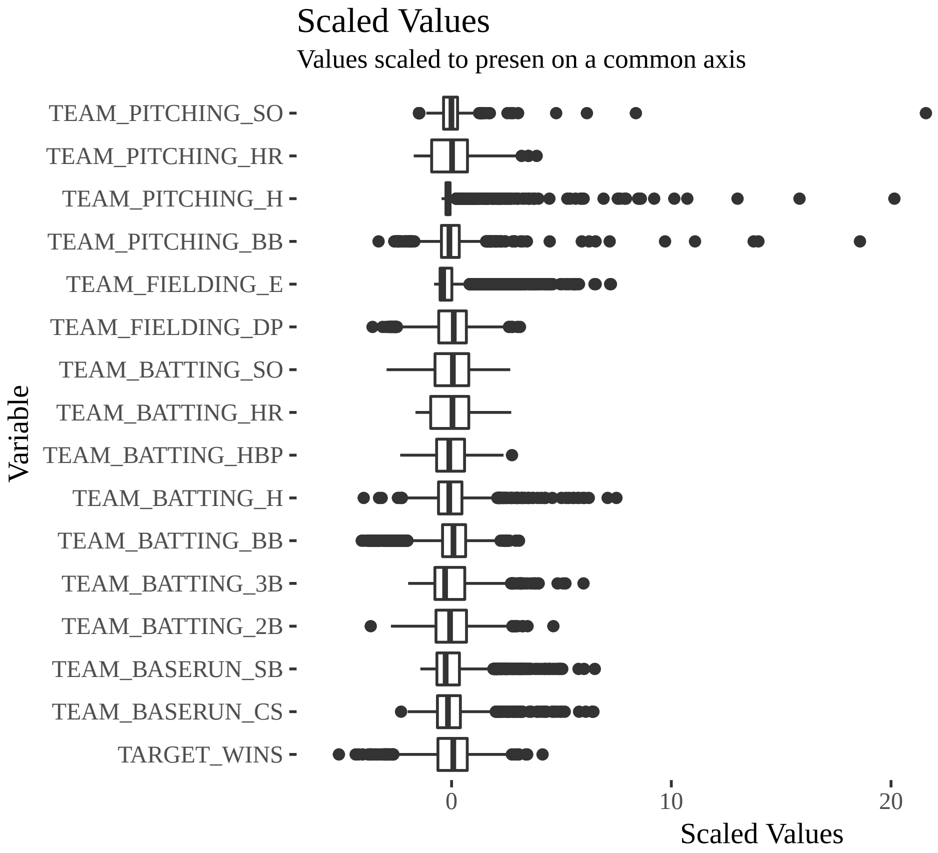


```
## Warning: Removed 407 rows containing non-finite values (stat_bin).
```



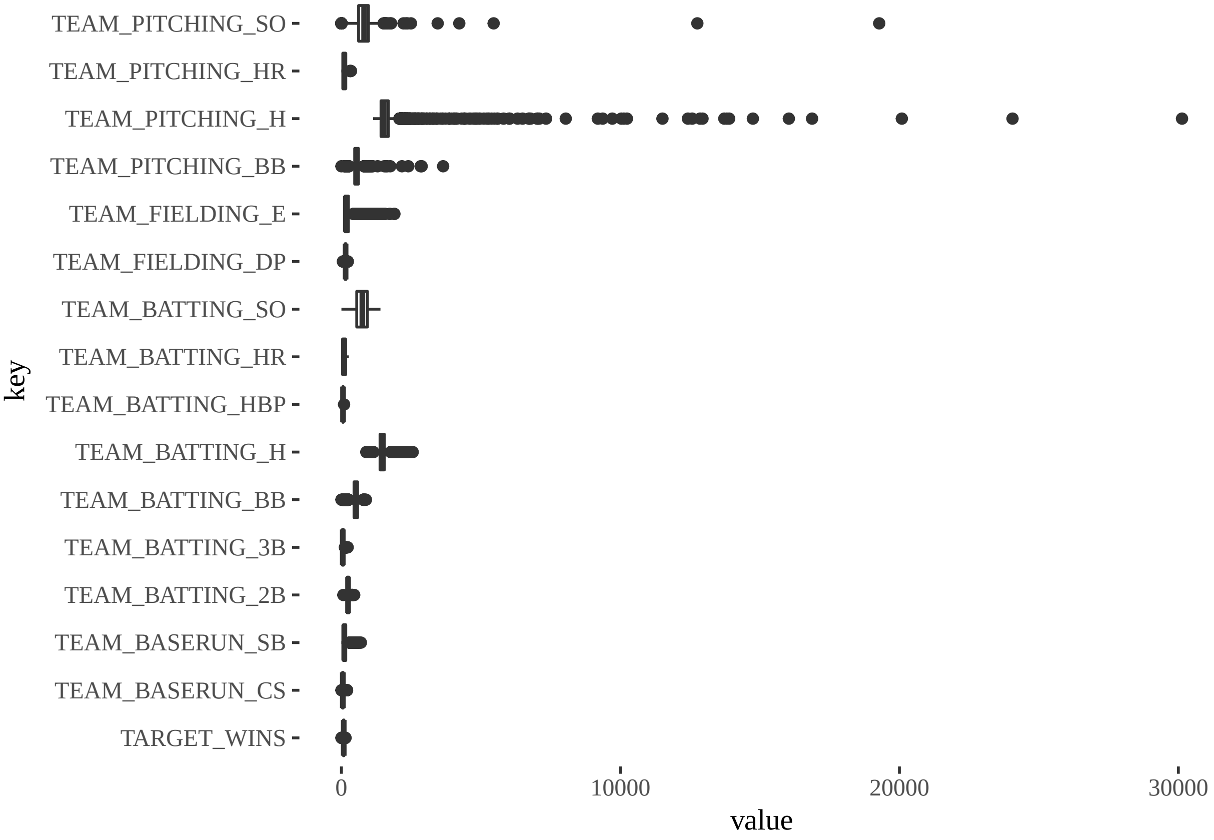
From these plots we can see that many variables are aproximetly normally distrubuted. Notable exceptions are `TEAM_BATTING_3B`, `TEAM_BATTING_HR`, `TEAM_PITCHING_H`.

## Bar plots.



While there are significant outliers,

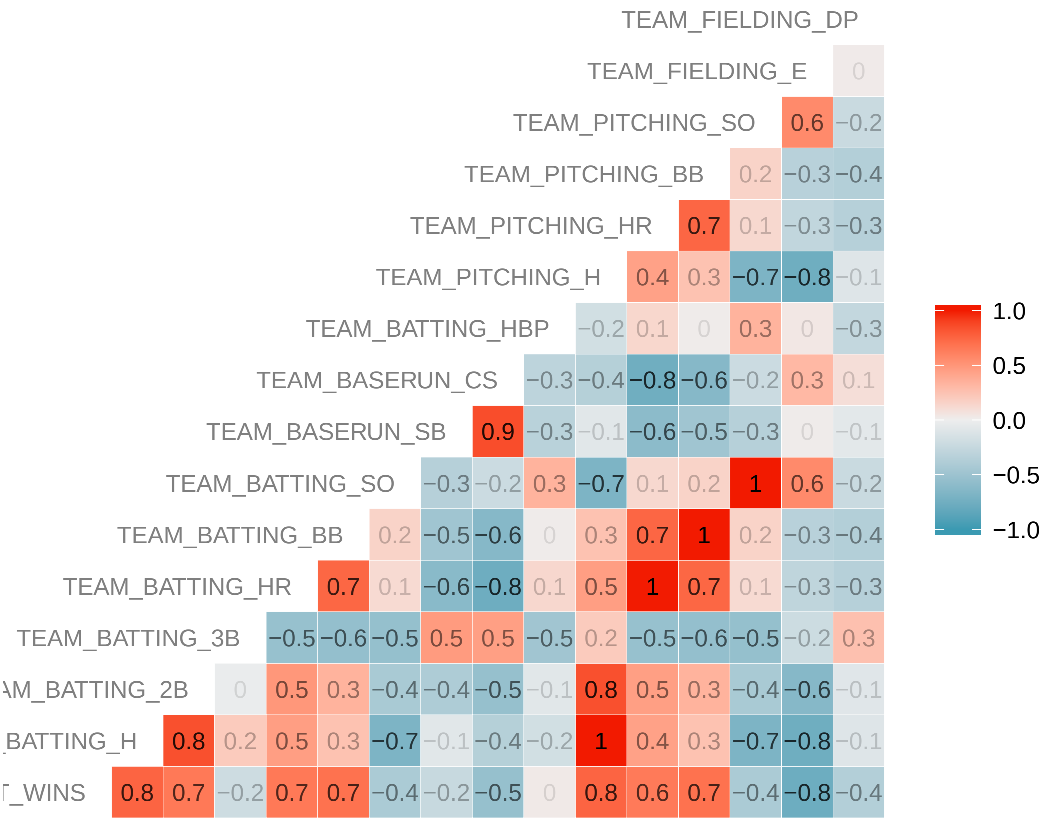
```
## Warning: Removed 3478 rows containing non-finite values (stat_boxplot).
```



While some values are extream outliers, we shoulnd't be too worried because compared to the total number of observations they are small.

## Heat Maps

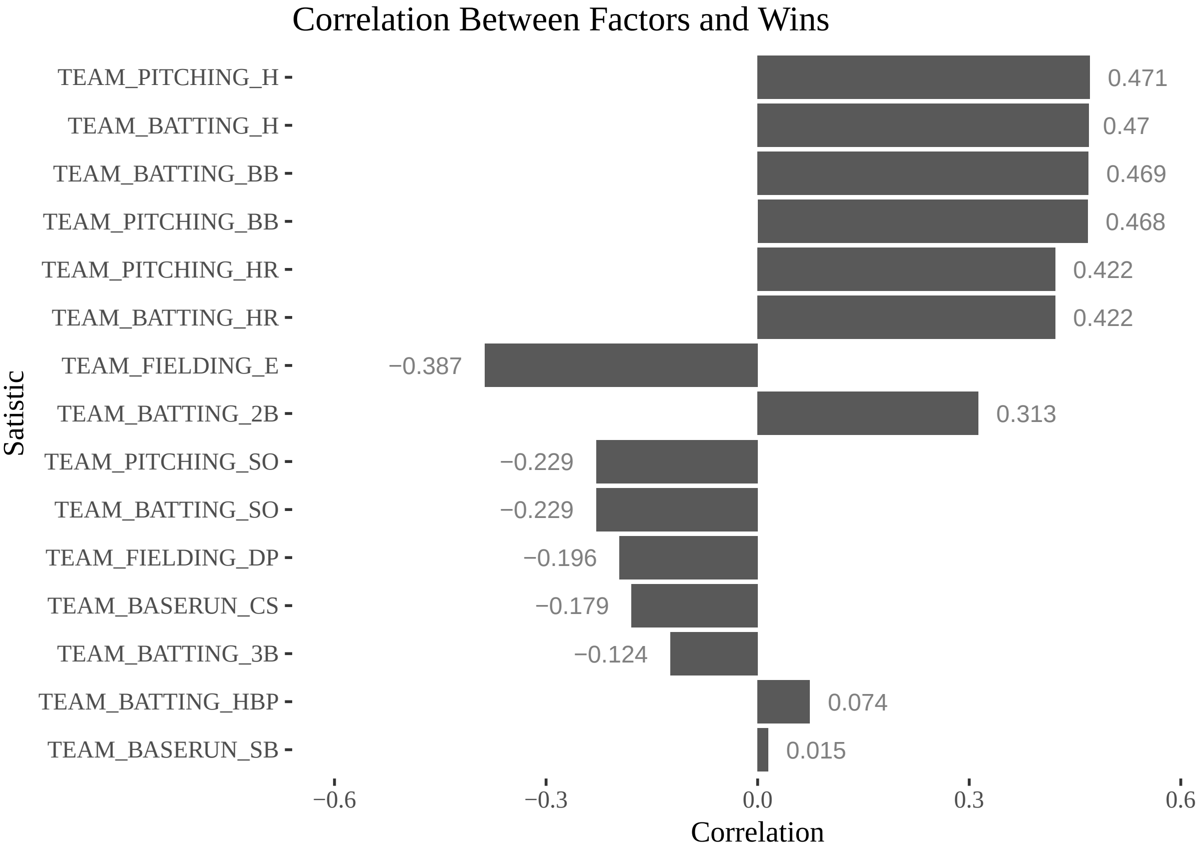
Special thanks to this site We can see that there are some variables completely correlated variables. These mean that They can be excluded from the regression.



```
##   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
```

## Correlation

Hwere we see that the variables most correlated `WINS` are `TEAM_BATTING_H` and `TEAM_PITCHING_H` since these are correlated (from above) we can count them as the same.



We also should check if there is correlation between the rows that had tons of NA values.

```
## [1] -0.002610647
```

The correlation with missing variables for `TEAM_BATTING_HPB` is virtually 0. Thus, we can ignore it and remove it from our model.

```
## [1] -0.124493
```

```
## [1] 0.450468
```

```
## [1] -0.1245477
```

We see that the corelation is not significant (p = -0.1245477) and futher, even if it was significant, the effect is so small it might be worth deleting the column instead.

# 2. DATA PREPARATION

Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this. Here are some possible transformations. a. Fix missing values (maybe with a Mean or Median value) b. Create flags to suggest if a variable was missing c. Transform data by putting it into buckets d. Mathematical transforms such as log or square root (or use Box-Cox) e. Combine variables (such as ratios or adding or multiplying) to create new variables

Given our data analysis in the previous section I feel comfortable removing the variable TEAM_BATTING_HBP.

```
dfT <- dfT %>% select(-TEAM_BATTING_HBP)
dfE <- dfE %>% select(-TEAM_BATTING_HBP)
```

There were significant numbers of NA values in other variables. We will try to impute them.

```
## Warning: package 'mice' was built under R version 3.4.2
```

# 3. BUILD MODELS

Using the training data set, build at least three different multiple linear regression models, using different variables (or the same variables with different transformations). Since we have not yet covered automated variable selection methods, you should select the variables manually (unless you previously learned Forward or Stepwise selection, etc.). Since you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done. Discuss the coefficients in the models, do they make sense? For example, if a team hits a lot of Home Runs, it would be reasonably expected that such a team would win more games. However, if the coefficient is negative (suggesting that the team would lose more games), then that needs to be discussed. Are you keeping the model even though it is counter intuitive? Why? The boss needs to know.

First we will try to fit the model with all available data. This gives the output:

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = dfT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.5627  -6.6932  -0.1328   6.5249  27.8525
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      57.912438   6.642839   8.718  < 2e-16 ***
## TEAM_BATTING_H    0.015434   0.019626   0.786   0.4318
## TEAM_BATTING_2B  -0.070472   0.009369  -7.522 9.36e-14 ***
## TEAM_BATTING_3B   0.161551   0.022192   7.280 5.43e-13 ***
## TEAM_BATTING_HR   0.073952   0.085392   0.866   0.3866
## TEAM_BATTING_BB   0.043765   0.046454   0.942   0.3463
## TEAM_BATTING_SO   0.018250   0.023463   0.778   0.4368
## TEAM_BASERUN_SB   0.035880   0.008687   4.130 3.83e-05 ***
## TEAM_BASERUN_CS   0.052124   0.018227   2.860   0.0043 **
## TEAM_PITCHING_H   0.019044   0.018381   1.036   0.3003
## TEAM_PITCHING_HR  0.022997   0.082092   0.280   0.7794
## TEAM_PITCHING_BB -0.004180   0.044692  -0.094   0.9255
## TEAM_PITCHING_SO -0.038176   0.022447  -1.701   0.0892 .
## TEAM_FIELDING_E  -0.155876   0.009946 -15.672  < 2e-16 ***
## TEAM_FIELDING_DP -0.112885   0.013137  -8.593  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.556 on 1471 degrees of freedom
##   (790 observations deleted due to missingness)
## Multiple R-squared:  0.4386, Adjusted R-squared:  0.4333
## F-statistic:  82.1 on 14 and 1471 DF,  p-value: < 2.2e-16
```

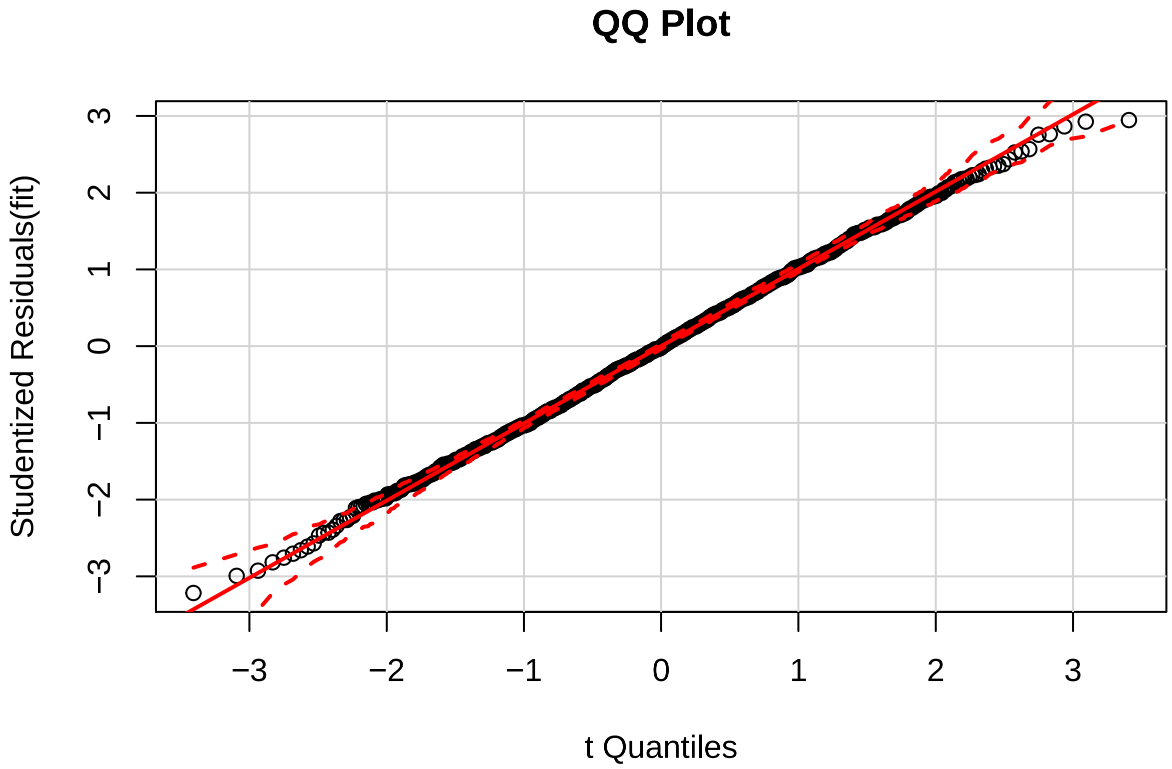
```
## Warning: package 'car' was built under R version 3.4.3
```

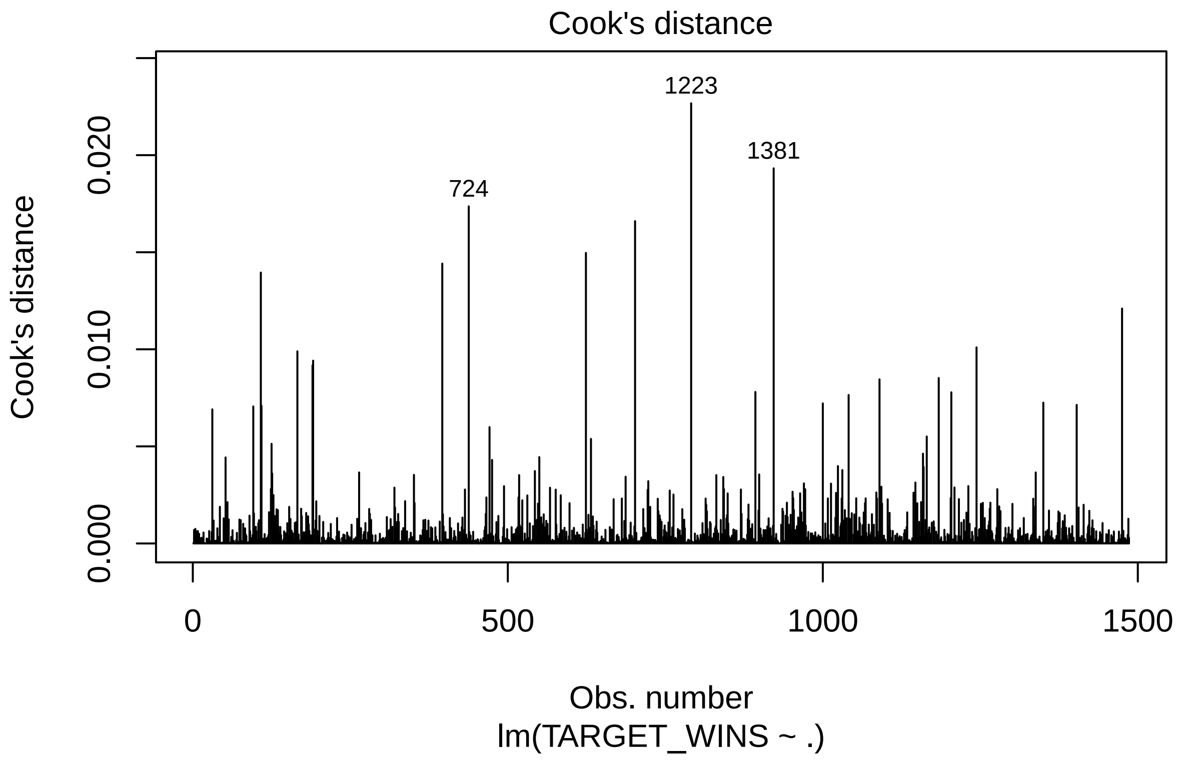
```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferonni p
## 205 -3.216895          0.0013241           NA
```

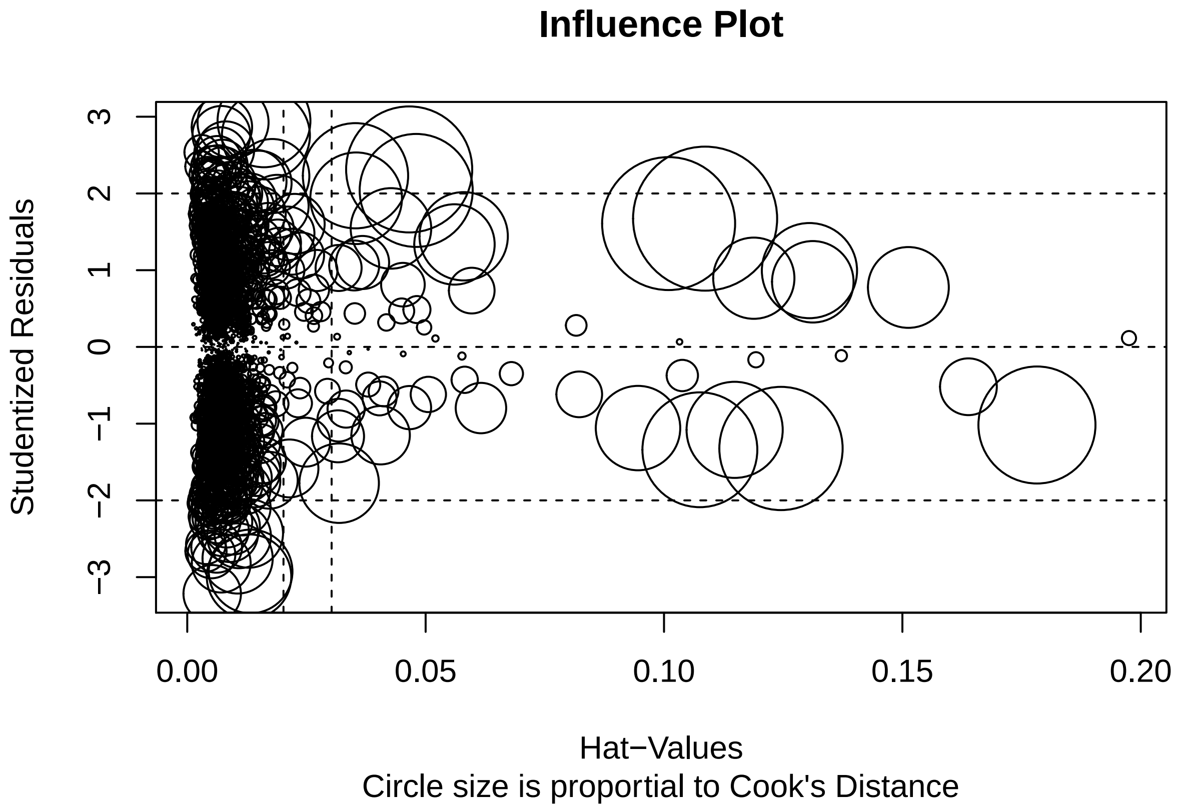


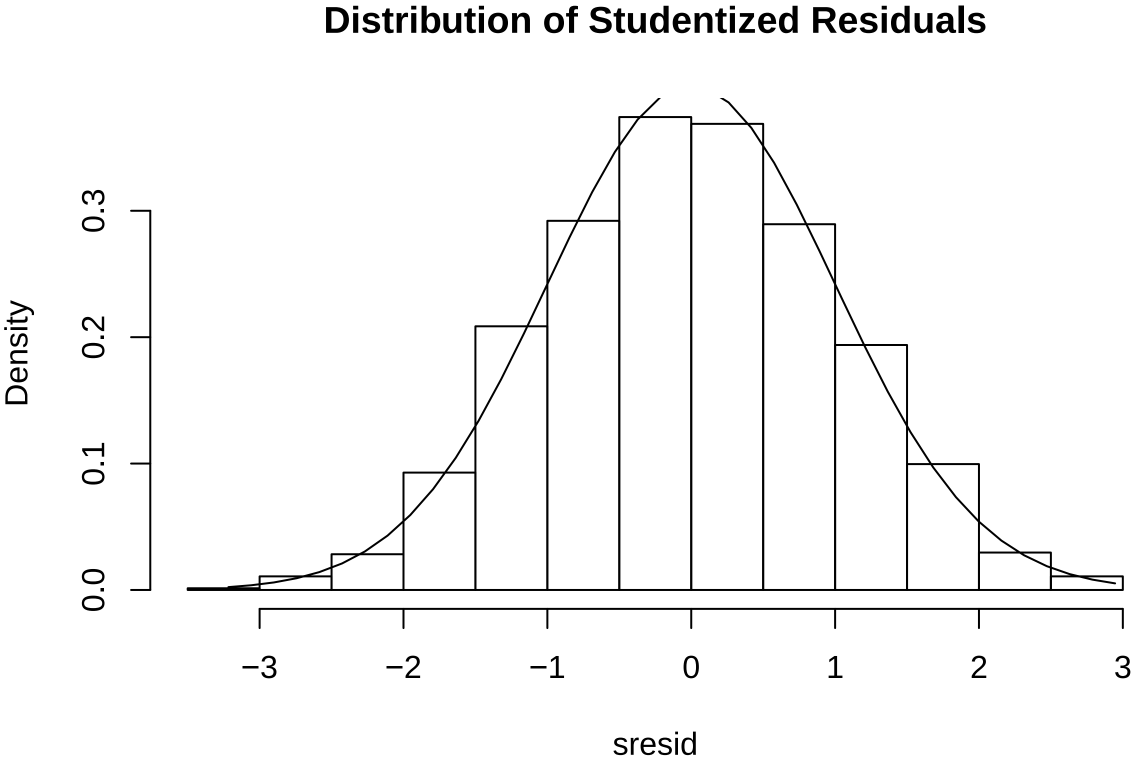


```
## Warning: package 'MASS' was built under R version 3.4.3
```

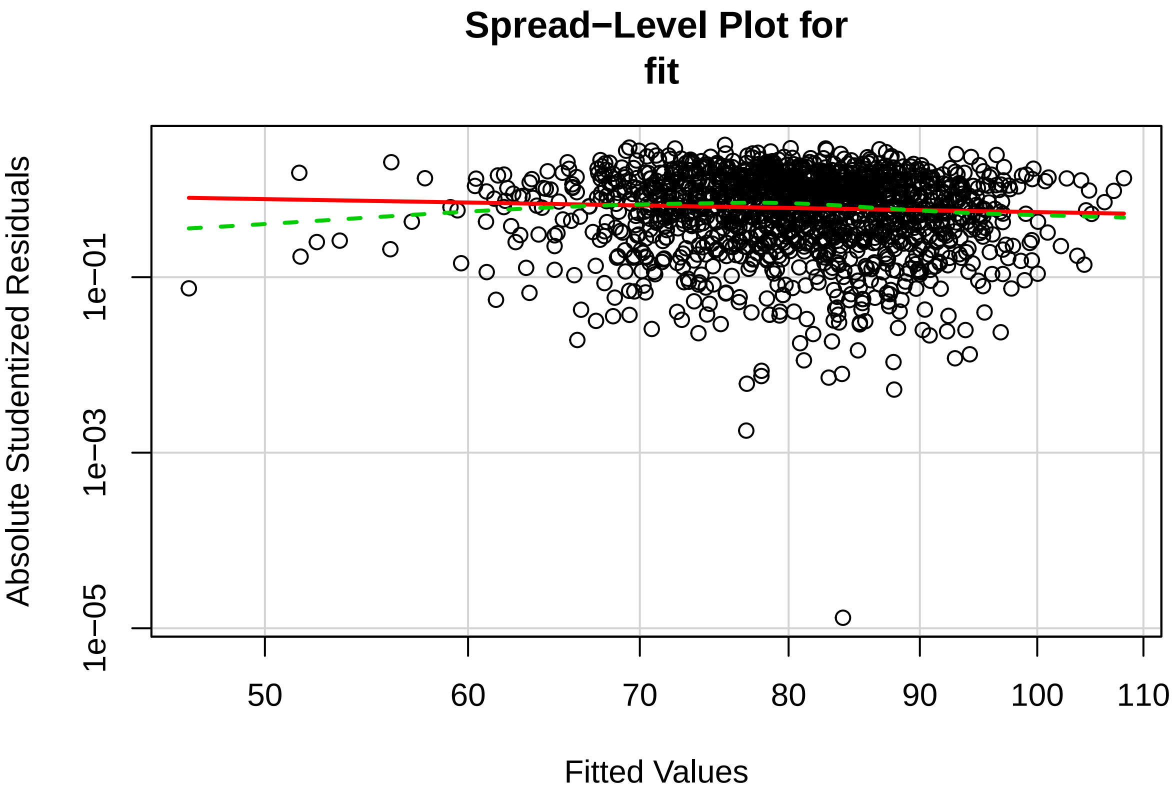
```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```



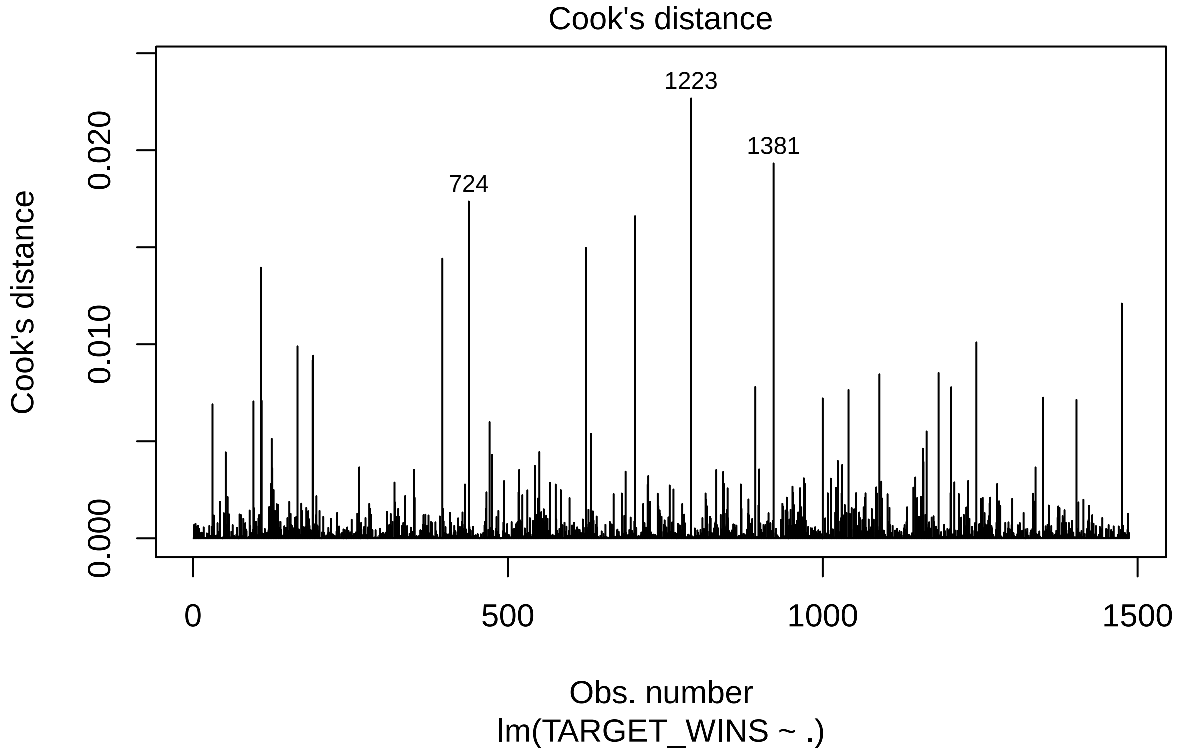


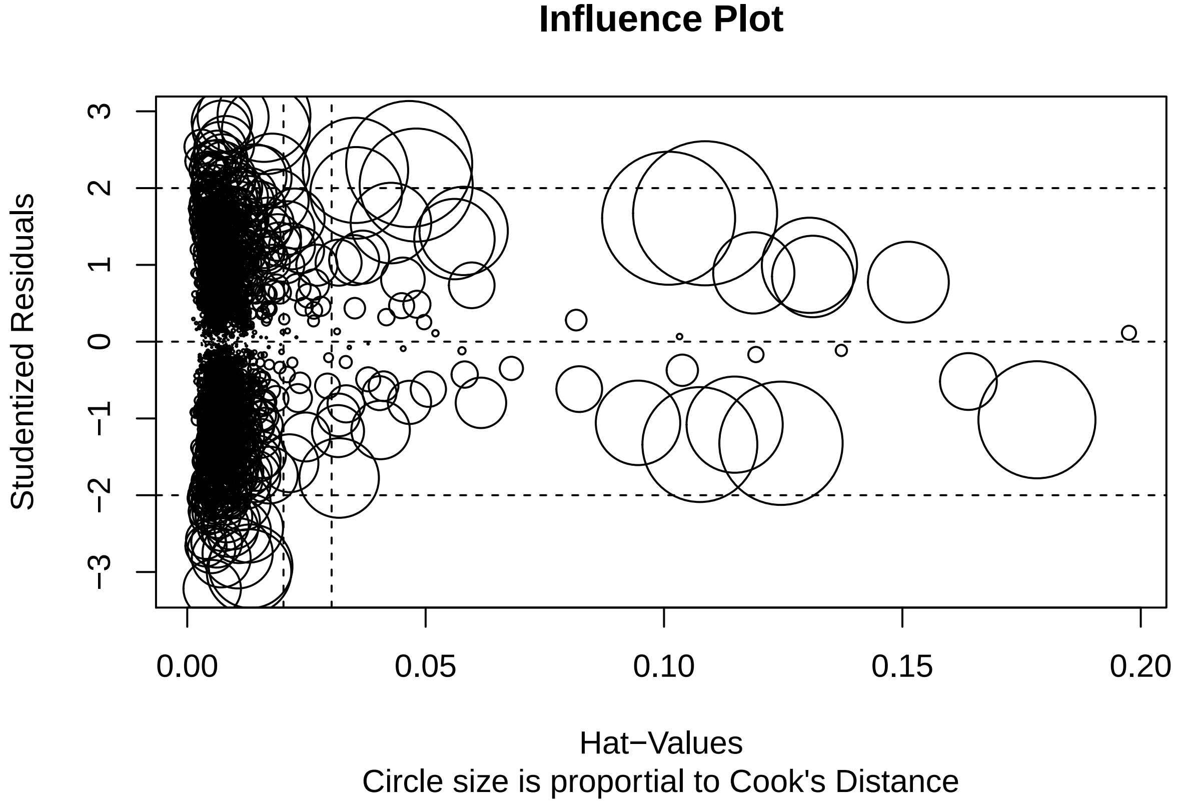
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 8.37572    Df = 1     p = 0.003802668
```



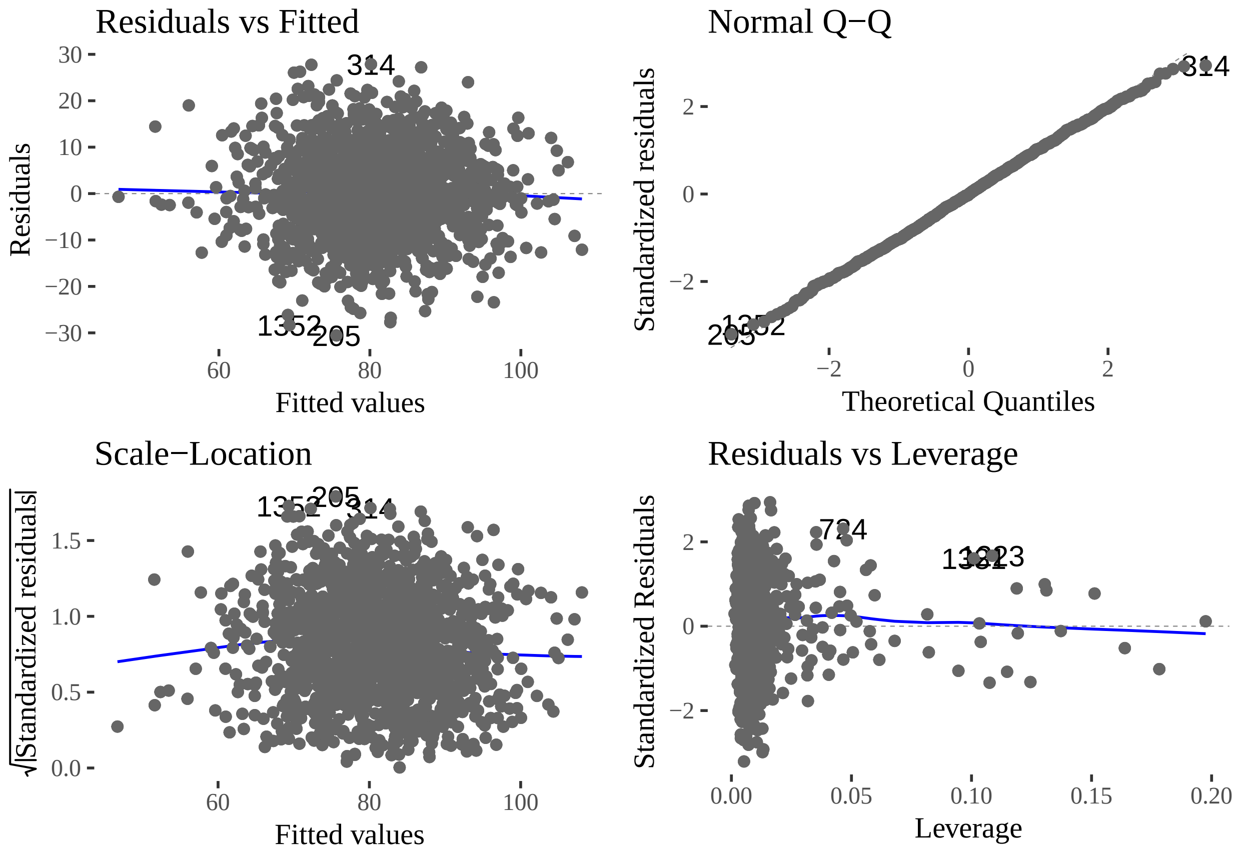
```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.3955704      1.208475       0
##  Alternative hypothesis: rho != 0
```

Running the diagnostic on this regression we can see that it looks pretty good. The QQ plot especially makes it look like the regression does not suffer from major issues.





```
## Warning: package 'ggfortify' was built under R version 3.4.3
```



These aditional graphs, especially the residuals, do not imply bias. I also don't know enought about baseball to feel justified removing the datapoints with high cooks distance.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B +
##     TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_FIELDING_E + TEAM_FIELDING_DP,
##     data = dfT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.687  -7.955  -0.154   8.008  37.873
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      92.152078   3.534132  26.075  < 2e-16 ***
## TEAM_BATTING_2B   0.024128   0.007505   3.215  0.00133 **
## TEAM_BATTING_3B   0.269509   0.021720  12.408  < 2e-16 ***
## TEAM_BASERUN_SB   0.019670   0.010031   1.961  0.05008 .
## TEAM_BASERUN_CS   0.002149   0.021423   0.100  0.92012
## TEAM_FIELDING_E  -0.162860   0.011250 -14.477  < 2e-16 ***
## TEAM_FIELDING_DP -0.048528   0.015288  -3.174  0.00153 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.47 on 1479 degrees of freedom
##   (790 observations deleted due to missingness)
## Multiple R-squared:  0.1875, Adjusted R-squared:  0.1842
## F-statistic: 56.89 on 6 and 1479 DF,  p-value: < 2.2e-16
```

There is significant reduction in the $R^2$ if we we only control for the significant values.

```
## Analysis of Variance Table
##
## Model 1: TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BASERUN_SB +
##     TEAM_BASERUN_CS + TEAM_FIELDING_E + TEAM_FIELDING_DP
## Model 2: 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
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1479 194409
## 2   1471 134323  8     60086 82.252 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we remove the completly co-linear variables we get an even better R-squared.

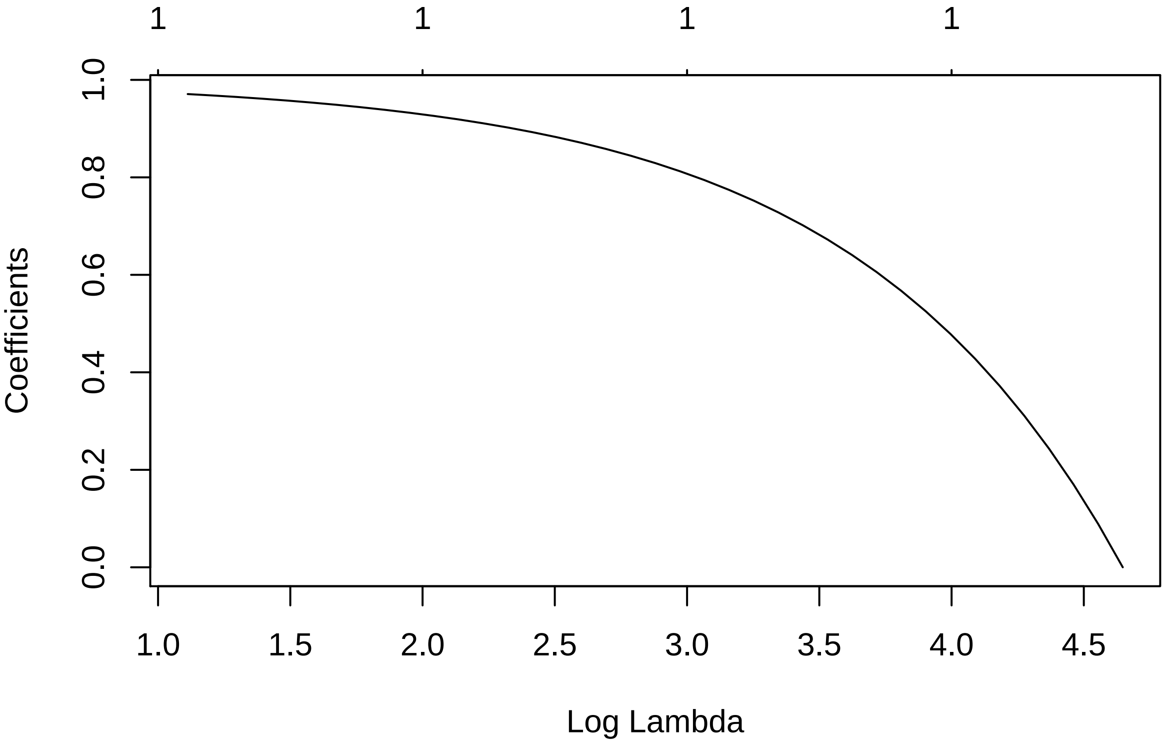
```
##
## Call:
## lm(formula = TARGET_WINS ~ . - TEAM_BATTING_SO - TEAM_BATTING_BB -
##     TEAM_BATTING_HR - TEAM_BATTING_H, data = dfT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.005  -6.608   0.233   6.612  31.644
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.050e+02  3.925e+00  26.740  < 2e-16 ***
## TEAM_BATTING_2B  -1.821e-02  7.399e-03  -2.461   0.0140 *
## TEAM_BATTING_3B   1.998e-01  2.221e-02   8.996  < 2e-16 ***
## TEAM_BASERUN_SB   4.830e-02  8.760e-03   5.514 4.14e-08 ***
## TEAM_BASERUN_CS   3.942e-02  1.862e-02   2.118   0.0344 *
## TEAM_PITCHING_H  -5.787e-04  2.159e-03  -0.268   0.7887
## TEAM_PITCHING_HR  1.273e-01  8.322e-03  15.301  < 2e-16 ***
## TEAM_PITCHING_BB  2.962e-02  3.169e-03   9.347  < 2e-16 ***
## TEAM_PITCHING_SO -3.270e-02  1.921e-03 -17.023  < 2e-16 ***
## TEAM_FIELDING_E  -1.587e-01  1.017e-02 -15.605  < 2e-16 ***
## TEAM_FIELDING_DP -1.065e-01  1.344e-02  -7.927 4.41e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.795 on 1475 degrees of freedom
##   (790 observations deleted due to missingness)
## Multiple R-squared:  0.4085, Adjusted R-squared:  0.4045
## F-statistic: 101.9 on 10 and 1475 DF,  p-value: < 2.2e-16
```

There is also significant difference between the models. Since the All model performs better WRT the $R^2$ it should be pefered.

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = complete(tempData, 1))
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.848  -8.309   0.246   8.251  51.675
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      35.6354861  5.1859569   6.872 8.18e-12 ***
## TEAM_BATTING_H    0.0413444  0.0035570  11.624  < 2e-16 ***
## TEAM_BATTING_2B  -0.0183100  0.0088341  -2.073  0.03832 *
## TEAM_BATTING_3B   0.0358669  0.0162591   2.206  0.02749 *
## TEAM_BATTING_HR   0.0587626  0.0262571   2.238  0.02532 *
## TEAM_BATTING_BB   0.0160631  0.0055989   2.869  0.00416 **
## TEAM_BATTING_SO  -0.0160197  0.0024420  -6.560 6.64e-11 ***
## TEAM_BASERUN_SB   0.0516263  0.0050743  10.174  < 2e-16 ***
## TEAM_BASERUN_CS   0.0103483  0.0103325   1.002  0.31668
## TEAM_PITCHING_H   0.0015642  0.0003795   4.121 3.91e-05 ***
## TEAM_PITCHING_HR  0.0263168  0.0233220   1.128  0.25927
## TEAM_PITCHING_BB -0.0068071  0.0039788  -1.711  0.08725 .
## TEAM_PITCHING_SO  0.0020047  0.0008853   2.264  0.02364 *
## TEAM_FIELDING_E  -0.0431249  0.0026584 -16.222  < 2e-16 ***
## TEAM_FIELDING_DP -0.1134331  0.0128500  -8.827  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.52 on 2261 degrees of freedom
## Multiple R-squared:  0.372,  Adjusted R-squared:  0.3681
## F-statistic: 95.68 on 14 and 2261 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = complete(tempData, 2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.704  -8.411   0.207   8.347  50.797
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      36.5168974  5.1994373   7.023 2.86e-12 ***
## TEAM_BATTING_H    0.0418355  0.0035768  11.696  < 2e-16 ***
## TEAM_BATTING_2B  -0.0179102  0.0088422  -2.026   0.0429 *
## TEAM_BATTING_3B   0.0257026  0.0164943   1.558   0.1193
## TEAM_BATTING_HR   0.0643211  0.0263547   2.441   0.0147 *
## TEAM_BATTING_BB   0.0123282  0.0055763   2.211   0.0271 *
## TEAM_BATTING_SO  -0.0169496  0.0024841  -6.823 1.14e-11 ***
## TEAM_BASERUN_SB   0.0532008  0.0051302  10.370  < 2e-16 ***
## TEAM_BASERUN_CS   0.0026719  0.0102632   0.260   0.7946
## TEAM_PITCHING_H   0.0015044  0.0003832   3.926 8.89e-05 ***
## TEAM_PITCHING_HR  0.0224096  0.0233940   0.958   0.3382
## TEAM_PITCHING_BB -0.0034978  0.0039907  -0.877   0.3809
## TEAM_PITCHING_SO  0.0014350  0.0008874   1.617   0.1060
## TEAM_FIELDING_E  -0.0413860  0.0026344 -15.710  < 2e-16 ***
## TEAM_FIELDING_DP -0.1130479  0.0127012  -8.901  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.57 on 2261 degrees of freedom
## Multiple R-squared:  0.3671, Adjusted R-squared:  0.3632
## F-statistic: 93.67 on 14 and 2261 DF,  p-value: < 2.2e-16

## Warning: package 'glmnet' was built under R version 3.4.2

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 3.4.2

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':
##
##     expand

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.4.3

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded glmnet 2.0-13

## Warning in plotCoef(x$beta, lambda = x$lambda, df = x$df, dev = x
## $dev.ratio, : 1 or less nonzero coefficients; glmnet plot is not meaningful
```



```
##
## Call:
## lm(formula = TARGET_WINS ~ . + I(TEAM_BASERUN_SB/TEAM_BASERUN_CS) +
##     I((TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) +
##         2 * (TEAM_BATTING_2B) + 3 * TEAM_BATTING_3B + 4 * TEAM_BATTING_HR),
##     data = dfT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.2669  -6.6797  -0.0982   6.5361  27.8256
##
## Coefficients: (1 not defined because of singularities)
##
## (Intercept)
## 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
## I(TEAM_BASERUN_SB/TEAM_BASERUN_CS)
## I((TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) + 2 * (TEAM_BATTING_2B) + 3
##
## (Intercept)
## 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
## I(TEAM_BASERUN_SB/TEAM_BASERUN_CS)
## I((TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) + 2 * (TEAM_BATTING_2B) + 3
##
## (Intercept)
## 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
## I(TEAM_BASERUN_SB/TEAM_BASERUN_CS)
## I((TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) + 2 * (TEAM_BATTING_2B) + 3
##
## (Intercept)
## 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
## I(TEAM_BASERUN_SB/TEAM_BASERUN_CS)
## I((TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) + 2 * (TEAM_BATTING_2B) + 3
##
## (Intercept)
## 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
## I(TEAM_BASERUN_SB/TEAM_BASERUN_CS)
## I((TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) + 2 * (TEAM_BATTING_2B) + 3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.553 on 1470 degrees of freedom
##   (790 observations deleted due to missingness)
## Multiple R-squared:  0.4393, Adjusted R-squared:  0.4336
## F-statistic: 76.79 on 15 and 1470 DF,  p-value: < 2.2e-16
```

# 4. SELECT MODELS

Decide on the criteria for selecting the best multiple linear regression model. Will you select a model with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your model. For the multiple linear regression model, will you use a metric such as Adjusted R2, RMSE, etc.? Be sure to explain how you can make inferences from the model, discuss multi-collinearity issues (if any), and discuss other relevant model output. Using the training data set, evaluate the multiple linear regression model based on (a) mean squared error, (b) R2, (c) F-statistic, and (d) residual plots. Make predictions using the evaluation data set.

Based off of the models, we can see that the model with the hightiest $R^2$ was the one with the co-linear variables removed with an 0.4045048

# Apendix

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(ggthemes)
library(GGally)
dfT <- read_csv('moneyball-training-data.csv')
dfT <- dfT %>% select(-INDEX)
glimpse(dfT)
dfE <- read_csv('moneyball-evaluation-data.csv')
dfE <- dfE %>% select(-INDEX)
glimpse(dfE)
naByCol <- function(df){
  x = data.frame(varName = character(),
                 numNA = integer())
  for (i in colnames(df)) {
    y =  sum(is.na(df[,i]))
    newrow = data.frame(varName = i, numNA = y)
    x <- rbind(x, newrow)
  }
  p = ggplot(x, aes(x = varName, y = numNA)) +
    geom_bar(stat = 'identity') +
    xlab("Variabel")+ ylab('Number of NAs')+
    ggtitle("Number of NAs in each Variable") +
    theme_tufte() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))

  return(p)
}
naByCol(dfT)
naByCol(dfE)
ggplot(data = gather(dfT), mapping = aes(x = value)) +
  geom_histogram(bins = 20) + facet_wrap(~key, scales = 'free') +
  theme_tufte()

ggplot(data = gather(dfE), mapping = aes(x = value)) +
  geom_histogram(bins = 20) + facet_wrap(~key, scales = 'free') +
  theme_tufte()
dfT %>%
  scale() %>%
  as_tibble() %>%
  gather() %>%
  ggplot(aes(x = key, y = value)) +
  geom_boxplot()+
  theme_tufte() +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1))
  coord_flip() +
  ylab('Scaled Values')+
  xlab('Variable')+
  ggtitle('Scaled Values', subtitle = 'Values scaled to presen on a common axis')

dfT %>%
  gather() %>%
  ggplot(aes(x = key, y = value)) +
  geom_boxplot()+
  theme_tufte() +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1))
  coord_flip()

corr <- round(cor(dfT, use = "complete.obs"), 1) # Complete obs b/c of all NAs
ggcorr(corr, hjust = 1, size = 3, color = "grey50",
       layout.exp = 1, label = TRUE, label_size = 3,
       label_alpha = TRUE)
summary(dfT)
corr2 <- round(cor(dfT[,-1], dfT$TARGET_WINS, use = "complete.obs"), 3) # For target:
corr2 <- as.data.frame(corr2) %>% rownames_to_column(var = "Row_name" )%>% as_tibble()
corr2 <- corr2 %>% rename(Correlation = V1)
corrPlotFunc <- function(corr2){
ggplot(data = corr2,
       aes(x = reorder(Row_name, abs(Correlation)),
           y = Correlation))+
  geom_bar(stat = 'identity') +
  geom_text(aes(label=Correlation),
            hjust = ifelse(corr2$Correlation >= 0, -0.3, 1.3),
            size = 3, color = 'grey50') +
  coord_flip()+
  ylim(-.6, .6)+
  xlab("Satistic")+
  ggtitle("Correlation Between Factors and Wins")+
  theme_tufte()
}
corrPlotFunc(corr2)
dfMissing <- dfT %>% mutate(isMissing = ifelse(is.na(.$TEAM_BATTING_HBP), 1, 0))
corCoef <- cor(dfMissing$TARGET_WINS, dfMissing$isMissing)
corCoef
t <- corCoef * sqrt((nrow(dfT) - 2) / (1 - corCoef ^2))
t
pt(q = t, df = nrow(dfT) - 2)
tVal <- corCoef * sqrt((nrow(dfMissing - 2)) / (1 - corCoef^2))
```

```
tVal
dfT <- dfT %>% select(-TEAM_BATTING_HBP)
dfE <- dfE %>% select(-TEAM_BATTING_HBP)
library(mice)
tempData <- mice(dfT, m =5,maxit=50,meth='pmm',seed=500)

#imputed <- mice(dfT, m = 5, maxit = 50, meth = 'pmm', seed = 500)
fitAll <- lm(TARGET_WINS ~ ., dfT)
summary(fitAll)
regressionDiagnostic <- function(fit){
  ## https://www.statmethods.net/stats/rdiagnostics.html
  library(car) # Required
  print(outlierTest(fit))
  qqPlot(fit, main = 'QQ Plot')
  #av.Plots(fit)
  cutoff <- 4/((nrow(dfT)-length(fitAll$coefficients)-2))
  plot(fitAll, which=4, cook.levels=cutoff)
  # Influence Plot
  influencePlot(fitAll, id.method="identify", main="Influence Plot",
                sub="Circle size is proportial to Cook's Distance" )
  library(MASS)
  sresid <- studres(fit)
  hist(sresid, freq=FALSE,
       main="Distribution of Studentized Residuals")
  xfit<-seq(min(sresid),max(sresid),length=40)
  yfit<-dnorm(xfit)
  lines(xfit, yfit)
  print(ncvTest(fitAll))
  # plot studentized residuals vs. fitted values
  spreadLevelPlot(fit)
  print(durbinWatsonTest(fit))
}


regressionDiagnostic(fitAll)

cutoff <- 4/((nrow(dfT)-length(fitAll$coefficients)-2))
plot(fitAll, which=4, cook.levels=cutoff)
# Influence Plot
influencePlot(fitAll,   id.method="identify", main="Influence Plot", sub="Circle size is proportial to C
library(ggfortify)
autoplot(fitAll) + geom_point(color = 'grey40') +  theme_tufte()
fitSig <- lm(TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_
summary(fitSig)
anova(fitSig, fitAll)
fitNonCorr <- lm(TARGET_WINS ~ . -TEAM_BATTING_SO -TEAM_BATTING_BB - TEAM_BATTING_HR -TEAM_BATTING_H ,
summary(fitNonCorr)
fit1 <- lm(TARGET_WINS ~ ., complete(tempData, 1))
summary(fit1)
fit2 <- lm(TARGET_WINS ~ ., complete(tempData, 2))
summary(fit2)
library(glmnet)
x <- dfT %>% na.omit()
```

```r
x <- as.matrix(x[,-1])
y <- as.matrix(x[,1])
lassReg <- glmnet(x,y, alpha = 1, family="gaussian")
plot(lassReg, xvar = "lambda")
interact <- lm(TARGET_WINS ~ . + I(TEAM_BASERUN_SB / TEAM_BASERUN_CS) + I((TEAM_BATTING_H - TEAM_BATTIN
summary(interact)
```