Homework #1: Predicting Major League Baseball Wins

CUNY SPS DATA 621 Spring 2019

Group # 4

February 27, 2019

Contents

1 Objective

The the goal of this report is to develop a model which can accurately predict the number of wins of of a Major League Baseball (MLB) team based on historical performance.

2 Data exploration

2.1 Data dictionary

The table below table below describes the variables in the dataset.

variable name	definition	theoretical effect
INDEX	Identification Variable (do not use)	None
TARGET_WINS	Number of wins	outcome variable
BATTING_H	Base Hits by batters (1B,2B,3B,HR)	Positive Impact on Wins
BATTING_2B	Doubles by batters (2B)	Positive Impact on Wins
BATTING_3B	Triples by batters (3B)	Positive Impact on Wins
BATTING_HR	Homeruns by batters (4B)	Positive Impact on Wins
BATTING_BB	Walks by batters	Positive Impact on Wins
BATTING_HBP	Batters hit by pitch (get a free base)	Positive Impact on Wins
BATTING_SO	Strikeouts by batters	Negative Impact on Wins
BASERUN_SB	Stolen bases	Positive Impact on Wins
BASERUN_CS	Caught stealing	Negative Impact on Wins
FIELDING_E	Errors	Negative Impact on Wins
FIELDING_DP	Double Plays	Positive Impact on Wins
PITCHING_BB	Walks allowed	Negative Impact on Wins
PITCHING_H	Hits allowed	Negative Impact on Wins
PITCHING_HR	Homeruns allowed	Negative Impact on Wins
PITCHING_SO	Strikeouts by pitchers	Positive Impact on Wins

2.2 Summary statistics

The training data set contains 2276 observations and 17 variables including the response variable, `TARGET_WINS`. Of these columns all are numeric .

In the following tables the summary statistics for all of the variables in the dataset. They have been broken up into two tables for ease of reading. While some aspects of the distribution of the variables are easier seen in visualization. Two notable statistics are that `PITCHING_SO` has a strongly positive skew which may need to be addressed before developing a model; other predictors show skew and kurtosis, but to a lesser degree. Also, `BATTING_HBP` has a considerable amount of missing data.

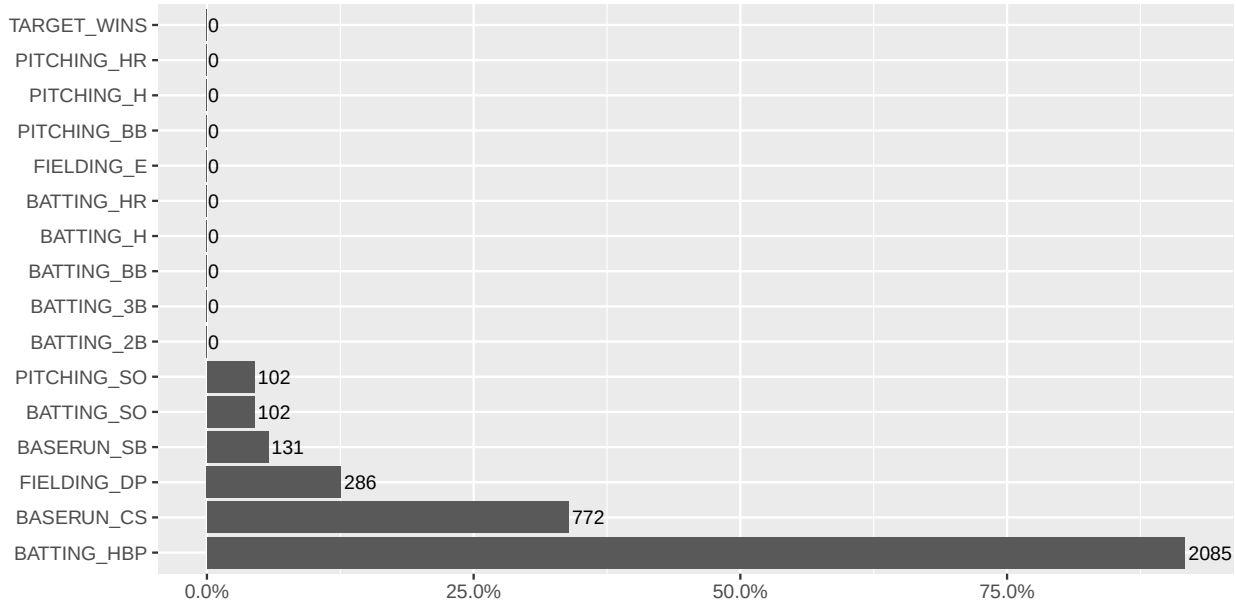
metric	min	Q1	mean	median	Q3	max
BASERUN_CS	0	38	52.8	49	62	201
BASERUN_SB	0	66	124.8	101	156	697
BATTING_2B	69	208	241.2	238	273	458
BATTING_3B	0	34	55.25	47	72	223
BATTING_BB	0	451	501.6	512	580	878
BATTING_H	891	1383	1469	1454	1537	2554
BATTING_HBP	29	50.5	59.36	58	67	95
BATTING_HR	0	42	99.61	102	147	264
BATTING_SO	0	548	735.6	750	930	1399
FIELDING_DP	52	131	146.4	149	164	228
FIELDING_E	65	127	246.5	159	249.2	1898
PITCHING_BB	0	476	553	536.5	611	3645
PITCHING_H	1137	1419	1779	1518	1682	30132
PITCHING_HR	0	50	105.7	107	150	343
PITCHING_SO	0	615	817.7	813.5	968	19278
TARGET_WINS	0	71	80.79	82	92	146

metric	missing	sd	mad	skewness	kurtosis
BASERUN_CS	772	22.96	17.79	1.976	7.62
BASERUN_SB	131	87.79	60.79	1.972	5.49
BATTING_2B	0	46.8	47.44	0.2151	0.006161
BATTING_3B	0	27.94	23.72	1.109	1.503
BATTING_BB	0	122.7	94.89	-1.026	2.183
BATTING_H	0	144.6	114.2	1.571	7.279
BATTING_HBP	2085	12.97	11.86	0.3186	-0.112
BATTING_HR	0	60.55	78.58	0.186	-0.9631
BATTING_SO	102	248.5	284.7	-0.2978	-0.3208
FIELDING_DP	286	26.23	23.72	-0.3889	0.1817
FIELDING_E	0	227.8	62.27	2.99	10.97
PITCHING_BB	0	166.4	98.59	6.744	96.97
PITCHING_H	0	1407	174.9	10.33	141.8
PITCHING_HR	0	61.3	74.13	0.2878	-0.6046
PITCHING_SO	102	553.1	257.2	22.17	671.2
TARGET_WINS	0	15.75	14.83	-0.3987	1.027

2.3 Visualizations

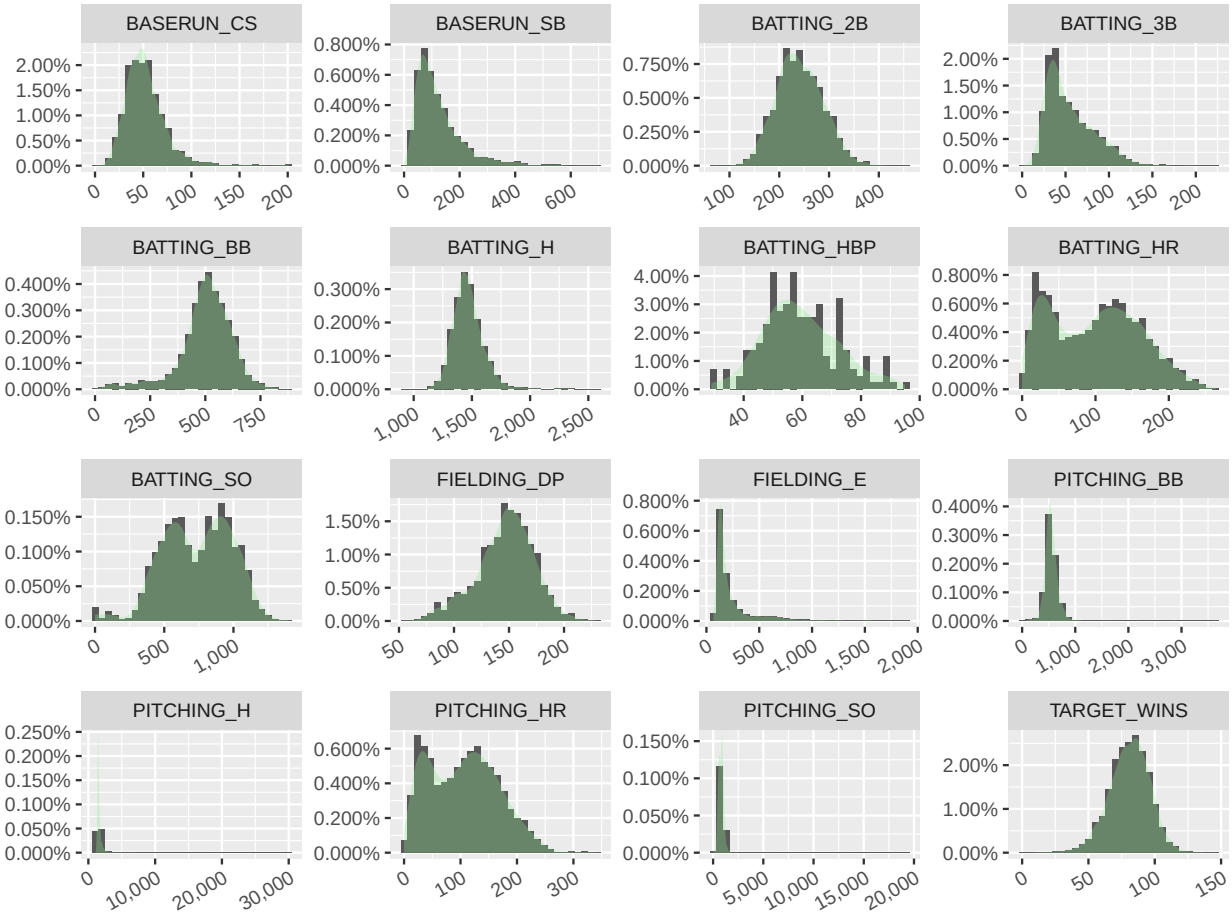
2.3.1 Missing data

As mentioned in review of the summary statistics `BATTING_HBP` has approximately 92% of the data missing and `BASERUN_CS` has roughly 34%. It is not immediately apparent whether these missing values are not applicable or actually missing. If it is the latter the proportion of missing data for these two variables is too large to reasonably consider addressing through substitution or imputation.

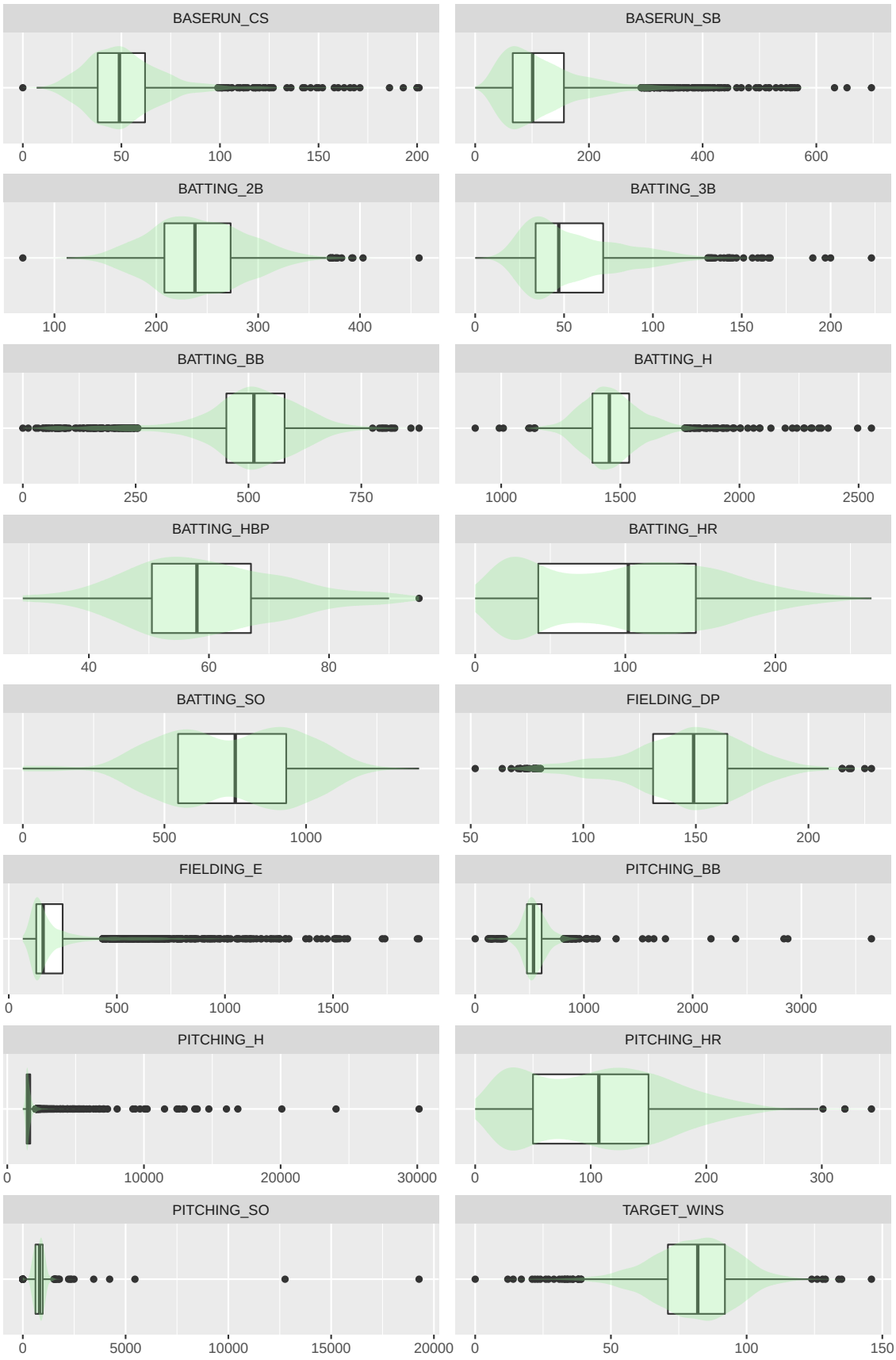


2.3.2 Univariate distributions

The plot below of a density plot overlayed on a histogram for all variables shows that more variables than those identified using the skewness statistics appear to have a skewed distribution. Some like `BASERUN_SB` may benefit from transformation, however other such as `BATTING_2B` may not require transformation as the base size of the data is sufficient to allow for some deviation from a normal distribution.

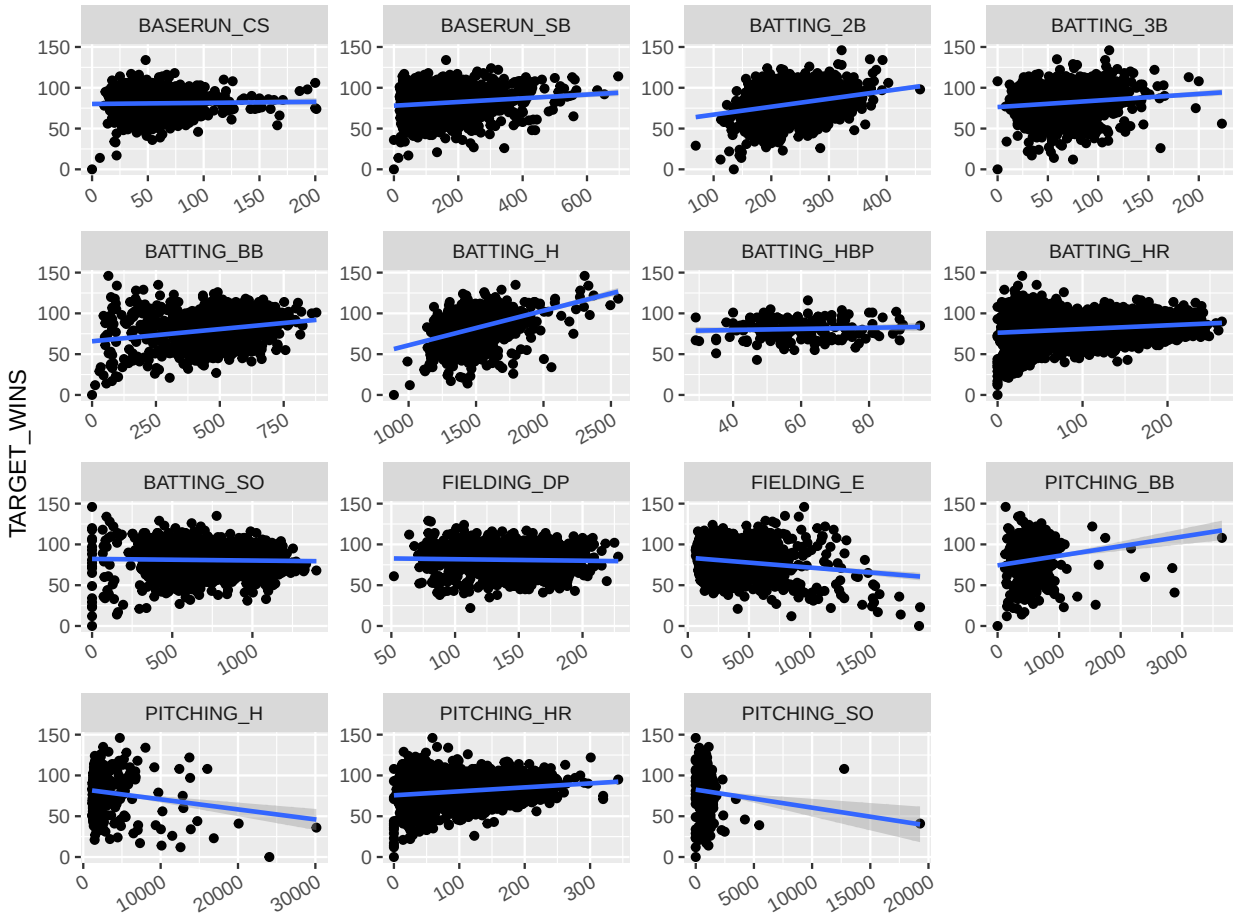


The histogram and density show that some of the predictors have long tails indicative of outliers. The plot below shows a boxplot and violin plot for each variable. Many of the variables appear to have outliers with predictors like `FIELDING_E` and `PITCHING_H` appear to have the most potential outliers. How these outliers will be handled will be dependent on a mix of reference data and statistical techniques.


BASERUN_CS
0
50
100
150
200
BASERUN_SB
0
200
400
600
BATTING_2B
100
200
300
400
BATTING_3B
0
50
100
150
200
BATTING_BB
0
250
500
750
BATTING_H
1000
1500
2000
2500
BATTING_HBP
40
60
80
BATTING_HR
0
100
200
BATTING_SO
0
500
1000
FIELDING_DP
50
100
150
200
FIELDING_E
0
500
1000
1500
PITCHING_BB
0
1000
2000
3000
PITCHING_H
0
10000
20000
30000
PITCHING_HR
0
100
200
300
PITCHING_SO
0
5000
10000
15000
20000
TARGET_WINS
0
50
100
150

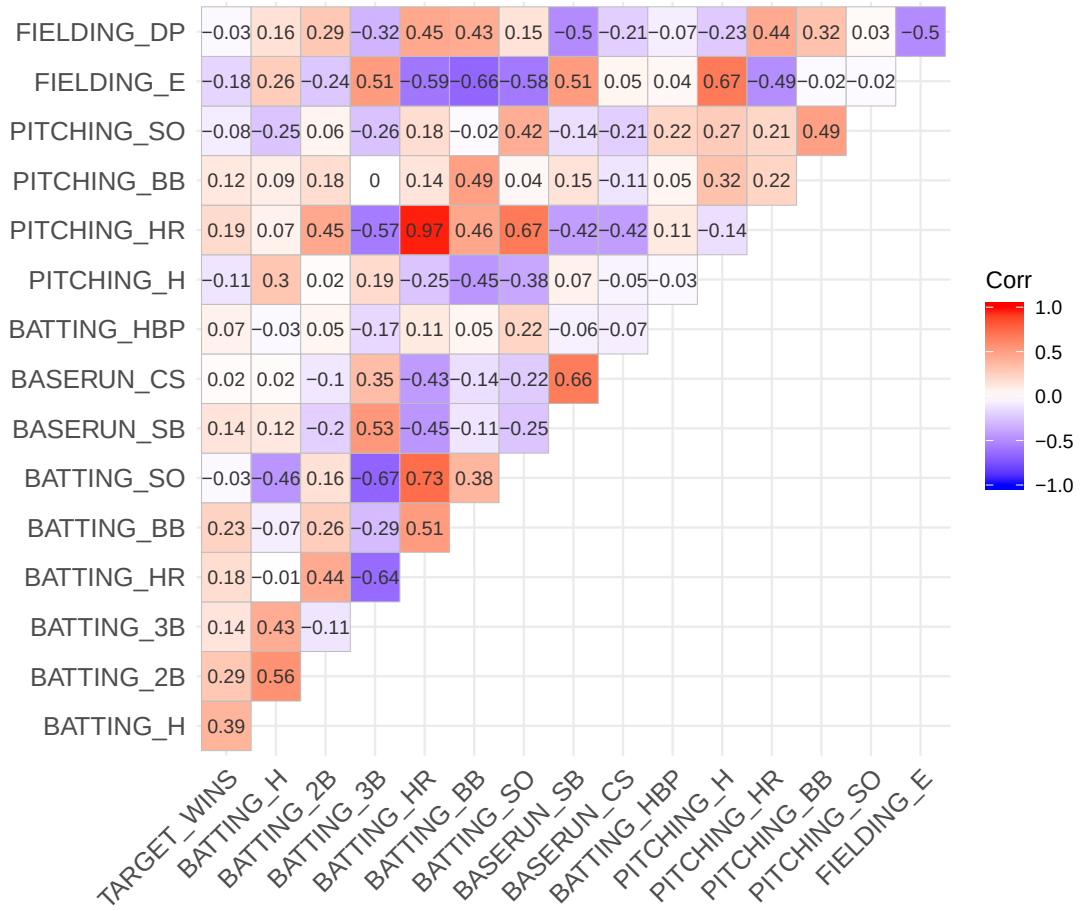
2.3.3 Bivariate relationships

The plot below shows scatter plots of all the response variable against each predictor in the data overlayed with a regression line to help visualize the relationship. Since the univariate summary statistics and visualizations showed that nor all predictors are reasonably normal in their distribution caution needs to be employed when making any inferences. Comparing the apparent relationships to the theoretical impact in the data dictionary shows that in the untransformed data most of theoretical assumptions align with the observations in the data; `PITCH_HR` is a notable exception which may suggest relationships among the predictors.



2.3.3.1 Correlation matrix

The scatterplot matrix below shows the relationships, Pearson Correlation Coefficient, between all variables in the data. The first column which represents the relationships between the response variable and each predictor further supports the observations made from the scatterplots. The matrix also shows that there are relationships between the some of the predictors as well. The strongest correlation is between `PITCHING_HR` and `BATTING_HR` which makes sense a pitcher would need to allow a homerun for a batter to hit a homerun.



3 Data preparation

Before modeling can be done, the issues identified during the data exploration namely creating predictors for information not explicitly presented in the data, non-normal distributions and missing data need to be addressed.

3.1 Variable creation

One predictor which was not explicitly described with he current predictors were single base hits by batters. This has been computed by calculating the difference in `BATTING_H` each of the sum of each of the base hits and stored as `BATTING_1B`.

3.2 Variable removal

For missing data, `BATTER_HBP` there is too much data missing to reasonably attempt substitution or other imputation techniques and will be dropped. The other predictors with missing data have enough data that imputation is possible.

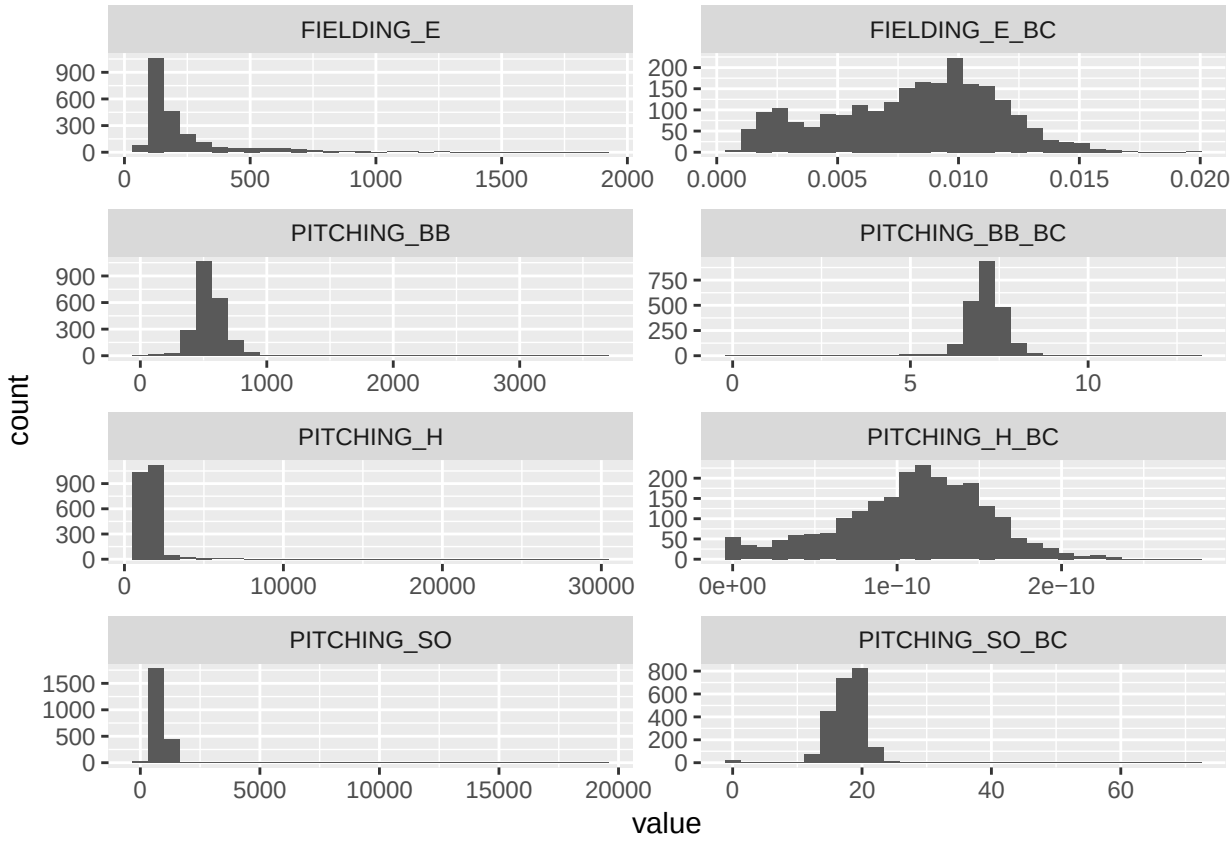
The two predictors `BATTING_HR` and `PITCHING_HR` have a near perfect Pearson correlation coefficient. Given this strong relationship including both in a model would not improve prediction and one can be dropped – `PITCHING_HR` has been dropped.

3.3 Imputation

Of the three predictors remaining with missing values two of them have a relatively small portion of the observations missing. For these two single imputation substituting the median for the missing values would be sufficient. However `FIELDING_DP` has enough missing observations that single imputation would likely be detrimental to the explanatory capability of the predictor that another technique should be employed. After reviewing a few methods of multiple imputation Multiple Imputation Chained Equations (MICE) was selected for its strength in handling imputation for observations with more than one predictor missing.

3.4 Transformations

The data contains enough observations that some deviation from normal distributions are acceptable. However for the predictors which are strongly skewed such as `FIELDING_E` a box-cox transformation has been applied.



3.5 Outliers

Determining whether data is an outlier can be quite nuanced. Beside purely statistical approaches decision about outliers can depend on knowledge of the data both in how it was collected or knowledge about the specific subject matter. Typically a combination of both approaches would be employed, however given limited knowledge of baseball, known rules changes which would impact comparison of statistics, and a limited time frame to gain practical familiarity only purely statistical approaches have been employed.

3.5.1 Statistical approach

The season and team statistics provided in the data have been scaled to reflect the current number of games and team in a season, however it is not clear whether or not any adjustments have been made for rule changes which could potentially impact results. Since this is unknown outliers will be determined using the Median Absolute Deviation (MAD); as a metric of central tendency, median is less effected by extreme values than mean. Anything that exceeds three MAD of the median will be considered an outlier and dropped.

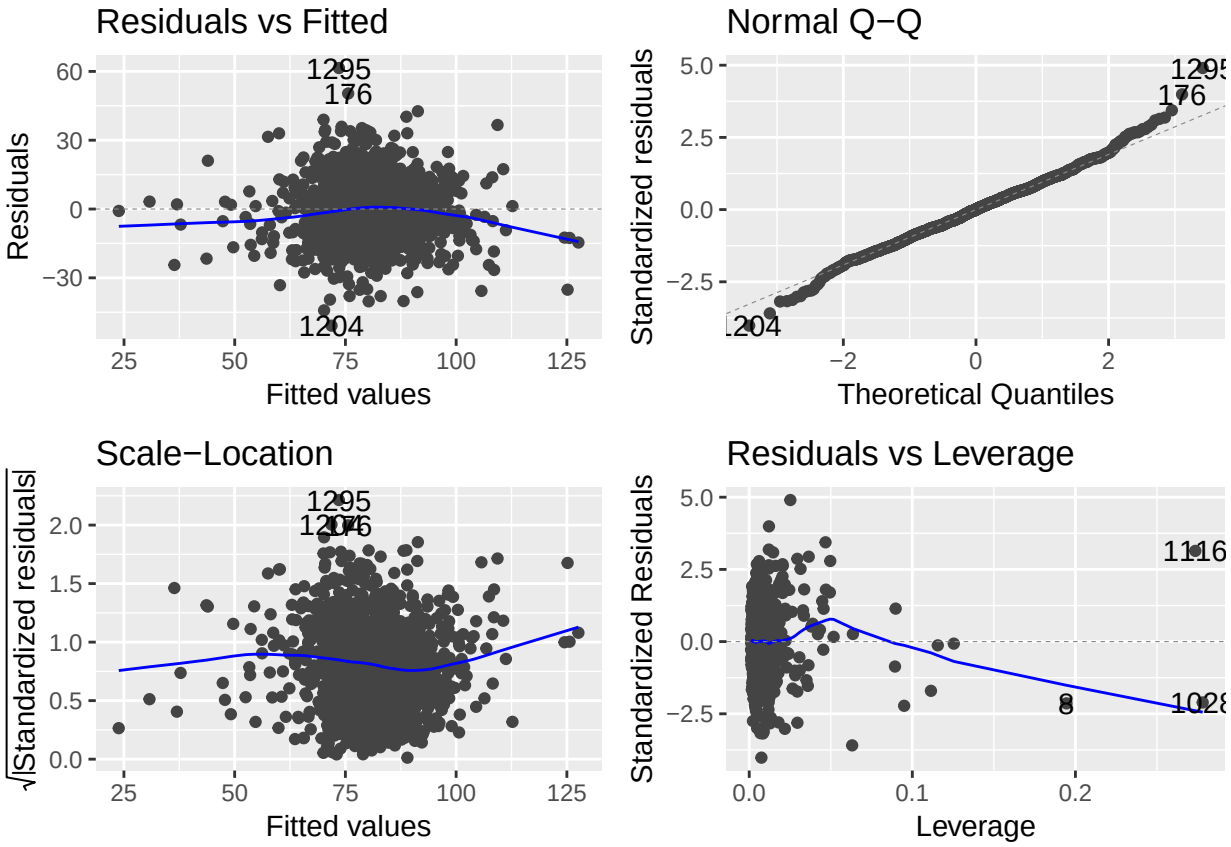
4 Build models

For purposes of model development and evaluation the training data is split 70/30 with 70% of the data being used for model development, and 30% being used to evaluate the model.

4.1 Model 1: imputation only

This model serves as a baseline for evaluating the benefits of variable transformation and outlier removal on prediction accuracy. Not all predictors were statistically significant so a stepwise regression tuning by the Akike Information Criterion (AIC) was employed for predictor selection. The adjusted r-squared indicates that the model explains just over 34% of the variance in the training data. Looking at the model evaluation plots, Normal Q-Q plot shows that the model struggles to predict `TARGET_WINS` at both tails of the distribution. The Residuals vs. Leverage plot suggests that some observations may be influencing the regression. Transforming some of the non-normally distributed predictors or removing outliers may improve the model

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_H + BATTING_2B + BATTING_3B +
##     BATTING_HR + BATTING_SO + BASERUN_SB + PITCHING_H + PITCHING_BB +
##     FIELDING_E + FIELDING_DP, data = dfMDLTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.858  -8.220  -0.002   8.140  61.475
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.2820163  6.1414930   5.908 4.24e-09 ***
## BATTING_H    0.0423825  0.0044968   9.425  < 2e-16 ***
## BATTING_2B  -0.0183301  0.0107732  -1.701  0.08905 .
## BATTING_3B   0.0360046  0.0195125   1.845  0.06519 .
## BATTING_HR   0.0858141  0.0110683   7.753 1.59e-14 ***
## BATTING_SO  -0.0128547  0.0027235  -4.720 2.57e-06 ***
## BASERUN_SB   0.0521845  0.0052062  10.024  < 2e-16 ***
## PITCHING_H   0.0012706  0.0004984   2.549  0.01088 *
## PITCHING_BB  0.0075701  0.0025122   3.013  0.00263 **
## FIELDING_E  -0.0436684  0.0033177 -13.162  < 2e-16 ***
## FIELDING_DP -0.1233687  0.0153051  -8.061 1.48e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.7 on 1582 degrees of freedom
## Multiple R-squared:  0.3488, Adjusted R-squared:  0.3447
## F-statistic: 84.73 on 10 and 1582 DF,  p-value: < 2.2e-16
```

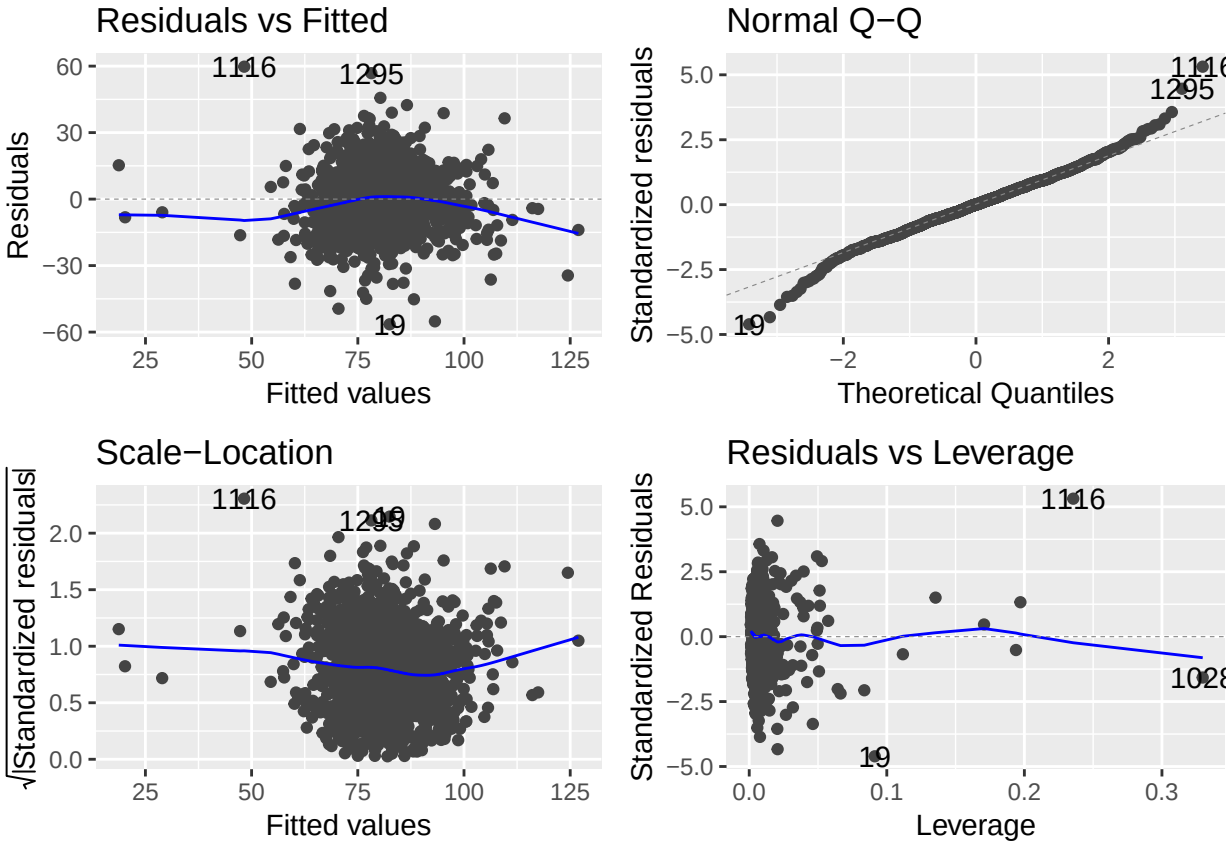


4.2 Model 2: imputation and transformation

Box-Cox transformations applied to the most heavily skewed predictors and then the same stepwise AIC mechanism was employed for predictor selection. The plots show that this model does not have the same issue with leverage that the first model had, but the adjusted r-squared is lower. As with the first model the Normal Q-Q plot suggests that the model struggles to accurately predict at the tails of the distribution. Removing outliers may help.

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_H + BATTING_3B + BATTING_HR +
##     BATTING_BB + BATTING_SO + BASERUN_SB + FIELDING_DP + FIELDING_E_BC +
##     PITCHING_H_BC + PITCHING_SO_BC + PITCHING_BB_BC, data = dfMDLTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.490  -7.793   0.060   8.328  59.756
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.514e+00  9.720e+00   0.464 0.642397
## BATTING_H       4.203e-02  4.101e-03  10.247  < 2e-16 ***
## BATTING_3B      8.255e-02  2.014e-02   4.099 4.36e-05 ***
## BATTING_HR      6.491e-02  1.196e-02   5.427 6.63e-08 ***
## BATTING_BB      3.426e-02  6.142e-03   5.578 2.86e-08 ***
```

```
## BATTING_SO     -3.046e-02  4.131e-03  -7.375 2.64e-13 ***
## BASERUN_SB      3.952e-02  4.902e-03   8.062 1.46e-15 ***
## FIELDING_DP    -1.232e-01  1.550e-02  -7.944 3.69e-15 ***
## FIELDING_E_BC   2.078e+03  1.997e+02  10.404  < 2e-16 ***
## PITCHING_H_BC   5.304e+10  1.475e+10   3.596 0.000333 ***
## PITCHING_SO_BC  1.020e+00  2.145e-01   4.754 2.17e-06 ***
## PITCHING_BB_BC -2.852e+00  1.282e+00  -2.225 0.026246 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.86 on 1581 degrees of freedom
## Multiple R-squared:  0.3328, Adjusted R-squared:  0.3282
## F-statistic: 71.69 on 11 and 1581 DF,  p-value: < 2.2e-16
```

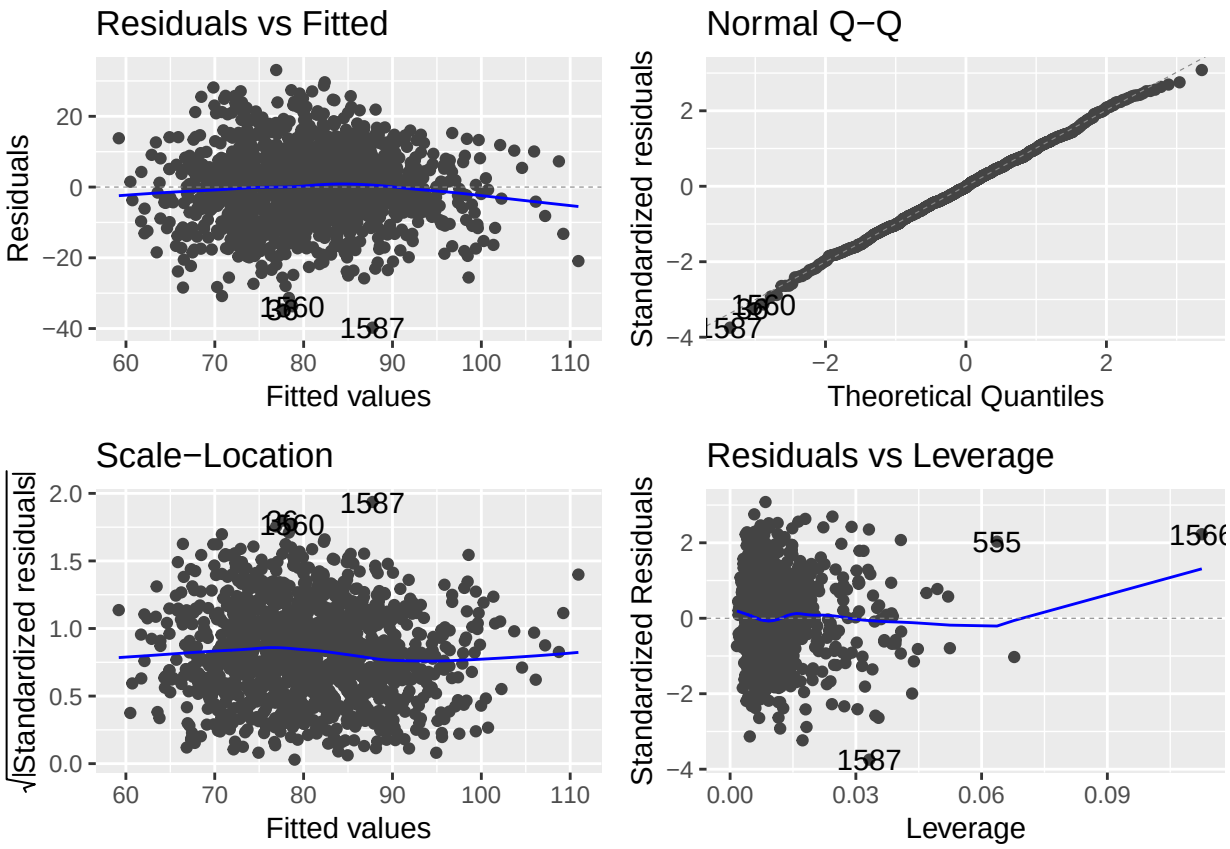


4.3 Model 3: imputation, transformation, and outlier removal

As mentioned previously observations which were more than 3 MAD away from the median were considered to be potential outliers. The adjusted r-squared for this model is the best of the three and reviewing the plots the residuals appear more normal than model one or two.

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_H + BATTING_2B + BATTING_3B +
##     BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB + FIELDING_DP +
##     FIELDING_E_BC + PITCHING_H_BC + PITCHING_SO_BC + PITCHING_BB_BC,
##     data = dfMDLTrain[!(dfMDLTrain$INDEX %in% outlierIndex),
```

```
##         ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.755  -7.160  -0.268   7.370  33.087
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.286e+01  3.497e+01  -0.939 0.347665
## BATTING_H       5.883e-02  1.009e-02   5.830 7.05e-09 ***
## BATTING_2B     -4.137e-02  1.136e-02  -3.641 0.000282 ***
## BATTING_3B      1.597e-01  2.511e-02   6.360 2.82e-10 ***
## BATTING_HR      8.526e-02  1.169e-02   7.291 5.41e-13 ***
## BATTING_BB      9.176e-02  2.003e-02   4.582 5.07e-06 ***
## BATTING_SO     -8.064e-02  9.931e-03  -8.120 1.10e-15 ***
## BASERUN_SB      4.552e-02  7.069e-03   6.440 1.69e-10 ***
## FIELDING_DP    -1.276e-01  1.517e-02  -8.408  < 2e-16 ***
## FIELDING_E_BC   2.132e+03  1.913e+02  11.146  < 2e-16 ***
## PITCHING_H_BC   9.830e+10  3.463e+10   2.838 0.004608 **
## PITCHING_SO_BC  6.073e+00  9.426e-01   6.442 1.67e-10 ***
## PITCHING_BB_BC -1.272e+01  4.578e+00  -2.779 0.005535 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.79 on 1261 degrees of freedom
## Multiple R-squared:  0.3653, Adjusted R-squared:  0.3592
## F-statistic: 60.47 on 12 and 1261 DF,  p-value: < 2.2e-16
```



4.4 Model 4: imputation and backward-elimination

This model utilizes cross-validation and backward elimination as an alternative method of feature selection. The underlying data is the same as model 1.

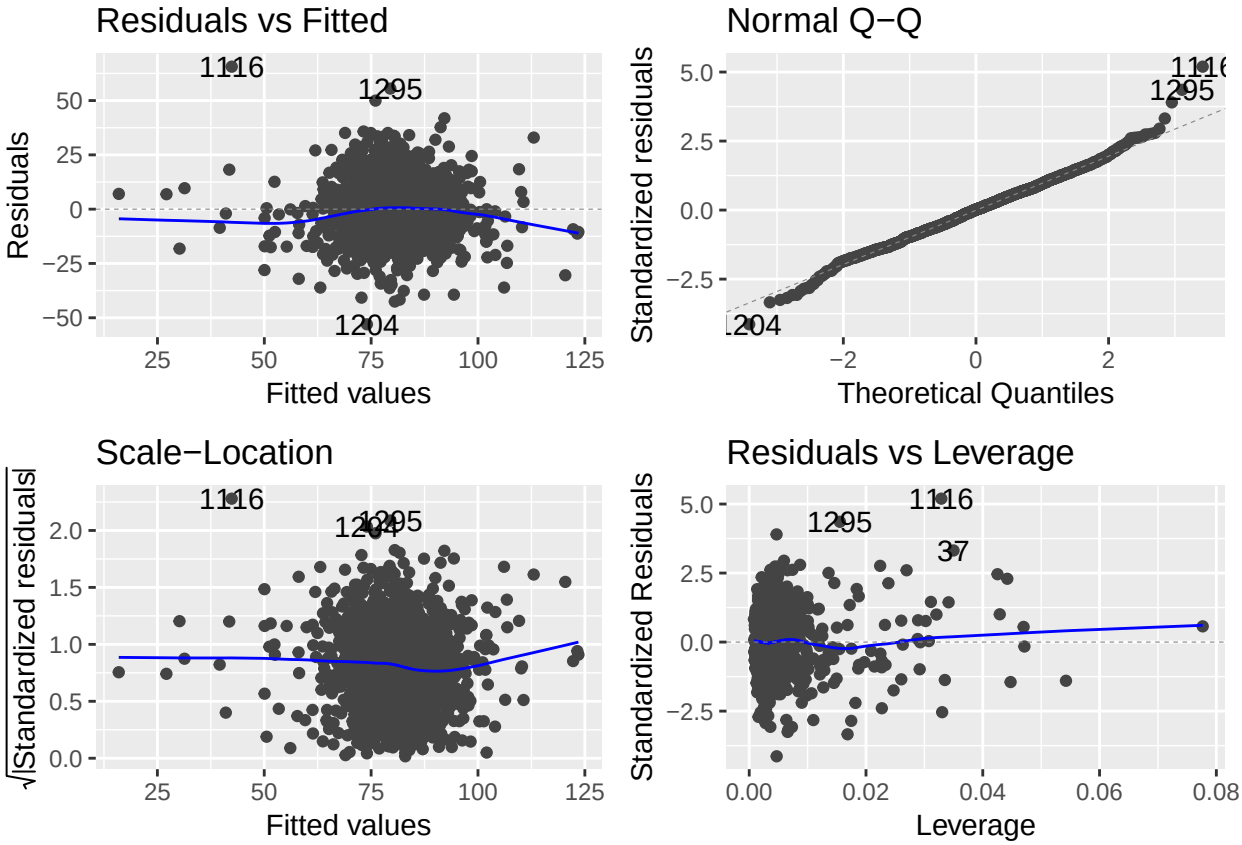
nvmax	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	14.57	0.1509	11.31	0.9263	0.1024	0.4835
2	13.96	0.2078	10.98	0.7704	0.05293	0.4582
3	13.21	0.2912	10.47	0.5115	0.04413	0.4
4	13.1	0.3034	10.36	0.5402	0.04346	0.3568
5	12.98	0.3165	10.26	0.5252	0.04371	0.316
6	12.95	0.3194	10.21	0.6259	0.04053	0.3312
7	13.05	0.3115	10.2	0.8194	0.06291	0.4687
8	13.03	0.3149	10.2	0.7729	0.06592	0.4662
9	13	0.3173	10.16	0.7742	0.06206	0.4574
10	12.97	0.32	10.13	0.779	0.06264	0.4645
11	12.97	0.32	10.12	0.7924	0.06294	0.4567
12	13.02	0.3153	10.14	0.8417	0.06179	0.4666
13	13.06	0.3119	10.16	0.8686	0.06483	0.4655
14	13.06	0.3119	10.16	0.8686	0.06483	0.4655

	nvmax
6	6

```
## Subset selection object
## 13 Variables  (and intercept)
##             Forced in Forced out
## BATTING_H       FALSE      FALSE
## BATTING_2B      FALSE      FALSE
## BATTING_3B      FALSE      FALSE
## BATTING_HR      FALSE      FALSE
## BATTING_BB      FALSE      FALSE
## BATTING_SO      FALSE      FALSE
## BASERUN_SB      FALSE      FALSE
## BASERUN_CS      FALSE      FALSE
## PITCHING_H      FALSE      FALSE
## PITCHING_BB     FALSE      FALSE
## PITCHING_SO     FALSE      FALSE
## FIELDING_E      FALSE      FALSE
## FIELDING_DP     FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: backward
##          BATTING_H BATTING_2B BATTING_3B BATTING_HR BATTING_BB BATTING_SO
## 1  ( 1 ) "*"       " "        " "        " "        " "        " "
## 2  ( 1 ) "*"       " "        " "        " "        " "        " "
## 3  ( 1 ) "*"       " "        " "        " "        " "        " "
## 4  ( 1 ) "*"       " "        " "        " "        " "        " "
## 5  ( 1 ) "*"       " "        " "        "*"        " "        " "
## 6  ( 1 ) "*"       " "        " "        "*"        " "        "*"
##          BASERUN_SB BASERUN_CS PITCHING_H PITCHING_BB PITCHING_SO
## 1  ( 1 ) " "        " "        " "        " "          " "
## 2  ( 1 ) " "        " "        " "        " "          " "
## 3  ( 1 ) "*"        " "        " "        " "          " "
## 4  ( 1 ) "*"        " "        " "        " "          " "
## 5  ( 1 ) "*"        " "        " "        " "          " "
## 6  ( 1 ) "*"        " "        " "        " "          " "
##          FIELDING_E FIELDING_DP
## 1  ( 1 ) " "        " "
## 2  ( 1 ) "*"        " "
## 3  ( 1 ) "*"        " "
## 4  ( 1 ) "*"        "*"
## 5  ( 1 ) "*"        "*"
## 6  ( 1 ) "*"        "*"

##
## Call:
## lm(formula = TARGET_WINS ~ FIELDING_DP + FIELDING_E + BASERUN_SB +
##     BATTING_SO + BATTING_HR + BATTING_H, data = dfMDLTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.012  -8.579   0.212   8.370  65.611
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 38.388014   5.467090   7.022 3.24e-12 ***
## FIELDING_DP -0.099038   0.014679  -6.747 2.11e-11 ***
## FIELDING_E  -0.036269   0.002210 -16.412  < 2e-16 ***
## BASERUN_SB   0.054626   0.004794  11.395  < 2e-16 ***
## BATTING_SO  -0.014571   0.002573  -5.664 1.75e-08 ***
## BATTING_HR   0.082276   0.010474   7.855 7.31e-15 ***
## BATTING_H    0.040958   0.003118  13.136  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.84 on 1586 degrees of freedom
## Multiple R-squared:  0.3326, Adjusted R-squared:   0.33
## F-statistic: 131.7 on 6 and 1586 DF,  p-value: < 2.2e-16
```



5 Select model

5.1 Model statistics

In considering how to evaluate the models additional modeling statistics were produced. Since the objective is to accurately predict `TARGET_WINS` the Root-Mean Square Error (RMSE) will be used as the primary metric for model evaluation. The RMSE is proportionally effected by residuals so larger errors, worse predictions, have a larger impact on the score.

Model statistics

	r.squared	adj.r.squared	fstat.value	RMSE
model one	0.3488	0.3447	84.73	12.66
model two	0.3328	0.3282	71.69	12.81
model three	0.3653	0.3592	60.47	10.73
model four	0.3326	0.33	131.7	12.81

Looking at the modeling statistics model three has the lowest RMSE. However since this model was created by removing approximately 450 potential outliers, this model has a greater potential to overfit the data. The prediction table below shows that model three has the highest RMSE on the reserved evaluation data. This suggests that this model was overfit to the training data. Comparing the remaining two models, model one has both the highest adjusted r-squared and the smallest differential between RMSE on the training and reserved evaluation data.

Prediction statistics

	RMSE
model one	12.41
model two	12.33
model three	13.87
model four	12.44

5.2 Prediction

Predictions of `TARGET_WINS` using the evaluation data and selected model are included in the provided comma separated values (CSV) file.

6 Appendix

6.1 Session info

```
## Session info -------------------------------------------------------------

##  setting  value
##  version  R version 3.5.3 (2019-03-11)
##  system   x86_64, linux-gnu
##  ui       X11
##  language en_US
##  collate  en_US.UTF-8
##  tz       America/New_York
##  date     2019-04-06

## Packages -----------------------------------------------------------------

##  package      * version    date       source
##  abind          1.4-5      2016-07-21 CRAN (R 3.5.1)
##  assertthat     0.2.1      2019-03-21 CRAN (R 3.5.3)
##  backports      1.1.3      2018-12-14 cran (@1.1.3)
##  base         * 3.5.3      2019-03-11 local
##  broom          0.5.1      2018-12-05 cran (@0.5.1)
##  car          * 3.0-0      2018-04-02 CRAN (R 3.5.1)
##  carData      * 3.0-1      2018-03-28 CRAN (R 3.5.1)
```

```
##  caret        * 6.0-80     2018-05-26 CRAN (R 3.5.1)
##  cellranger     1.1.0      2016-07-27 CRAN (R 3.5.1)
##  class          7.3-15     2019-01-01 CRAN (R 3.5.2)
##  codetools      0.2-16     2018-12-24 CRAN (R 3.5.2)
##  colorspace     1.4-1      2019-03-18 CRAN (R 3.5.3)
##  compiler       3.5.3      2019-03-11 local
##  crayon         1.3.4      2017-09-16 CRAN (R 3.5.0)
##  curl           3.2        2018-03-28 CRAN (R 3.5.1)
##  CVST           0.2-2      2018-05-26 CRAN (R 3.5.1)
##  data.table     1.11.4     2018-05-27 CRAN (R 3.5.1)
##  datasets     * 3.5.3      2019-03-11 local
##  ddalpha        1.3.4      2018-06-23 CRAN (R 3.5.1)
##  DEoptimR       1.0-8      2016-11-19 CRAN (R 3.5.1)
##  devtools       1.13.6     2018-06-27 CRAN (R 3.5.1)
##  digest         0.6.18     2018-10-10 CRAN (R 3.5.1)
##  dimRed         0.1.0      2017-05-04 CRAN (R 3.5.1)
##  dplyr        * 0.8.0.1    2019-02-15 cran (@0.8.0.1)
##  DRR            0.0.3      2018-01-06 CRAN (R 3.5.1)
##  e1071        * 1.7-0      2018-07-28 CRAN (R 3.5.1)
##  evaluate       0.13       2019-02-12 CRAN (R 3.5.2)
##  forcats        0.3.0      2018-02-19 CRAN (R 3.5.1)
##  foreach        1.4.4      2017-12-12 CRAN (R 3.5.1)
##  foreign        0.8-71     2018-07-20 CRAN (R 3.5.1)
##  generics       0.0.2      2018-11-29 cran (@0.0.2)
##  geometry       0.3-6      2015-09-09 CRAN (R 3.5.1)
##  ggcorrplot   * 0.1.2      2018-09-11 CRAN (R 3.5.1)
##  ggfortify    * 0.4.5      2018-05-26 CRAN (R 3.5.1)
##  ggplot2      * 3.1.0      2018-10-25 CRAN (R 3.5.1)
##  glue           1.3.1      2019-03-12 CRAN (R 3.5.3)
##  gower          0.1.2      2017-02-23 CRAN (R 3.5.1)
##  graphics     * 3.5.3      2019-03-11 local
##  grDevices    * 3.5.3      2019-03-11 local
##  grid           3.5.3      2019-03-11 local
##  gridExtra      2.3        2017-09-09 CRAN (R 3.5.1)
##  gtable         0.3.0      2019-03-25 CRAN (R 3.5.3)
##  haven          1.1.2      2018-06-27 CRAN (R 3.5.1)
##  hms            0.4.2      2018-03-10 CRAN (R 3.5.1)
##  htmltools      0.3.6      2017-04-28 CRAN (R 3.5.0)
##  ipred          0.9-7      2018-08-14 CRAN (R 3.5.1)
##  iterators      1.0.10     2018-07-13 CRAN (R 3.5.1)
##  jomo           2.6-4      2018-08-30 CRAN (R 3.5.1)
##  kernlab        0.9-27     2018-08-10 CRAN (R 3.5.1)
##  knitr          1.22       2019-03-08 CRAN (R 3.5.2)
##  labeling       0.3        2014-08-23 CRAN (R 3.5.0)
##  lattice      * 0.20-38    2018-11-04 CRAN (R 3.5.1)
##  lava           1.6.3      2018-08-10 CRAN (R 3.5.1)
##  lazyeval       0.2.2      2019-03-15 CRAN (R 3.5.3)
##  leaps        * 3.0        2017-01-10 CRAN (R 3.5.2)
##  lme4           1.1-17     2018-04-03 CRAN (R 3.5.1)
##  lubridate      1.7.4      2018-04-11 CRAN (R 3.5.1)
##  magic          1.5-9      2018-09-17 CRAN (R 3.5.1)
##  magrittr       1.5        2014-11-22 CRAN (R 3.5.0)
##  MASS         * 7.3-51.1   2018-11-01 CRAN (R 3.5.1)
##  Matrix         1.2-17     2019-03-22 CRAN (R 3.5.3)
```

```
##  memoise        1.1.0      2017-04-21 CRAN (R 3.4.1)
##  methods      * 3.5.3      2019-03-11 local
##  mgcv           1.8-28     2019-03-21 CRAN (R 3.5.3)
##  mice         * 3.3.0      2018-07-27 CRAN (R 3.5.1)
##  minqa          1.2.4      2014-10-09 CRAN (R 3.5.1)
##  mitml          0.3-6      2018-07-10 CRAN (R 3.5.1)
##  ModelMetrics * 1.2.0      2018-08-10 CRAN (R 3.5.1)
##  munsell        0.5.0      2018-06-12 CRAN (R 3.5.0)
##  nlme           3.1-137    2018-04-07 CRAN (R 3.5.0)
##  nloptr         1.0.4      2017-08-22 CRAN (R 3.5.1)
##  nnet           7.3-12     2016-02-02 CRAN (R 3.5.0)
##  openxlsx       4.1.0      2018-05-26 CRAN (R 3.5.1)
##  pan            1.6        2018-06-29 CRAN (R 3.5.1)
##  pander       * 0.6.2      2018-07-08 CRAN (R 3.5.1)
##  parallel       3.5.3      2019-03-11 local
##  pillar         1.3.1      2018-12-15 CRAN (R 3.5.2)
##  pkgconfig      2.0.2      2018-08-16 cran (@2.0.2)
##  pls            2.7-0      2018-08-21 CRAN (R 3.5.1)
##  plyr           1.8.4      2016-06-08 CRAN (R 3.5.0)
##  prodlim        2018.04.18 2018-04-18 CRAN (R 3.5.1)
##  purrr        * 0.3.0      2019-01-27 cran (@0.3.0)
##  R6             2.4.0      2019-02-14 CRAN (R 3.5.2)
##  Rcpp           1.0.0      2018-11-07 cran (@1.0.0)
##  RcppRoll       0.3.0      2018-06-05 CRAN (R 3.5.1)
##  readxl         1.1.0      2018-04-20 CRAN (R 3.5.1)
##  recipes        0.1.3      2018-06-16 CRAN (R 3.5.1)
##  reshape2       1.4.3      2017-12-11 CRAN (R 3.5.0)
##  rio            0.5.10     2018-03-29 CRAN (R 3.5.1)
##  rlang          0.3.3      2019-03-29 CRAN (R 3.5.3)
##  rmarkdown      1.11       2018-12-08 CRAN (R 3.5.2)
##  robustbase     0.93-2     2018-07-27 CRAN (R 3.5.1)
##  rpart          4.1-13     2018-02-23 CRAN (R 3.5.0)
##  scales       * 1.0.0      2018-08-09 CRAN (R 3.5.1)
##  sfsmisc        1.1-2      2018-03-05 CRAN (R 3.5.1)
##  splines        3.5.3      2019-03-11 local
##  stats        * 3.5.3      2019-03-11 local
##  stats4         3.5.3      2019-03-11 local
##  stringi        1.4.3      2019-03-12 CRAN (R 3.5.3)
##  stringr        1.4.0      2019-02-10 CRAN (R 3.5.2)
##  survival       2.43-3     2018-11-26 CRAN (R 3.5.1)
##  tibble         2.0.1      2019-01-12 cran (@2.0.1)
##  tidyr        * 0.8.2      2018-10-28 cran (@0.8.2)
##  tidyselect     0.2.5      2018-10-11 cran (@0.2.5)
##  timeDate       3043.102   2018-02-21 CRAN (R 3.5.1)
##  tools          3.5.3      2019-03-11 local
##  utils        * 3.5.3      2019-03-11 local
##  withr          2.1.2      2018-03-15 CRAN (R 3.5.0)
##  xfun           0.3        2018-07-06 CRAN (R 3.5.1)
##  yaml           2.2.0      2018-07-25 CRAN (R 3.5.1)
##  zip            1.0.0      2017-04-25 CRAN (R 3.5.1)
```

6.2 R source code

See included Rmarkdown (rmd) document