DATA 621 Homework 1

Ilya Kats

Summary

This report covers an attempt to build a model to predict number of wins of a baseball team in a season based on several offensive and deffensive statistics. Resulting model explained about 36% of variability in the target variable and included most of the provided explanatory variables. Some potentially helpful variables were not included in the data set. For instance, number of At Bats can be used to calculate on-base percentage which may correlate strongly with winning percentage. The model can be revised with additional variables or further analysis.

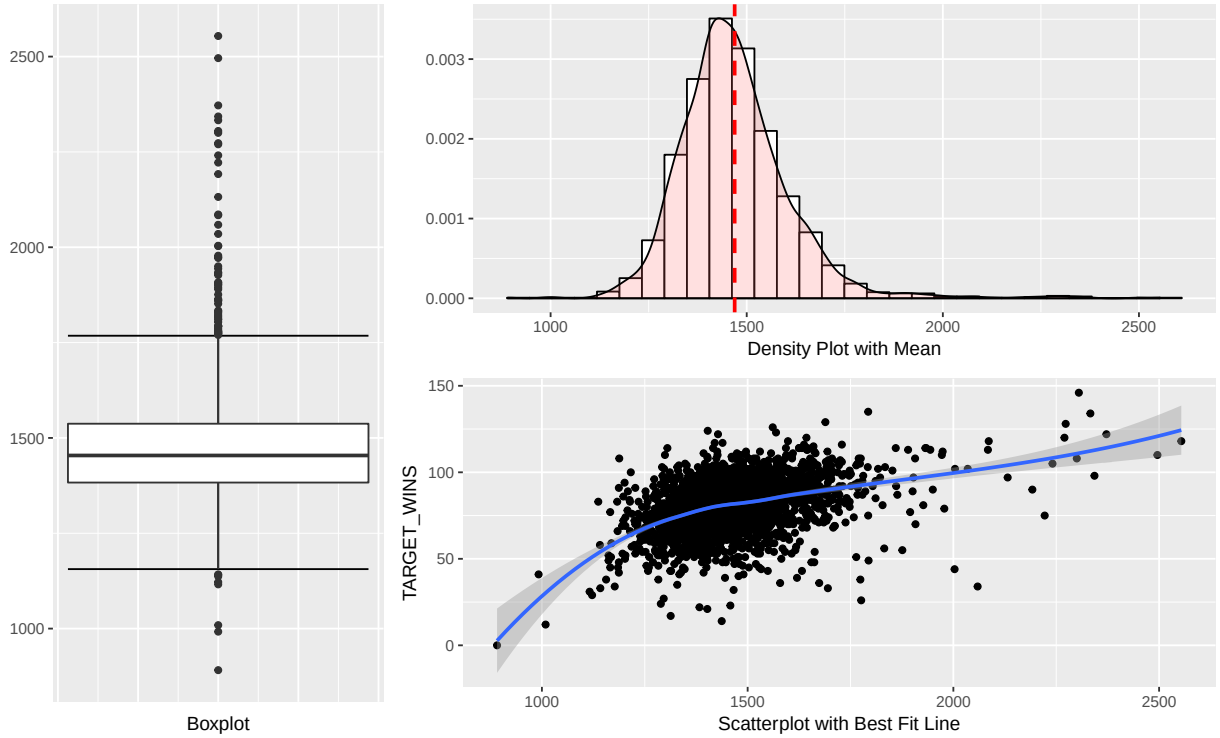
Data Exploration

The data set describes baseball team statistics for the years 1871 to 2006 inclusive. Each record in the data set represents the performance of the team for the given year adjusted to the current length of the season - 162 games. The data set includes 16 variables and the training set includes 2,276 records.

Each variable is presented below with corresponding basic statistics (minimum, median and maximum values, mean and standard deviation, number of records with missing values and zero values), boxplot, density plot with highlighted mean value, and scatterplot against outcome variable (`TARGET_WINS`) with best fit line. This information is used to check general validity of data and adjust as necessary.

TEAM_BATTING_H: Number of team base hits (includes singles, doubles, triples and home runs)

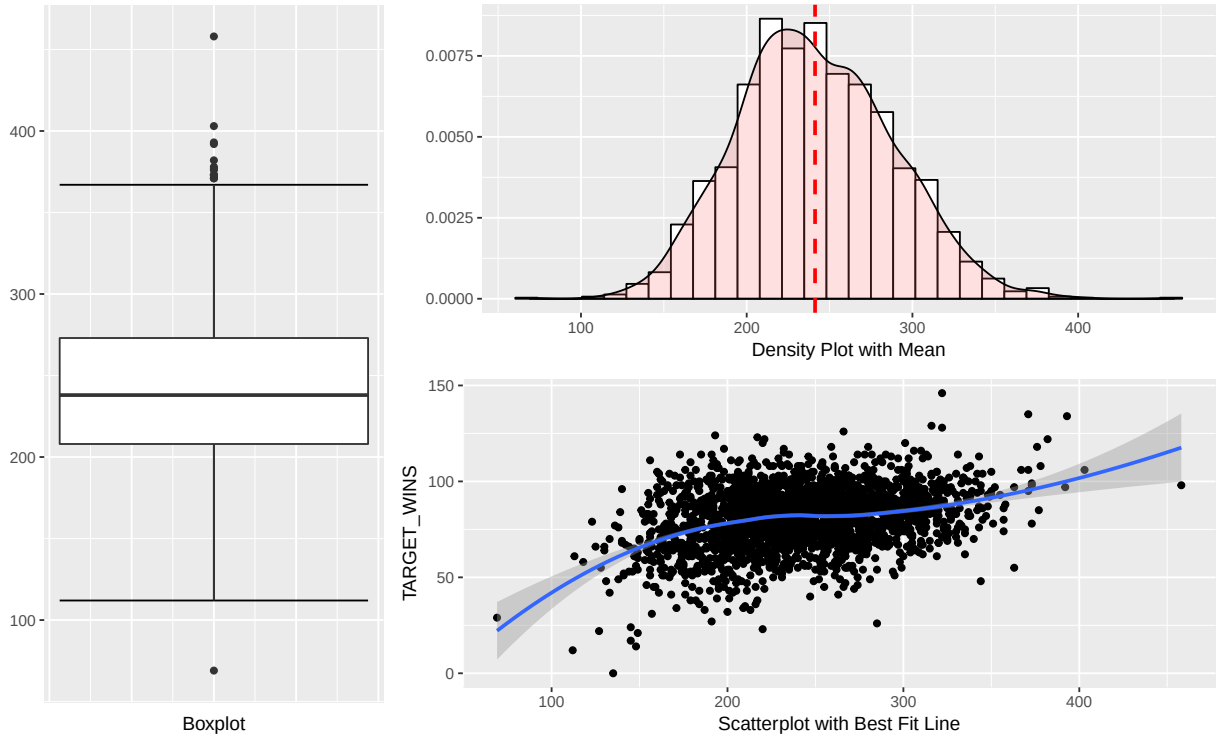
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
891	1454	1469.27	144.5912	2554	0	0



Analysis: There are no missing values. The range and distribution are reasonable.

TEAM_BATTING_2B: Number of team doubles

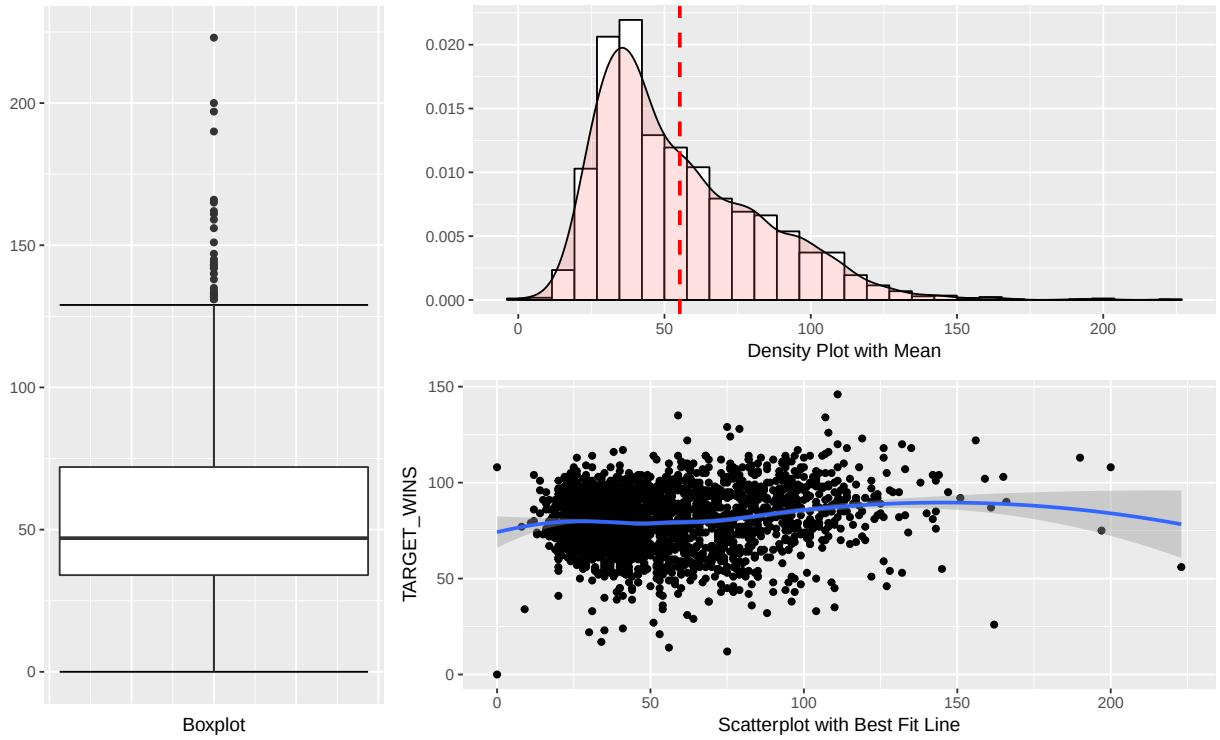
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
69	238	241.2469	46.80141	458	0	0



Analysis: There are no missing values. The range and distribution are reasonable.

TEAM_BATTING_3B: Number of team triples

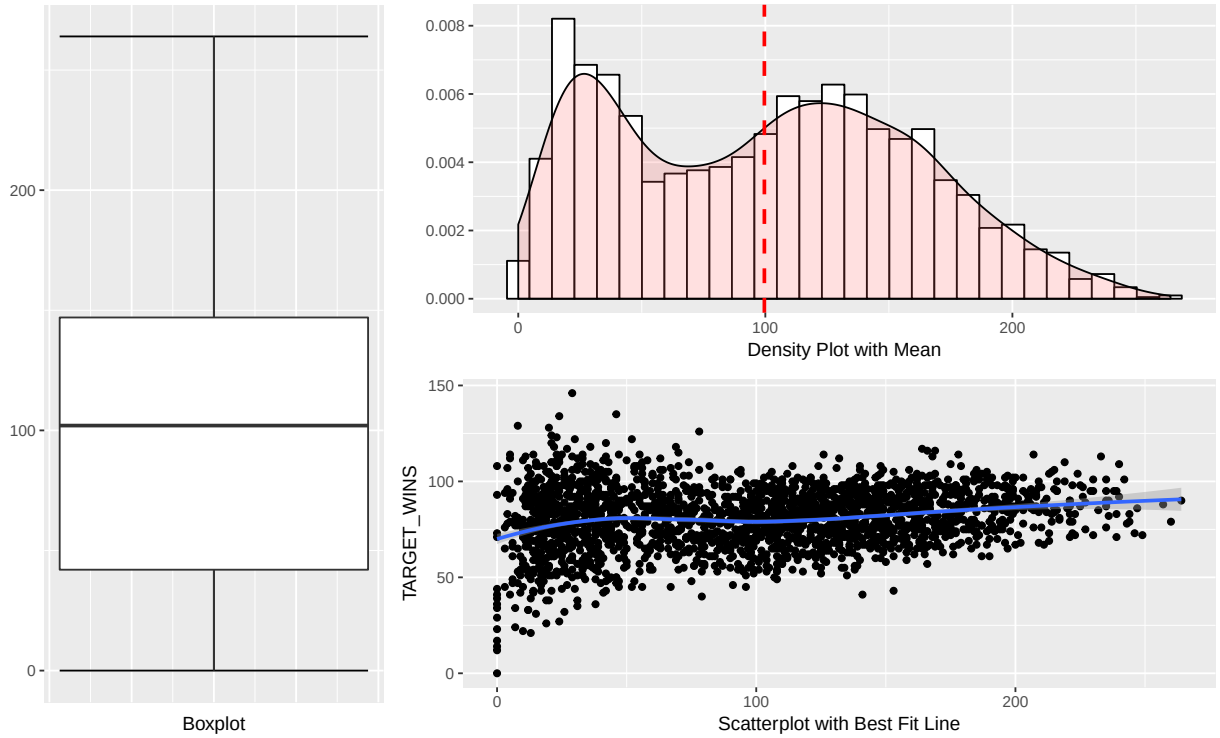
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	47	55.25	27.93856	223	0	2



Analysis: The range and distribution are reasonable. There are 2 records with zero values which is unrealistic for a team in a season. One record (index 1347) has 12 variables with missing values, including the outcome variable. This record will be deleted from the data set. Second record (index 1494) has 7 missing variables, but it does have some recorded values in all categories - batting, pitching and fielding. Zero value for `TEAM_BATTING_3B` can be replaced with the median (because the distribution is right-skewed, median value will provide more realistic estimate).

TEAM_BATTING_HR: Number of team home runs

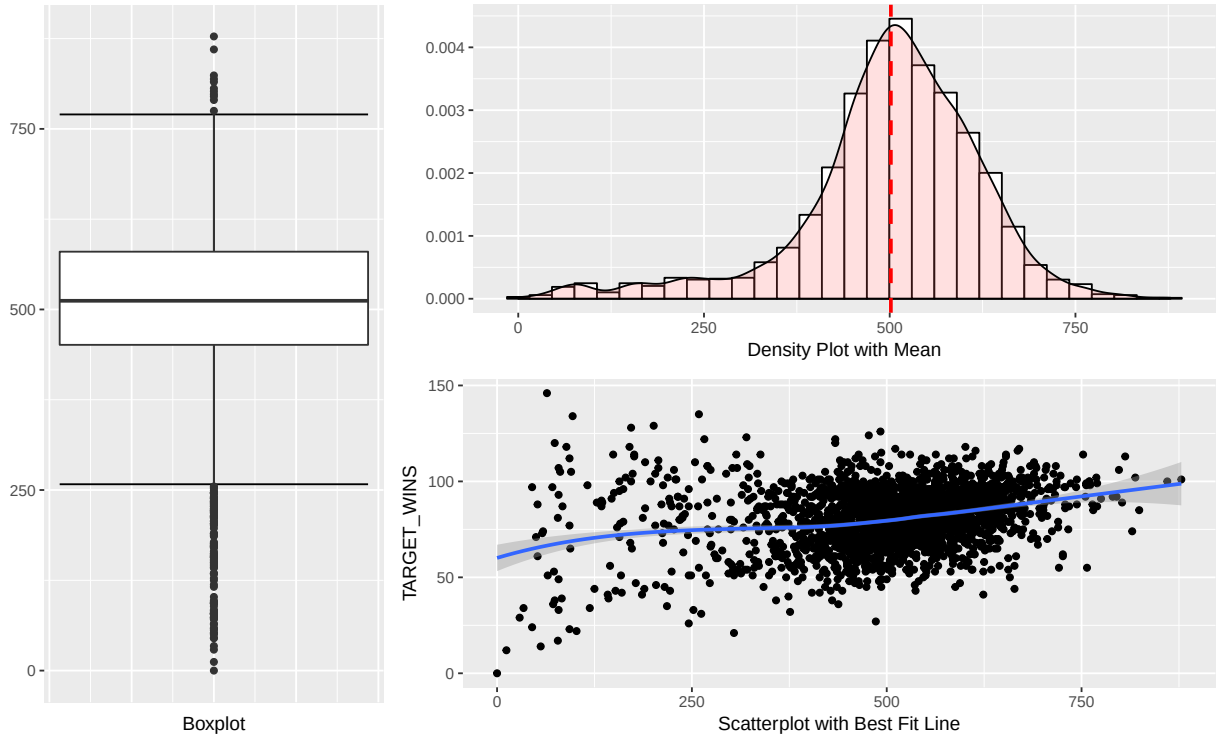
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	102	99.61204	60.54687	264	0	15



Analysis: The range is reasonable. The distribution is interesting because it is multimodal. Most likely this indicates major changes in game dynamics - perhaps, some rule adjustments started favoring batters. Or perhaps, this is an affect of steroid era. There are 15 records with zero values which is unrealistic for this variable. They can be imputed from other values.

TEAM_BATTING_BB: Number of team walks

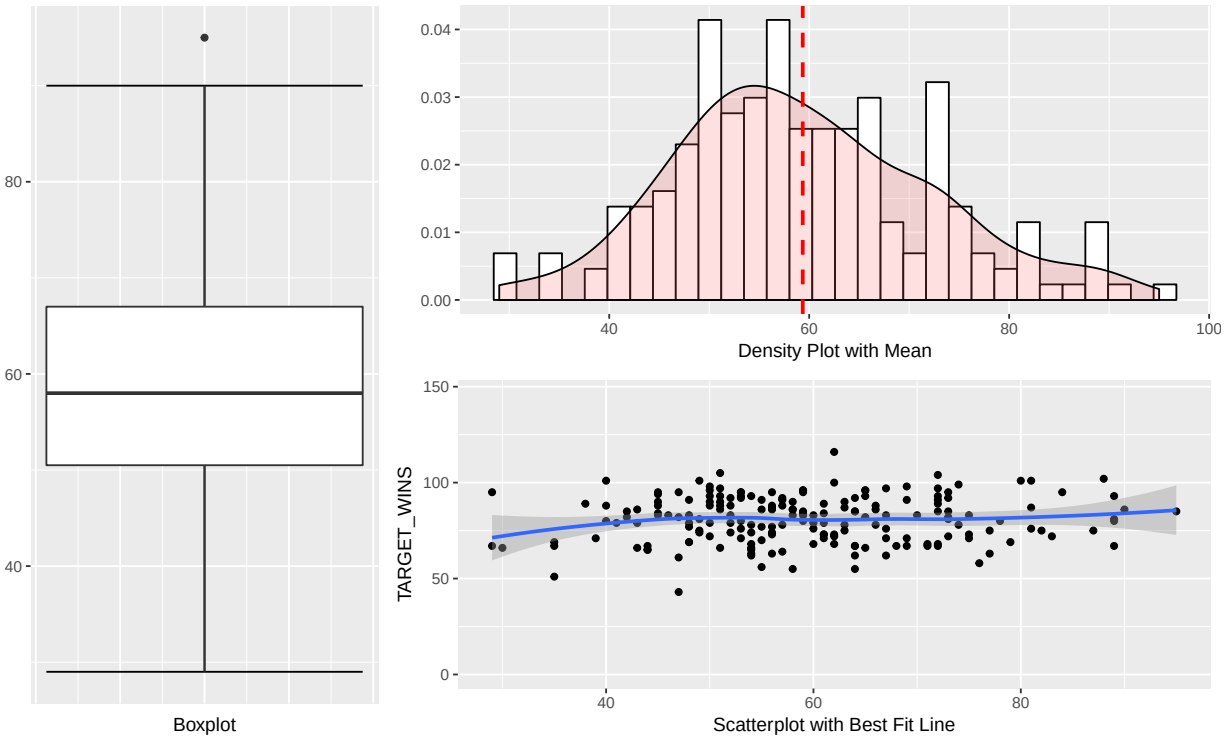
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	512	501.5589	122.6709	878	0	1



Analysis: The range and distribution are reasonable. There is one record (index 1347) that has a zero value. This record was discussed above (under `TEAM_BATTING_3B`) and it will be deleted from the data set.

TEAM_BATTING_HBP: Number of team batters hit by pitch

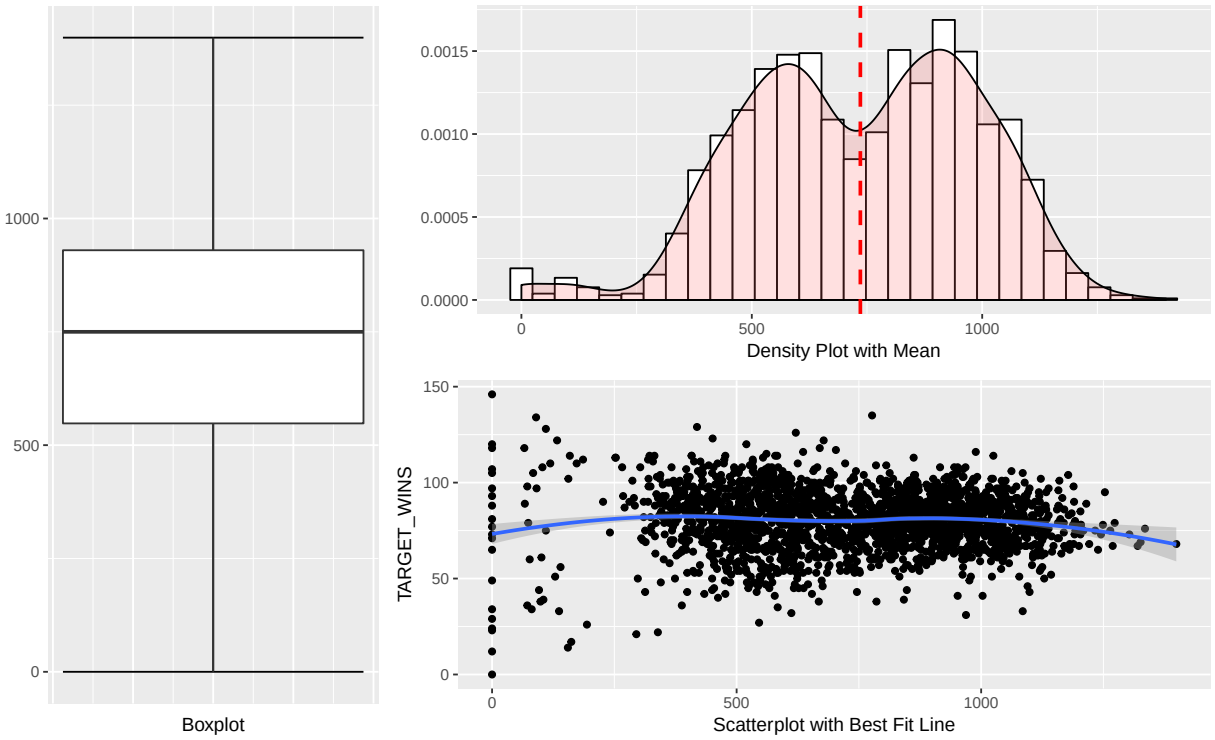
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
29	58	59.35602	12.96712	95	2085	0



Analysis: There are 2,085 records - 91.6% of data set - that are missing value. Because this variable is missing for majority of records, it will not be imputed and will be left out from the regression model.

TEAM_BATTING_SO: Number of team strikeouts by batters

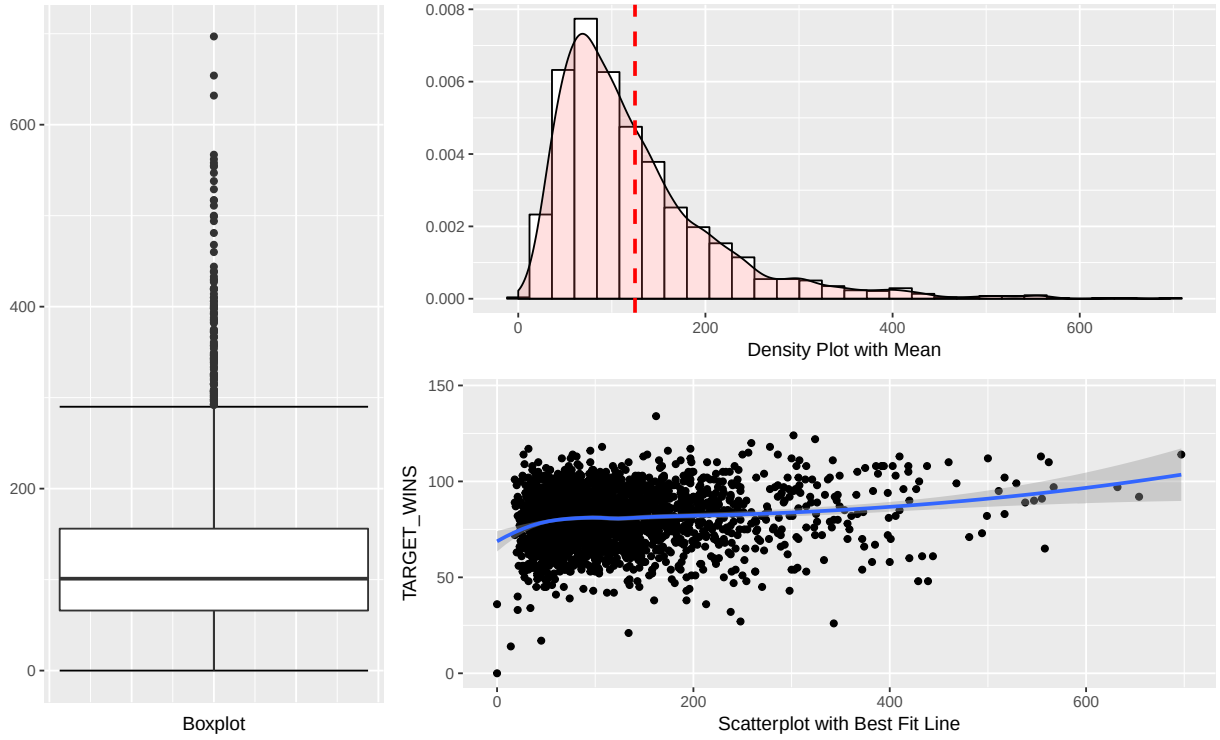
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	750	735.6053	248.5264	1399	102	20



Analysis: There are 122 records with missing or zero value (as wtih other variables a zero value is unrealistic). These values can be imputed. Similarly to homeruns, the distribution is multimodal, which is interesting enough for additional analysis. Another area of concern is a noticeable left tail. It is highly unlikely to have games without any strikeouts, so anything lower than 162 (average of 1 strikeout per game) is definitely suspect.

TEAM_BASERUN_SB: Number of team stolen bases

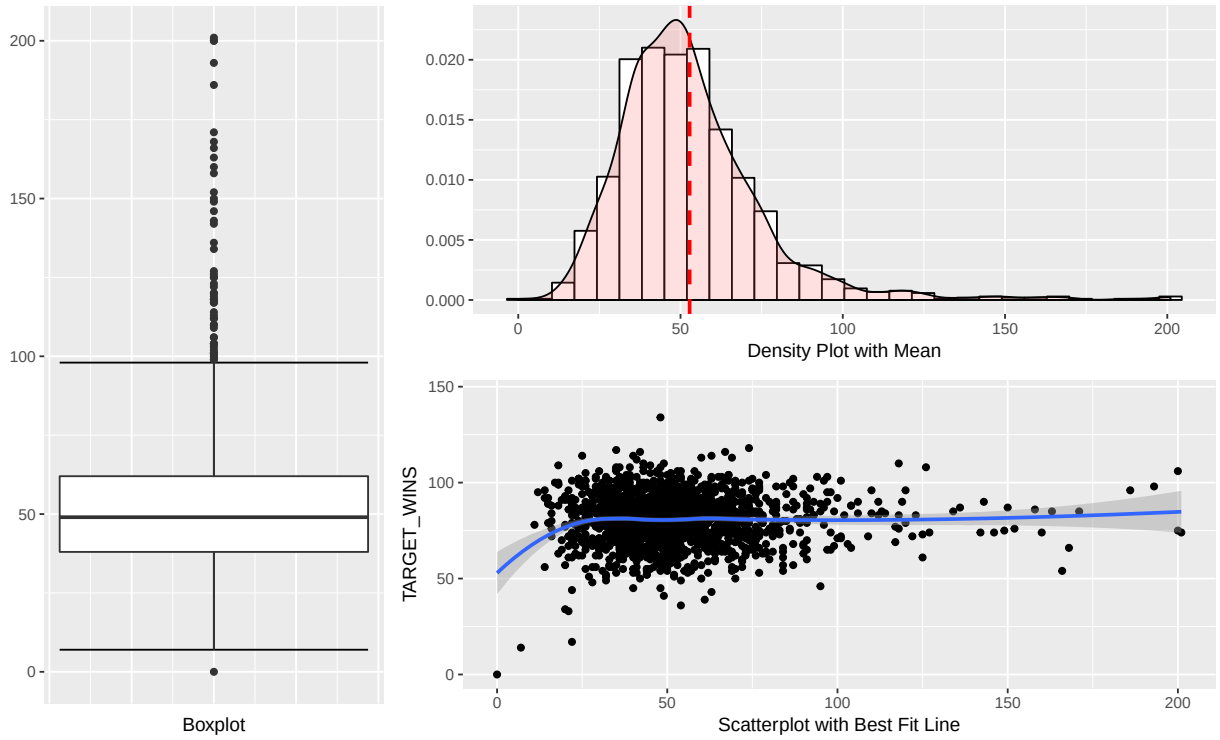
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	101	124.7618	87.79117	697	131	2



Analysis: The range and distribution are reasonable. The only issue are 133 records with missing or zero value. These values can be imputed in order to use these records in model building.

TEAM_BASERUN_CS: Number of team runners caught stealing

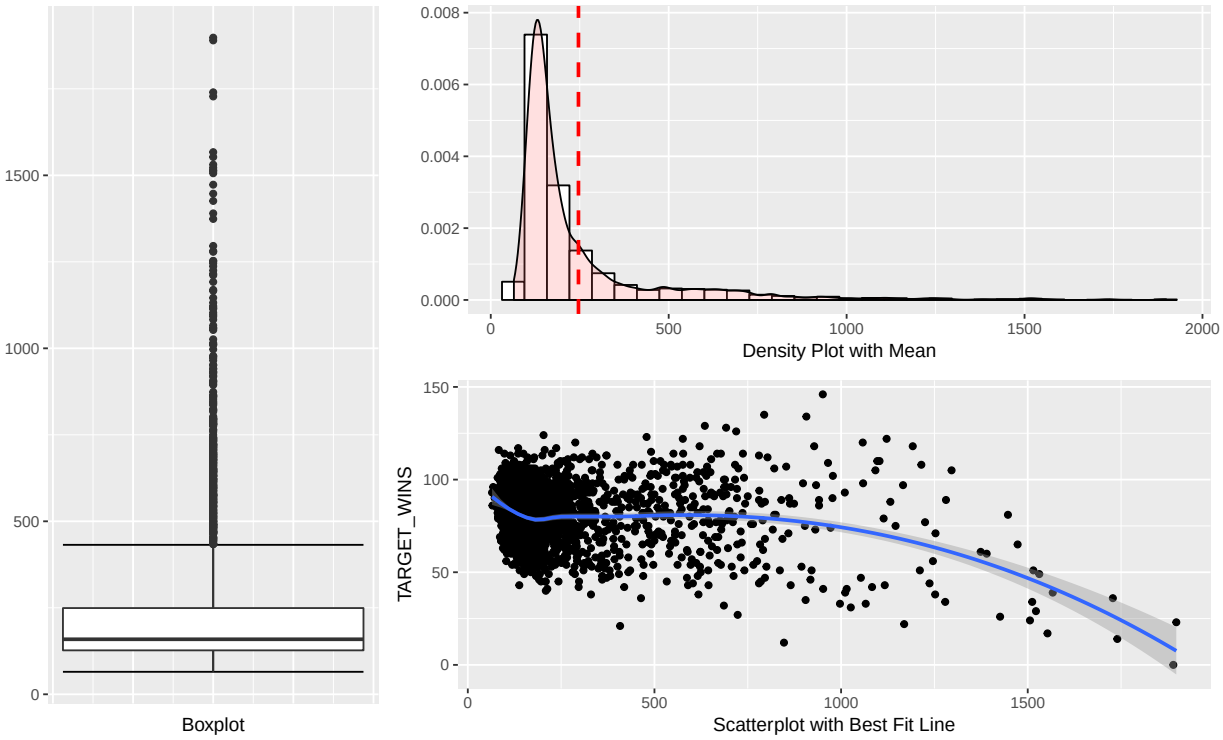
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	49	52.80386	22.95634	201	772	1



Analysis: The range and distribution are reasonable; however, there is significant number of missing values - 773, including one zero value. This represents a third of the entire data set. It may be possible to impute this value, but it may be necessary to leave this variable out of model building.

TEAM_FIELDING_E: Number of team fielding errors

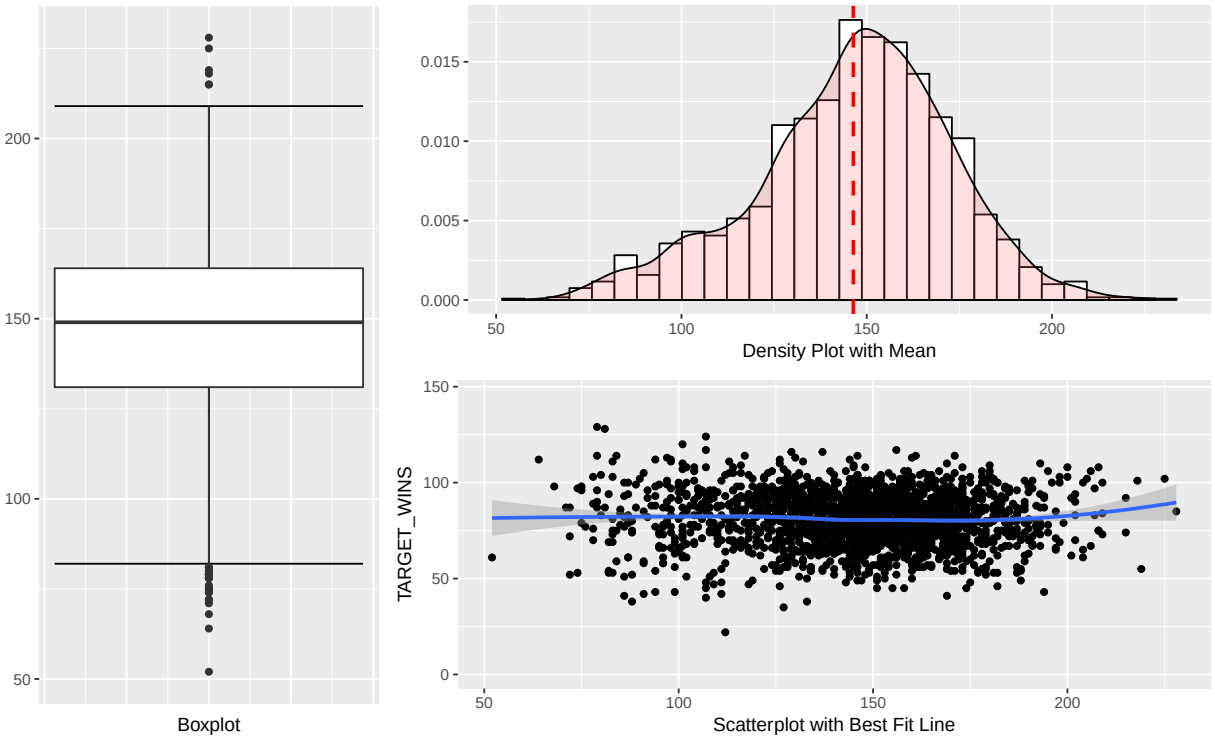
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
65	159	246.4807	227.771	1898	0	0



Analysis: There are no missing values. Distribution has a very long right tail. Values in the 1,000 and above range are highly suspect. One of the highest historical number of errors is 867 errors by Washington in 1886 for 122 games. That is equal about 1,151 errors for 162 game season. There are multiple values above that number. This may unfavorably influence a model.

TEAM_FIELDING_DP: Number of team fielding double plays

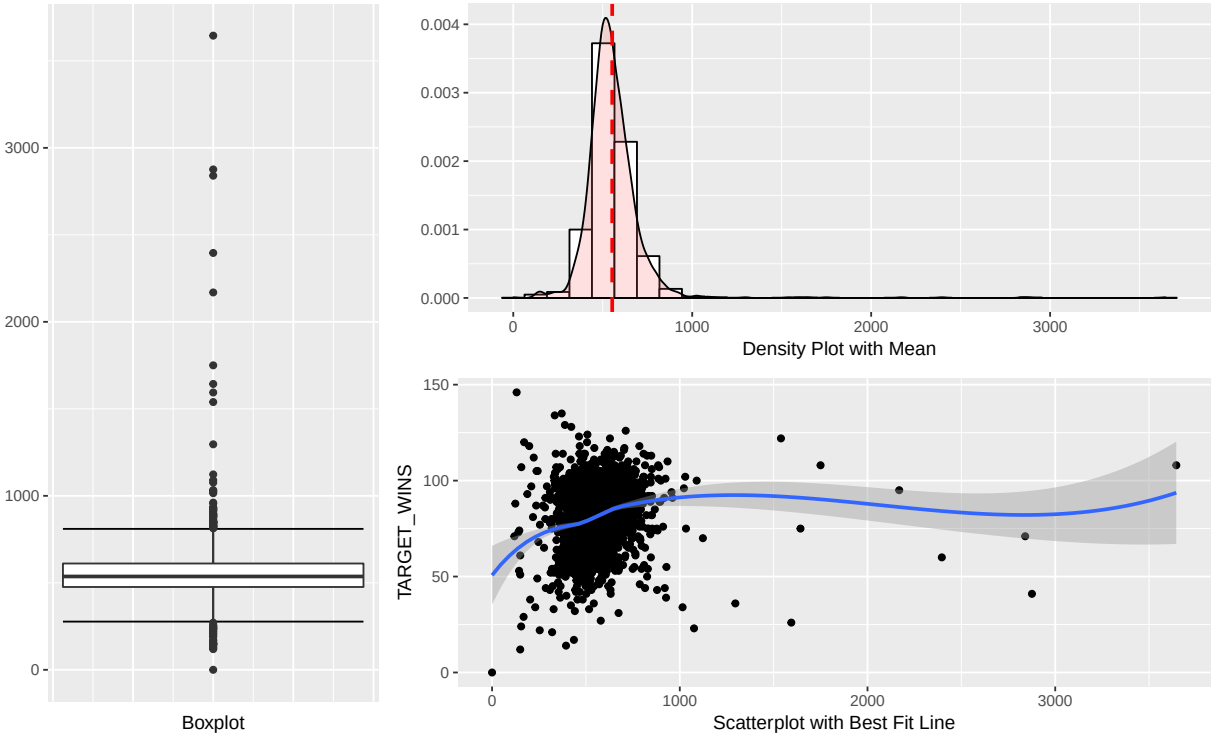
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
52	149	146.3879	26.22639	228	286	0



Analysis: The range and distribution are reasonable. Similar to a few other variables there is a medium number off missing values - 286 records. This value can be imputed.

TEAM_PITCHING_BB: Number of walks given up by pitchers

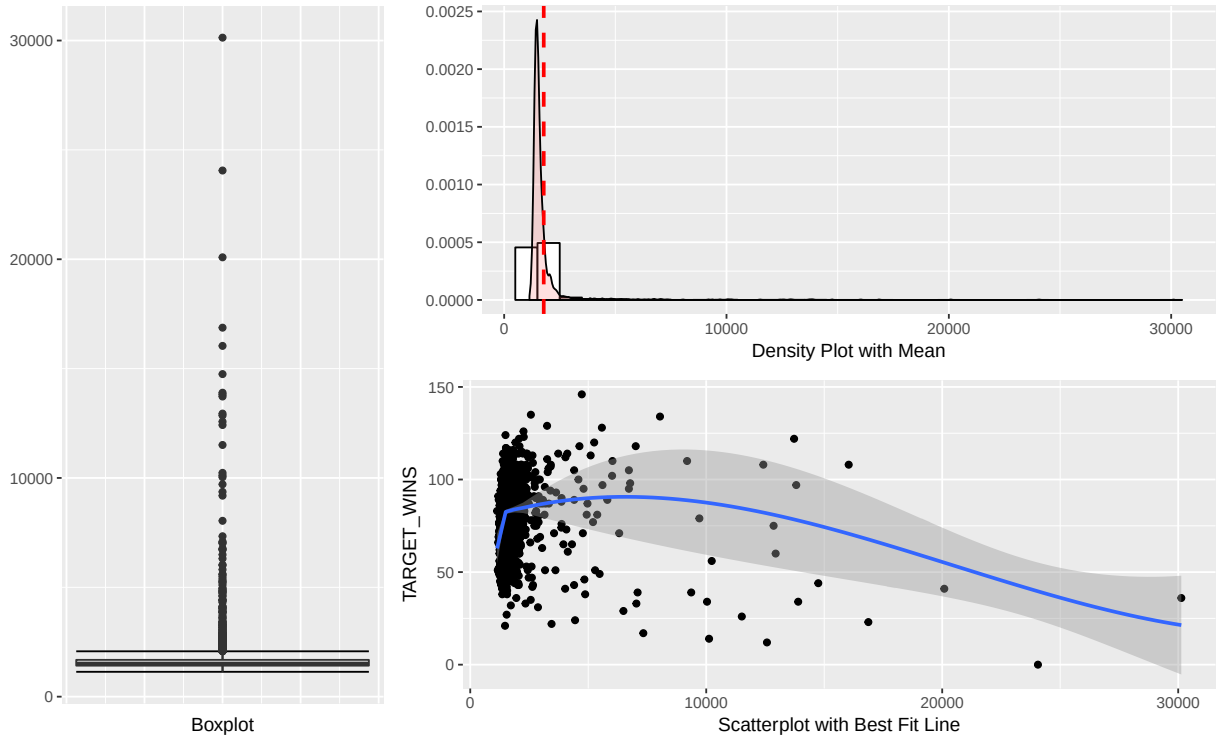
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	536.5	553.0079	166.3574	3645	0	1



Analysis: There are no missing values with the exception of record 1347 which will be deleted from model building. There are some unrealistic outliers. Current record of walks by a team in a season is held by 1949 Boston Red Sox - 835 walks in 155 games. For a 162 game season, this number is 873. This variable will be capped at 1,100 and any value over this will be set to this cap.

TEAM_PITCHING_H: Number of base hits given up by pitchers

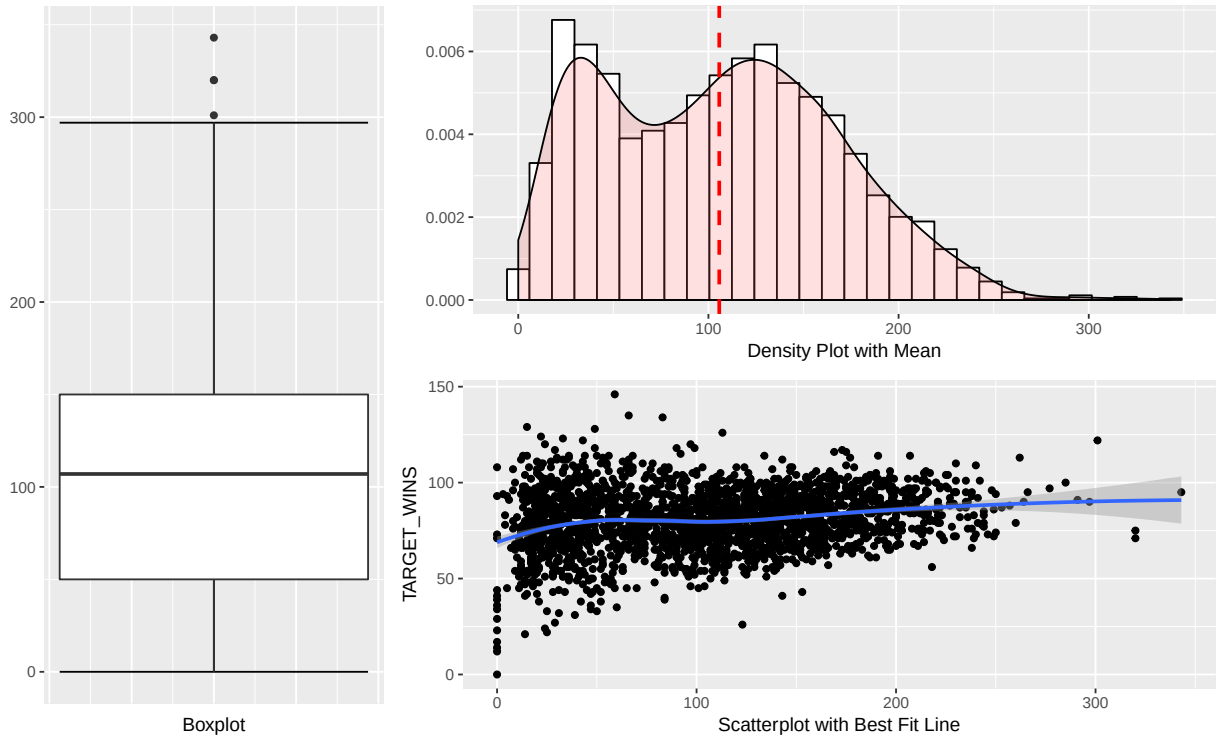
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
1137	1518	1779.21	1406.843	30132	0	0



Analysis: Similar to `TEAM_PITCHING_BB` above, there are no missing value, but there issues with outliers. Based on visualizations, this variable will be capped at 13,000 and any value over this will be set to this cap.

TEAM_PITCHING_HR: Number of home runs given up by pitchers

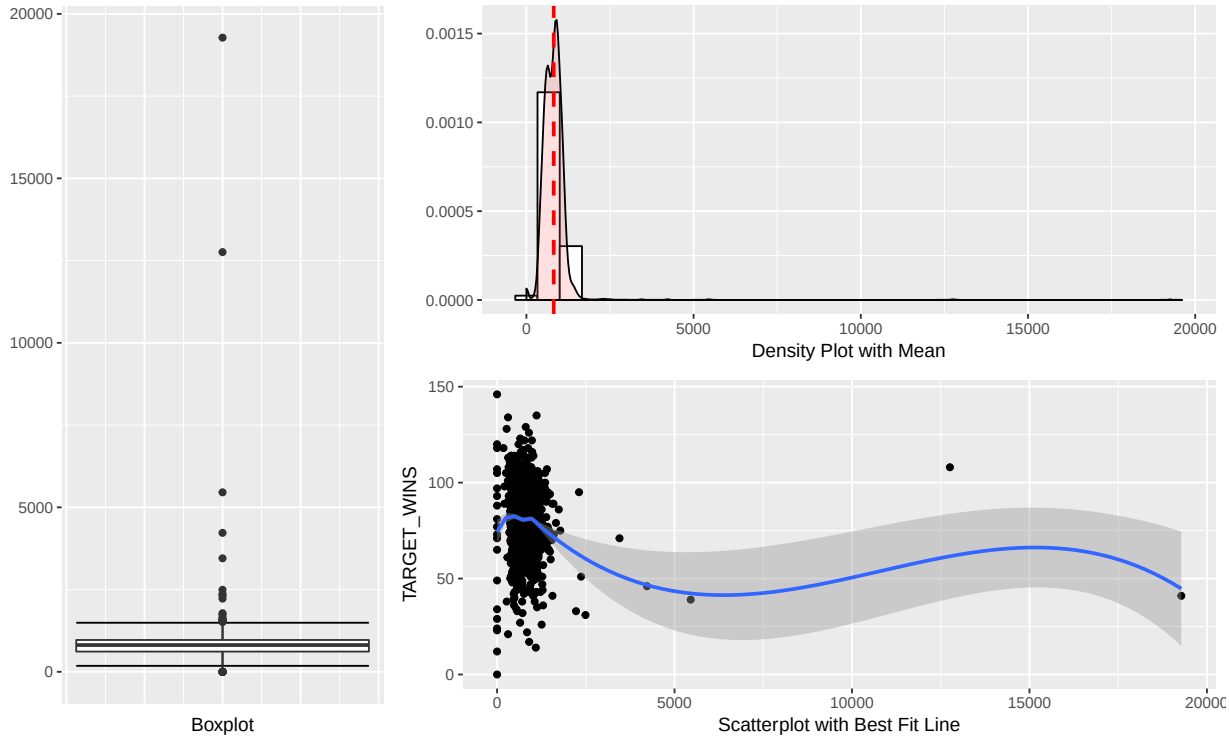
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	107	105.6986	61.29875	343	0	15



Analysis: This variable is more consistent than other pitching variables. The range and distribution are reasonable. Multimodality is interesting similar to a few other variables above. There are 15 zero values which can be imputed as needed.

TEAM_PITCHING_SO: Number of strikeouts by pitchers

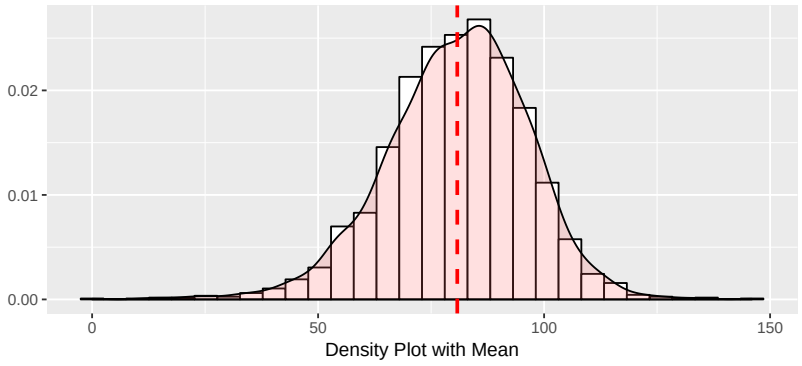
Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	813.5	817.7305	553.085	19278	102	20



Analysis: This variable has 122 missing or zero values. They can be imputed as needed. There is also an outlier issue. Based on visualizations, this variable will be capped at 2,500 and any value over this will be set to this cap.

TARGET_WINS: Number of wins (Outcome)

Min	Median	Mean	SD	Max	Num of NAs	Num of Zeros
0	82	80.79086	15.75215	146	0	1



Analysis: The range and distribution are reasonable. There are no missing values with the exception of record 1347.

Correlation Matrix

	Wins	H	2B	3B	HR	BB	SO	SB	CS	HBP	P-H	P-HR	P-BB	P-SO	E	DP
Wins	1	0.39	0.29	0.14	0.18	0.23	-0.03	0.14	0.02	0.07	-0.11	0.19	0.12	-0.08	-0.18	-0.03
H	0.39	1	0.56	0.43	-0.01	-0.07	-0.46	0.12	0.02	-0.03	0.3	0.07	0.09	-0.25	0.26	0.16
2B	0.29	0.56	1	-0.11	0.44	0.26	0.16	-0.2	-0.1	0.05	0.02	0.45	0.18	0.06	-0.24	0.29
3B	0.14	0.43	-0.11	1	-0.64	-0.29	-0.67	0.53	0.35	-0.17	0.19	-0.57	0	-0.26	0.51	-0.32
HR	0.18	-0.01	0.44	-0.64	1	0.51	0.73	-0.45	-0.43	0.11	-0.25	0.97	0.14	0.18	-0.59	0.45
BB	0.23	-0.07	0.26	-0.29	0.51	1	0.38	-0.11	-0.14	0.05	-0.45	0.46	0.49	-0.02	-0.66	0.43
SO	-0.03	-0.46	0.16	-0.67	0.73	0.38	1	-0.25	-0.22	0.22	-0.38	0.67	0.04	0.42	-0.58	0.15
SB	0.14	0.12	-0.2	0.53	-0.45	-0.11	-0.25	1	0.66	-0.06	0.07	-0.42	0.15	-0.14	0.51	-0.5
CS	0.02	0.02	-0.1	0.35	-0.43	-0.14	-0.22	0.66	1	-0.07	-0.05	-0.42	-0.11	-0.21	0.05	-0.21
HBP	0.07	-0.03	0.05	-0.17	0.11	0.05	0.22	-0.06	-0.07	1	-0.03	0.11	0.05	0.22	0.04	-0.07
P-H	-0.11	0.3	0.02	0.19	-0.25	-0.45	-0.38	0.07	-0.05	-0.03	1	-0.14	0.32	0.27	0.67	-0.23
P-HR	0.19	0.07	0.45	-0.57	0.97	0.46	0.67	-0.42	-0.42	0.11	-0.14	1	0.22	0.21	-0.49	0.44
P-BB	0.12	0.09	0.18	0	0.14	0.49	0.04	0.15	-0.11	0.05	0.32	0.22	1	0.49	-0.02	0.32
P-SO	-0.08	-0.25	0.06	-0.26	0.18	-0.02	0.42	-0.14	-0.21	0.22	0.27	0.21	0.49	1	-0.02	0.03
E	-0.18	0.26	-0.24	0.51	-0.59	-0.66	-0.58	0.51	0.05	0.04	0.67	-0.49	-0.02	-0.02	1	-0.5
DP	-0.03	0.16	0.29	-0.32	0.45	0.43	0.15	-0.5	-0.21	-0.07	-0.23	0.44	0.32	0.03	-0.5	1

Anything over 0.5 or under -0.5 is highlighted in blue. The matrix was created using complete pairwise observations.

A few conclusions:

- Not surprisingly there is a very strong correlation between home runs batted in and home runs given up by pitching.
- There is a negative correlation between number of triples and home runs. A less powerful team may not have enough power to hit home runs, but they get a lot of triples.
- THere is a strong positive correlation between number of strikeouts and home runs. More swings of the bat results in more home runs.

Data Preparation

As noted in the *Data Exploration* section, the following adjustments have been performed:

- Record 1347 having 0 for outcome variable `TARGET_WINS` has been removed.
- Variable `TEAM_BATTING_HBP` has been removed.
- Any zero value in all variables has been converted to `NA`.
- Any `NA` value has been imputed using `aregImpute` function of `Hmisc` R package. R^2 of imputations are as follows:

```
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_SO  TEAM_BASERUN_SB
##        0.7121069        0.9873939        0.9261344        0.7002203
##  TEAM_BASERUN_CS TEAM_FIELDING_DP TEAM_PITCHING_HR TEAM_PITCHING_SO
##        0.6794417        0.4676785        0.9845605        0.8395485
```

- Outliers for several variables have been capped: `TEAM_PITCHING_SO` at 2,500, `TEAM_PITCHING_H` at 13,000, and `TEAM_PITCHING_BB` at 1100.
- To even out the spread of `TEAM_FIELDING_E` which has a long right tail with low median value, it has been log-transformed.
- A new variable has been created to calculate number of singles batting in. It is equal to number of base hits minus doubles, triples and home runs.

Model Building

Model 1

The first model includes several variables, selected manually, that have higher than average correlation to the target variable. They cover hitting, walking and fielding errors.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB +
##     TEAM_FIELDING_E, data = bb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.971  -9.022   0.101   9.062  51.557
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.089556   3.311975   1.235    0.217
## TEAM_BATTING_H   0.049143   0.002100  23.403  < 2e-16 ***
## TEAM_BATTING_BB  0.016107   0.003136   5.137 3.03e-07 ***
## TEAM_FIELDING_E -0.014493   0.001768  -8.196 4.11e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.7 on 2271 degrees of freedom
## Multiple R-squared:  0.2356, Adjusted R-squared:  0.2346
## F-statistic: 233.3 on 3 and 2271 DF,  p-value: < 2.2e-16
```

All variables are significant, but the R^2 value is relatively small at 0.2356.

Model 2

The second model expand the base hit variable, `TEAM_BATTING_H`, into its components - singles, doubles, triples and home runs.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_S + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_FIELDING_E,
##     data = bb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.256  -8.827   0.093   8.755  60.128
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.097149   3.444196   2.351  0.01881 *
## TEAM_BATTING_S   0.045440   0.003160  14.381  < 2e-16 ***
## TEAM_BATTING_2B  0.022480   0.007413   3.033  0.00245 **
## TEAM_BATTING_3B  0.161033   0.015123  10.648  < 2e-16 ***
## TEAM_BATTING_HR  0.079003   0.007729  10.222  < 2e-16 ***
## TEAM_BATTING_BB  0.012572   0.003198   3.932 8.69e-05 ***
## TEAM_FIELDING_E -0.018552   0.001975  -9.393  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.53 on 2268 degrees of freedom
## Multiple R-squared:  0.2564, Adjusted R-squared:  0.2545
## F-statistic: 130.4 on 6 and 2268 DF,  p-value: < 2.2e-16
```

All variables are still significant and R^2 is slightly improved at 0.2574. Another variation of this model - with log-transformed fielding error variable - produced slightly worse results.

Model 3

The third model includes several variables manually selected to try and cover different aspects of the game:

- `TEAM_BATTING_SO:TEAM_BATTING_H` interaction covers offensive successes (hits) and failures (strikeouts).
- Similarly `TEAM_BATTING_BB:TEAM_BATTING_H` interaction covers interaction between hits and walks.
- `TEAM_BASERUN_SB` covers base running.
- `TEAM_FIELDING_DP` and `TEAM_FIELDING_E_LOG` cover fielding performance.
- `TEAM_PITCHING_HR` covers pitching performance.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_SO:TEAM_BATTING_H + TEAM_BATTING_BB:TEAM_BATTING_H +
##     TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_PITCHING_HR +
##     TEAM_FIELDING_E_LOG, data = bb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.521  -8.122   0.181   8.336  77.398
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.604e+02  6.782e+00  23.653  < 2e-16 ***
## TEAM_BATTING_SO                -8.645e-02  5.729e-03 -15.089  < 2e-16 ***
## TEAM_BASERUN_SB                 4.499e-02  4.226e-03  10.645  < 2e-16 ***
## TEAM_FIELDING_DP               -1.207e-01  1.336e-02  -9.034  < 2e-16 ***
## TEAM_PITCHING_HR                4.206e-02  7.603e-03   5.532 3.54e-08 ***
## TEAM_FIELDING_E_LOG            -1.264e+01  9.139e-01 -13.830  < 2e-16 ***
## TEAM_BATTING_SO:TEAM_BATTING_H  4.717e-05  4.388e-06  10.750  < 2e-16 ***
## TEAM_BATTING_H:TEAM_BATTING_BB  9.360e-06  1.973e-06   4.745 2.21e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.34 on 2267 degrees of freedom
## Multiple R-squared:  0.2772, Adjusted R-squared:  0.275
## F-statistic: 124.2 on 7 and 2267 DF,  p-value: < 2.2e-16
```

All variables are statistically significant and there is noticeable improvement of the R^2 value at 0.3054.

Model 4

The fourth model started with all variables and used backward elimination to arrive at the optimal model. It started with the following variables: TEAM_BATTING_S, TEAM_BATTING_2B, TEAM_BATTING_3B, TEAM_BATTING_HR, TEAM_BATTING_BB, TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_FIELDING_DP, TEAM_FIELDING_E, TEAM_PITCHING_BB, TEAM_PITCHING_H, TEAM_PITCHING_SO, and TEAM_PITCHING_HR. It was necessary to remove only one variable - TEAM_BASERUN_CS - to arrive at a model with all significant variables.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_S + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
##     TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_FIELDING_E + TEAM_PITCHING_BB +
##     TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_PITCHING_HR, data = bb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.981  -8.491   0.143   8.038  55.053
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      26.3547367  5.2696033   5.001 6.13e-07 ***
## TEAM_BATTING_S    0.0453156  0.0038067  11.904  < 2e-16 ***
## TEAM_BATTING_2B   0.0260188  0.0072131   3.607 0.000316 ***
## TEAM_BATTING_3B   0.1055750  0.0155874   6.773 1.60e-11 ***
## TEAM_BATTING_HR   0.2056336  0.0327574   6.277 4.11e-10 ***
## TEAM_BATTING_BB   0.0301819  0.0099565   3.031 0.002462 **
## TEAM_BATTING_SO  -0.0300436  0.0044912  -6.689 2.81e-11 ***
## TEAM_BASERUN_SB   0.0440435  0.0042113  10.458  < 2e-16 ***
## TEAM_FIELDING_DP -0.1153260  0.0127044  -9.078  < 2e-16 ***
## TEAM_FIELDING_E  -0.0417788  0.0028603 -14.606  < 2e-16 ***
## TEAM_PITCHING_BB -0.0173639  0.0082859  -2.096 0.036230 *
## TEAM_PITCHING_H   0.0020351  0.0005762   3.532 0.000421 ***
## TEAM_PITCHING_SO  0.0199824  0.0034955   5.717 1.23e-08 ***
## TEAM_PITCHING_HR -0.0872307  0.0294968  -2.957 0.003136 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.78 on 2261 degrees of freedom
## Multiple R-squared:  0.3377, Adjusted R-squared:  0.3339
## F-statistic: 88.68 on 13 and 2261 DF,  p-value: < 2.2e-16
```

The R^2 value is 0.3609.

Model 5

Additionally, several models were created by trying out some variables and there interactions. Variables were selected either based on theoretical expectation or correlation information from the first section. The following model has R^2 values of 0.3279, which is relatively close to the fourth model; however, this model has fewer variables and may be preferential because of its simplicity.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_S + TEAM_BATTING_3B +
##     TEAM_BATTING_HR + TEAM_BASERUN_SB + TEAM_FIELDING_E_LOG *
##     TEAM_PITCHING_H, data = bb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.059  -8.801   0.035   8.512  57.750
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        30.9340618  8.1098902   3.814  0.00014
## TEAM_BATTING_S                      0.0454458  0.0031100  14.613  < 2e-16
## TEAM_BATTING_3B                     0.1520675  0.0154675   9.831  < 2e-16
## TEAM_BATTING_HR                     0.0748215  0.0072468  10.325  < 2e-16
## TEAM_BASERUN_SB                     0.0459950  0.0038552  11.931  < 2e-16
## TEAM_FIELDING_E_LOG                -6.2474470  1.3855751  -4.509 6.85e-06
## TEAM_PITCHING_H                     0.0323585  0.0045579   7.099 1.67e-12
## TEAM_FIELDING_E_LOG:TEAM_PITCHING_H -0.0046453  0.0006433  -7.221 7.03e-13
##
## (Intercept)                        ***
## TEAM_BATTING_S                     ***
## TEAM_BATTING_3B                    ***
## TEAM_BATTING_HR                    ***
## TEAM_BASERUN_SB                    ***
## TEAM_FIELDING_E_LOG                ***
## TEAM_PITCHING_H                    ***
## TEAM_FIELDING_E_LOG:TEAM_PITCHING_H ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.02 on 2267 degrees of freedom
## Multiple R-squared:  0.3108, Adjusted R-squared:  0.3087
## F-statistic:   146 on 7 and 2267 DF,  p-value: < 2.2e-16
```

Model Selection

Based on R^2 value, the fourth model was selected for further analysis. This model also has the lowest AIC score.

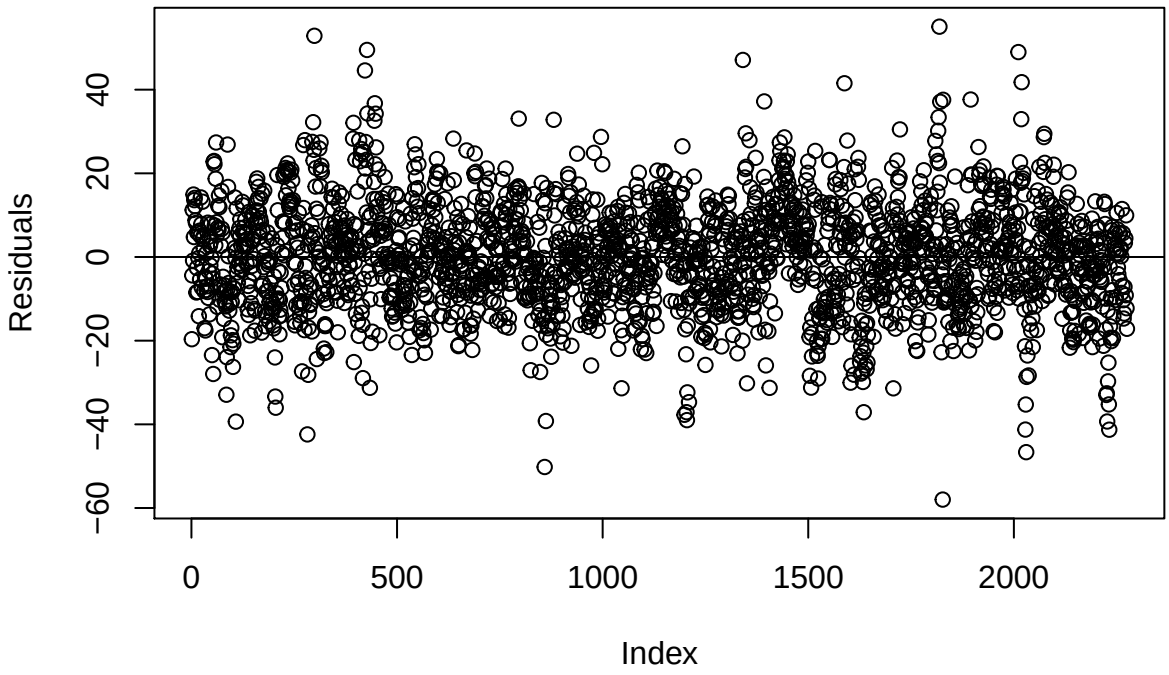
```
##    df      AIC
## m1  5 18372.69
## m2  8 18315.89
## m3  9 18253.31
## m4 15 18066.59
## m5  9 18145.19
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_S + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
##     TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_FIELDING_E + TEAM_PITCHING_BB +
##     TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_PITCHING_HR, data = bb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.981  -8.491   0.143   8.038  55.053
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      26.3547367  5.2696033   5.001 6.13e-07 ***
## TEAM_BATTING_S    0.0453156  0.0038067  11.904  < 2e-16 ***
## TEAM_BATTING_2B   0.0260188  0.0072131   3.607 0.000316 ***
## TEAM_BATTING_3B   0.1055750  0.0155874   6.773 1.60e-11 ***
## TEAM_BATTING_HR   0.2056336  0.0327574   6.277 4.11e-10 ***
## TEAM_BATTING_BB   0.0301819  0.0099565   3.031 0.002462 **
## TEAM_BATTING_SO  -0.0300436  0.0044912  -6.689 2.81e-11 ***
## TEAM_BASERUN_SB   0.0440435  0.0042113  10.458  < 2e-16 ***
## TEAM_FIELDING_DP -0.1153260  0.0127044  -9.078  < 2e-16 ***
## TEAM_FIELDING_E  -0.0417788  0.0028603 -14.606  < 2e-16 ***
## TEAM_PITCHING_BB -0.0173639  0.0082859  -2.096 0.036230 *
## TEAM_PITCHING_H   0.0020351  0.0005762   3.532 0.000421 ***
## TEAM_PITCHING_SO  0.0199824  0.0034955   5.717 1.23e-08 ***
## TEAM_PITCHING_HR -0.0872307  0.0294968  -2.957 0.003136 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.78 on 2261 degrees of freedom
## Multiple R-squared:  0.3377, Adjusted R-squared:  0.3339
## F-statistic: 88.68 on 13 and 2261 DF,  p-value: < 2.2e-16
```

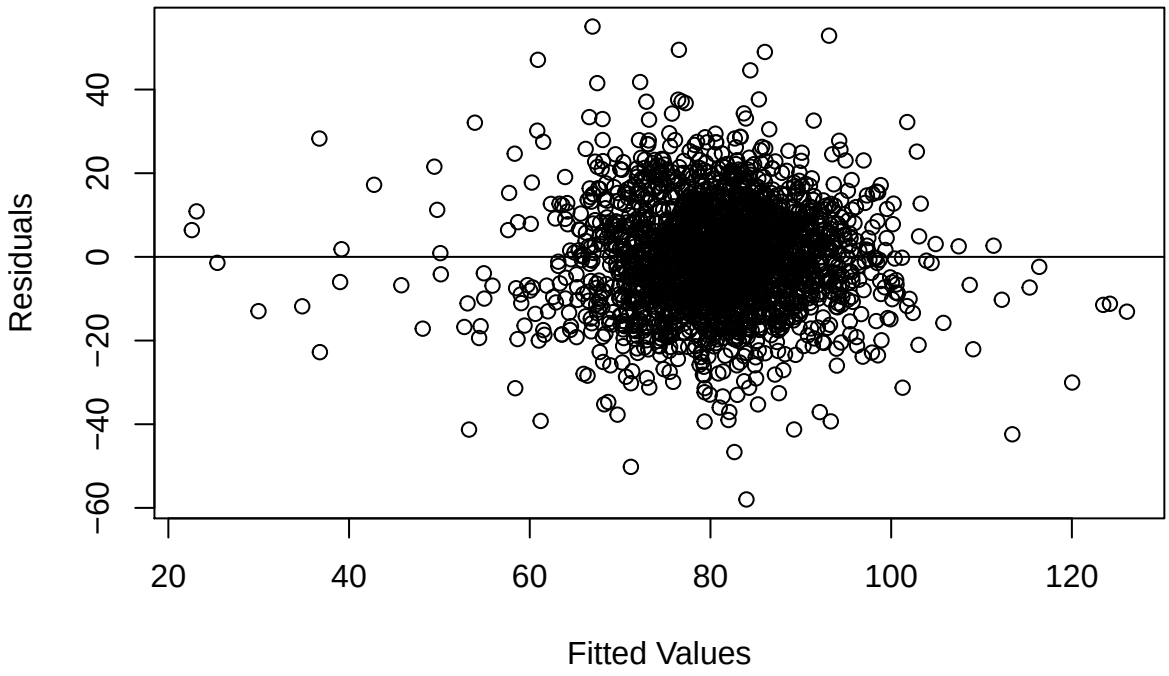
All variables used in this model have statistical significance at 0.01 level. The F-statistic is high with a p-value nearly 0 and, therefore, is significant. Median value of residuals is close to 0 and they are equally distributed. Standard errors are significantly smaller than estimated coefficients.

Only 4 variables are negatively correlated - strikeouts, double plays, errors, and home runs allowed. Remaining variables are positively correlated. Some correlation is counter-intuitive. For example, double plays are considered successful defensive moves and should increase the winning percentage. Similarly, allowing base hits should decrease winning percentage. The model indicates otherwise and there are probably there factors that influence these variables.

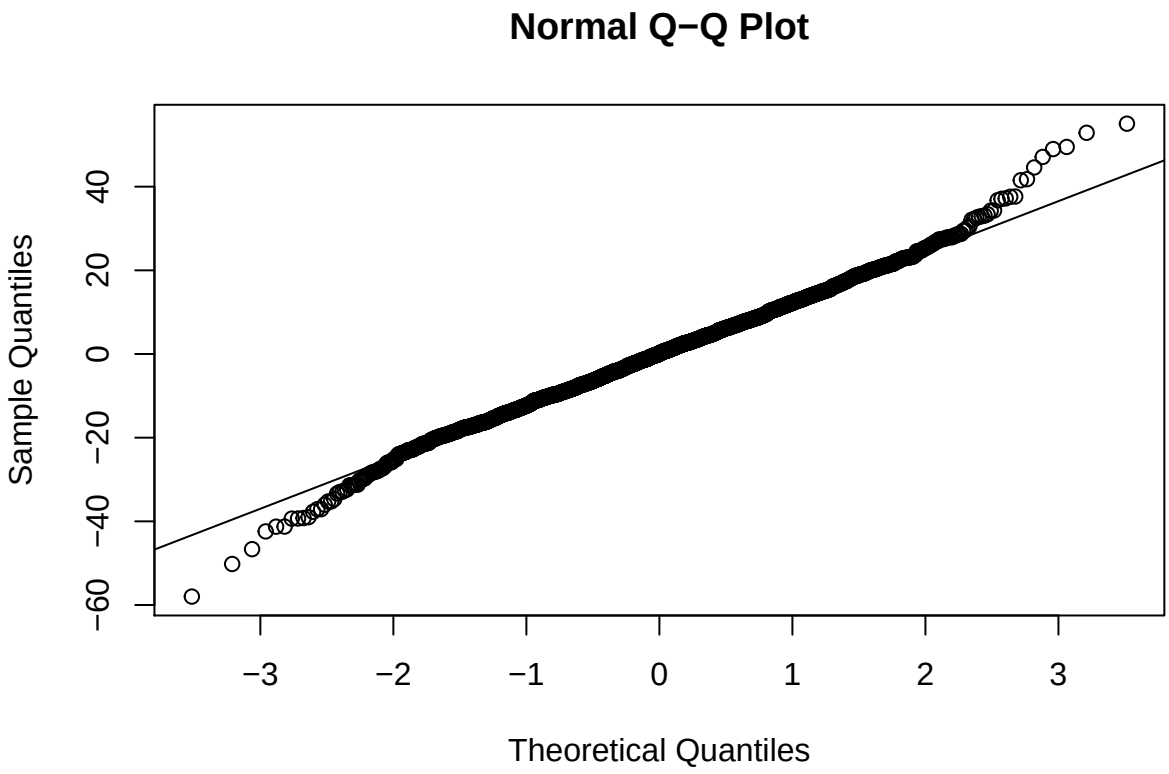
Consider residuals plotted against data index. There is no pattern.



Plotting fitted values against the residuals is more problematic. Although there is no pattern among residuals, there are some outliers and variability does not appear to be constant across the entire range.



Q-Q plot confirms that residuals are normally distributed.



Prediction

Using selected model and evaluation data (transformed similarly to training data), prediction table is as follows. It includes predicted number of wins along with confidence interval.

Index	Predicted Wins	CI Lower	CI Upper
9	62	60	64
10	65	63	66
14	74	73	76
47	87	86	89
60	68	65	72
63	74	72	77
74	85	83	87
83	76	74	77
98	70	69	72
120	73	72	74
123	69	67	70
135	82	80	84
138	81	80	83
140	82	80	84
151	85	83	87
153	77	76	79
171	74	72	75
184	79	77	80
193	75	73	77
213	91	89	92
217	82	80	83
226	84	82	85
230	80	79	81
241	71	70	73
291	82	81	84
294	88	87	89
300	42	37	46
348	74	73	75
350	83	81	86
357	74	72	76
367	90	89	92
368	85	84	87
372	82	81	84
382	83	82	85
388	80	79	81
396	86	85	88
398	76	75	76
403	90	89	92
407	83	80	87
410	92	90	93
412	83	81	84
414	92	90	93
436	16	10	23
440	109	106	112
476	96	94	99
479	94	92	96
481	99	97	101
501	76	75	77

Index	Predicted Wins	CI Lower	CI Upper
503	68	67	70
506	79	78	80
519	76	75	78
522	85	84	87
550	77	75	78
554	73	72	75
566	75	74	76
578	79	78	80
596	92	90	93
599	76	74	77
605	54	50	57
607	83	82	84
614	88	87	89
644	75	73	77
692	88	87	89
699	86	84	88
700	85	83	86
716	102	100	105
721	74	72	75
722	80	79	82
729	73	70	75
731	89	87	91
746	86	84	88
763	70	69	72
774	77	76	79
776	89	87	91
788	81	79	82
789	85	83	86
792	83	82	84
811	84	82	85
835	75	74	77
837	76	74	78
861	84	82	87
862	88	86	90
863	97	95	99
871	74	73	76
879	85	83	86
887	80	79	82
892	83	82	85
904	84	83	85
909	90	88	91
925	91	89	92
940	82	80	83
951	64	53	75
976	72	71	74
981	90	88	92
983	84	82	86
984	84	83	86
989	89	87	91
995	103	101	105
1000	87	85	89
1001	87	85	89

Index	Predicted Wins	CI Lower	CI Upper
1007	80	78	81
1016	73	72	75
1027	84	83	85
1033	84	82	85
1070	79	77	80
1081	74	72	77
1084	48	45	51
1098	77	76	79
1150	87	86	88
1160	51	49	54
1169	85	84	86
1172	86	84	88
1174	95	94	96
1176	92	91	93
1178	81	80	82
1184	78	77	80
1193	86	85	88
1196	81	80	82
1199	74	73	75
1207	79	77	81
1218	94	92	96
1223	63	61	65
1226	68	66	70
1227	63	61	66
1229	68	67	70
1241	88	86	89
1244	91	89	92
1246	77	75	78
1248	93	91	94
1249	92	90	93
1253	86	84	87
1261	80	78	81
1305	79	78	81
1314	85	84	87
1323	88	86	89
1328	77	74	80
1353	74	73	75
1363	77	76	78
1371	89	87	90
1372	81	80	82
1389	64	63	66
1393	77	75	79
1421	91	89	93
1431	72	71	74
1437	72	70	73
1442	71	70	72
1450	77	76	78
1463	79	78	80
1464	79	78	81
1470	83	82	84
1471	82	81	84
1484	81	80	82

Index	Predicted Wins	CI Lower	CI Upper
1495	51	41	60
1507	69	67	71
1514	77	76	78
1526	70	69	72
1549	90	88	92
1552	61	59	63
1556	92	90	94
1564	70	68	72
1585	104	102	106
1586	108	106	110
1590	94	92	95
1591	104	102	106
1592	98	95	100
1603	89	88	91
1612	82	80	83
1634	82	80	83
1645	73	72	74
1647	81	80	82
1673	89	87	91
1674	90	88	91
1687	80	79	82
1688	94	93	96
1700	82	81	84
1708	73	72	75
1713	77	76	79
1717	70	69	71
1721	74	73	75
1730	79	78	81
1737	89	87	91
1748	89	87	90
1749	86	85	87
1763	85	84	86
1768	72	64	80
1778	97	94	100
1780	81	78	83
1782	45	41	49
1784	60	57	63
1794	116	113	119
1803	68	67	70
1804	81	79	82
1819	76	74	77
1832	77	75	78
1833	79	77	81
1844	67	65	68
1847	77	76	78
1854	84	83	86
1855	79	78	80
1857	84	83	85
1864	75	73	77
1865	80	78	81
1869	74	72	75
1880	88	85	91

Index	Predicted Wins	CI Lower	CI Upper
1881	81	80	82
1882	84	82	85
1894	78	77	80
1896	78	76	79
1916	78	76	79
1918	75	74	77
1921	102	99	104
1926	92	90	94
1938	82	80	84
1979	64	63	66
1982	68	66	69
1987	83	82	85
1997	79	77	81
2004	94	93	96
2011	77	76	78
2015	79	78	80
2022	78	77	79
2025	74	72	76
2027	81	80	83
2031	73	71	75
2036	73	68	77
2066	75	74	76
2073	81	80	82
2087	79	78	81
2092	82	81	83
2125	65	63	68
2148	79	77	81
2162	93	92	95
2191	78	77	80
2203	89	88	91
2218	80	79	81
2221	75	74	76
2225	83	81	84
2232	78	76	79
2267	89	87	92
2291	72	71	74
2299	89	88	90
2317	86	85	88
2318	84	82	85
2353	82	80	83
2403	61	59	63
2411	87	86	89
2415	81	80	82
2424	85	84	86
2441	72	71	74
2464	84	83	86
2465	81	80	82
2472	62	59	65
2481	95	93	97
2487	19	12	27
2500	69	68	70
2501	77	75	78

Index	Predicted Wins	CI Lower	CI Upper
2520	83	81	84
2521	84	83	85
2525	77	75	79

APPENDIX: R Script

```
# Required libraries
library(dplyr)
library(ggplot2)
library(gridExtra)
library(knitr)
library(kableExtra)
library(Hmisc)

# Import data
bb <- read.csv("moneyball-training-data.csv")

# Basic statistic
nrow(bb); ncol(bb)
summary(bb)

# Get summary table
sumBB = data.frame(Variable = character(),
                   Min = integer(),
                   Median = integer(),
                   Mean = double(),
                   SD = double(),
                   Max = integer(),
                   Num_NAs = integer(),
                   Num_Zeros = integer())

for (i in 2:17) {
  sumBB <- rbind(sumBB, data.frame(Variable = colnames(bb)[i],
                                   Min = min(bb[,i], na.rm=TRUE),
                                   Median = median(bb[,i], na.rm=TRUE),
                                   Mean = mean(bb[,i], na.rm=TRUE),
                                   SD = sd(bb[,i], na.rm=TRUE),
                                   Max = max(bb[,i], na.rm=TRUE),
                                   Num_NAs = sum(is.na(bb[,i])),
                                   Num_Zeros = length(which(bb[,i]==0)))
                 )
}

# Exploratory plots (repeated for each variable)
kable(sumBB[sumBB[,1]=="TEAM_BASERUN_SB",2:8], row.names=FALSE)

# Boxplot
bp <- ggplot(bb, aes(x = 1, y = TEAM_BASERUN_SB)) +
  stat_boxplot(geom ='errorbar') + geom_boxplot() +
  xlab("Boxplot") + ylab("") + theme(axis.text.x=element_blank(),
                                     axis.ticks.x=element_blank())

# Density plot
hp <- ggplot(bb, aes(x = TEAM_BASERUN_SB)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white") +
  geom_density(alpha=.2, fill="#FF6666") + ylab("") + xlab("Density Plot with Mean") +
  geom_vline(aes(xintercept=mean(TEAM_BASERUN_SB, na.rm=TRUE)), color="red",
```

```
             linetype="dashed", size=1)

# Scatterplot
sp <- ggplot(data=bb, aes(x=TEAM_BASERUN_SB, y=TARGET_WINS)) +
  geom_point() + geom_smooth(method = "loess") +
  xlab("Scatterplot with Best Fit Line")

grid.arrange(bp, hp, sp, layout_matrix=rbind(c(1,2,2),c(1,3,3)))

# Correlation matrix
cm <- cor(bb, use="pairwise.complete.obs")
cm <- cm[2:17,2:17]
names <- c("Wins", "H", "2B", "3B", "HR", "BB", "SO", "SB", "CS", "HBP", "P-H",
           "P-HR", "P-BB", "P-SO", "E", "DP")
colnames(cm) <- names; rownames(cm) <- names
cm <- round(cm, 2)
cmout <- as.data.frame(cm) %>% mutate_all(function(x) {
  cell_spec(x, "html", color = ifelse(x>0.5 | x<(-0.5),"blue","black"))
  })
rownames(cmout) <- names
cmout %>%
  kable("html", escape = F, align = "c", row.names = TRUE) %>%
  kable_styling("striped", full_width = F)

bbBackup <- bb

# Remove observations with no target
bb <- bb[which(bb$TARGET_WINS!=0), ]

# Reset zero values
bb[which(bb$TEAM_BATTING_H==0),"TEAM_BATTING_H"] <- NA
bb[which(bb$TEAM_BATTING_2B==0),"TEAM_BATTING_2B"] <- NA
bb[which(bb$TEAM_BATTING_3B==0),"TEAM_BATTING_3B"] <- NA
bb[which(bb$TEAM_BATTING_HR==0),"TEAM_BATTING_HR"] <- NA
bb[which(bb$TEAM_BATTING_BB==0),"TEAM_BATTING_BB"] <- NA
bb[which(bb$TEAM_BATTING_SO==0),"TEAM_BATTING_SO"] <- NA
bb[which(bb$TEAM_BASERUN_SB==0),"TEAM_BASERUN_SB"] <- NA
bb[which(bb$TEAM_BASERUN_CS==0),"TEAM_BASERUN_CS"] <- NA
bb[which(bb$TEAM_FIELDING_E==0),"TEAM_FIELDING_E"] <- NA
bb[which(bb$TEAM_FIELDING_DP==0),"TEAM_FIELDING_DP"] <- NA
bb[which(bb$TEAM_PITCHING_BB==0),"TEAM_PITCHING_BB"] <- NA
bb[which(bb$TEAM_PITCHING_H==0),"TEAM_PITCHING_H"] <- NA
bb[which(bb$TEAM_PITCHING_HR==0),"TEAM_PITCHING_HR"] <- NA
bb[which(bb$TEAM_PITCHING_SO==0),"TEAM_PITCHING_SO"] <- NA

# Impute mimssing values
bbImpute <- aregImpute(~ 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_FIELDING_DP + TEAM_FIELDING_E + TEAM_PITCHING_BB +
                         TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_SO,
                       data = bb, n.impute = 10)
bbImpute
```

```
bbImpute$rsq

bbI <- impute.transcan(bbImpute, imputation=10, data=bb,
                       list.out=TRUE, pr=FALSE, check=FALSE)
bb$TEAM_BASERUN_SB <- bbI$TEAM_BASERUN_SB
bb$TEAM_BASERUN_CS <- bbI$TEAM_BASERUN_CS
bb$TEAM_BATTING_3B <- bbI$TEAM_BATTING_3B
bb$TEAM_BATTING_HR <- bbI$TEAM_BATTING_HR
bb$TEAM_BATTING_SO <- bbI$TEAM_BATTING_SO
bb$TEAM_FIELDING_DP <- bbI$TEAM_FIELDING_DP
bb$TEAM_PITCHING_HR <- bbI$TEAM_PITCHING_HR
bb$TEAM_PITCHING_SO <- bbI$TEAM_PITCHING_SO

# Adjust outliers
bb[which(bb$TEAM_PITCHING_SO>2500),"TEAM_PITCHING_SO"] <- 2500
bb[which(bb$TEAM_PITCHING_H>13000),"TEAM_PITCHING_H"] <- 13000
bb[which(bb$TEAM_PITCHING_BB>1100),"TEAM_PITCHING_BB"] <- 1100

# Creat singles
bb$TEAM_BATTING_S <- bb$TEAM_BATTING_H - bb$TEAM_BATTING_2B -
  bb$TEAM_BATTING_3B - bb$TEAM_BATTING_HR
summary(bb$TEAM_BATTING_S)

# Create log fielding error
bb$TEAM_FIELDING_E_LOG <- log(bb$TEAM_FIELDING_E)

# Model building
m1 <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_FIELDING_E, data=bb)
summary(m1)

m2 <- lm(TARGET_WINS ~ TEAM_BATTING_S + TEAM_BATTING_2B + TEAM_BATTING_3B +
           TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_FIELDING_E, data=bb)
summary(m2)
m2b <- lm(TARGET_WINS ~ TEAM_BATTING_S + TEAM_BATTING_2B + TEAM_BATTING_3B +
            TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_FIELDING_E_LOG, data=bb)
summary(m2b)

m3 <- lm(TARGET_WINS ~ TEAM_BATTING_SO:TEAM_BATTING_H + TEAM_BATTING_BB:TEAM_BATTING_H +
           TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_PITCHING_HR +
           TEAM_FIELDING_E_LOG, data=bb)
summary(m3)

m4 <- lm(TARGET_WINS ~ TEAM_BATTING_S + TEAM_BATTING_2B + TEAM_BATTING_3B +
           TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB +
           TEAM_BASERUN_CS + TEAM_FIELDING_DP + TEAM_FIELDING_E + TEAM_PITCHING_BB +
           TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_PITCHING_HR, data=bb)
summary(m4)
m4 <- lm(TARGET_WINS ~ TEAM_BATTING_S + TEAM_BATTING_2B + TEAM_BATTING_3B +
           TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB +
           TEAM_FIELDING_DP + TEAM_FIELDING_E + TEAM_PITCHING_BB +
           TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_PITCHING_HR, data=bb)
summary(m4)
```

```
m5 <- lm(TARGET_WINS ~ TEAM_BATTING_S  + TEAM_BATTING_3B +
           TEAM_BATTING_HR + TEAM_BASERUN_SB  +
           TEAM_FIELDING_E_LOG*TEAM_PITCHING_H, data=bb)
summary(m5)

# Residuals plots
plot(m4$residuals, ylab="Residuals")
abline(h=0)

plot(m4$fitted.values, m4$residuals, xlab="Fitted Values", ylab="Residuals")
abline(h=0)

qqnorm(m4$residuals)
qqline(m4$residuals)

# Test data for prediction
bbTest <- read.csv("moneyball-evaluation-data.csv")

bbTest[which(bbTest$TEAM_BATTING_H==0),"TEAM_BATTING_H"] <- NA
bbTest[which(bbTest$TEAM_BATTING_2B==0),"TEAM_BATTING_2B"] <- NA
bbTest[which(bbTest$TEAM_BATTING_3B==0),"TEAM_BATTING_3B"] <- NA
bbTest[which(bbTest$TEAM_BATTING_HR==0),"TEAM_BATTING_HR"] <- NA
bbTest[which(bbTest$TEAM_BATTING_BB==0),"TEAM_BATTING_BB"] <- NA
bbTest[which(bbTest$TEAM_BATTING_SO==0),"TEAM_BATTING_SO"] <- NA
bbTest[which(bbTest$TEAM_BASERUN_SB==0),"TEAM_BASERUN_SB"] <- NA
bbTest[which(bbTest$TEAM_BASERUN_CS==0),"TEAM_BASERUN_CS"] <- NA
bbTest[which(bbTest$TEAM_FIELDING_E==0),"TEAM_FIELDING_E"] <- NA
bbTest[which(bbTest$TEAM_FIELDING_DP==0),"TEAM_FIELDING_DP"] <- NA
bbTest[which(bbTest$TEAM_PITCHING_BB==0),"TEAM_PITCHING_BB"] <- NA
bbTest[which(bbTest$TEAM_PITCHING_H==0),"TEAM_PITCHING_H"] <- NA
bbTest[which(bbTest$TEAM_PITCHING_HR==0),"TEAM_PITCHING_HR"] <- NA
bbTest[which(bbTest$TEAM_PITCHING_SO==0),"TEAM_PITCHING_SO"] <- NA

# Impute mimssing values
bbImpute <- aregImpute(~ 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_FIELDING_DP +
                         TEAM_FIELDING_E + TEAM_PITCHING_BB + TEAM_PITCHING_H +
                         TEAM_PITCHING_HR + TEAM_PITCHING_SO,
                       data = bbTest, n.impute = 10)
bbImpute
bbImpute$rsq

bbI <- impute.transcan(bbImpute, imputation=10, data=bbTest,
                       list.out=TRUE, pr=FALSE, check=FALSE)
bbTest$TEAM_BATTING_HR <- bbI$TEAM_BATTING_HR
bbTest$TEAM_BATTING_SO <- bbI$TEAM_BATTING_SO
bbTest$TEAM_BASERUN_SB <- bbI$TEAM_BASERUN_SB
bbTest$TEAM_BASERUN_CS <- bbI$TEAM_BASERUN_CS
bbTest$TEAM_FIELDING_DP <- bbI$TEAM_FIELDING_DP
bbTest$TEAM_PITCHING_HR <- bbI$TEAM_PITCHING_HR
bbTest$TEAM_PITCHING_SO <- bbI$TEAM_PITCHING_SO
```

```r
# Adjust outliers
bbTest[which(bbTest$TEAM_PITCHING_SO>2500),"TEAM_PITCHING_SO"] <- 2500
bbTest[which(bbTest$TEAM_PITCHING_H>13000),"TEAM_PITCHING_H"] <- 13000
bbTest[which(bbTest$TEAM_PITCHING_BB>1100),"TEAM_PITCHING_BB"] <- 1100

bbTest$TEAM_BATTING_S <- bbTest$TEAM_BATTING_H - bbTest$TEAM_BATTING_2B -
  bbTest$TEAM_BATTING_3B - bbTest$TEAM_BATTING_HR

bbTest$PREDICT_WIN <- predict(m4, newdata=bbTest, interval="confidence")

bbPredict <- cbind(bbTest$INDEX, bbTest$PREDICT_WIN[, 1], bbTest$PREDICT_WIN[, 2],
                   bbTest$PREDICT_WIN[, 3])
colnames(bbPredict) <- c("Index", "Predicted Wins", "CI Lower", "CI Upper")
round(bbPredict,0)
```