Multiple Linear Regression Modeling

DATA 621 - Business Analytics and Data Mining

Rafal Decowski

Contents

Introduction

The dataset contains statistics of professional baseball teams from the years 1871 to 2006. There are 2276 rows, each representing the performance of the team in the given year. Each row contains 16 unique variables, one of which is the ultimate mark of the teams' success – number of wins.

The goal of this project is to investigate the data and build a multiple linear regression model that predicts the number of wins.

Data Exploration

Dataset

There are 2276 rows and 16 columns. All the variables are of integer type which makes it easier to perform calculations and statistical analysis.

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT
INDEX	Identification Variable (do not use)	None
TARGET_WINS	Number of wins	
TEAM_BATTING_H	Base Hits by batters (1B,2B,3B,HR)	Positive Impact on Wins
TEAM_BATTING_2B	Doubles by batters (2B)	Positive Impact on Wins
TEAM_BATTING_3B	Triples by batters (3B)	Positive Impact on Wins
TEAM_BATTING_HR	Homeruns by batters (4B)	Positive Impact on Wins
TEAM_BATTING_BB	Walks by batters	Positive Impact on Wins
TEAM_BATTING_HBP	Batters hit by pitch (get a free base)	Positive Impact on Wins
TEAM_BATTING_SO	Strikeouts by batters	Negative Impact on Wins
TEAM_BASERUN_SB	Stolen bases	Positive Impact on Wins
TEAM_BASERUN_CS	Caught stealing	Negative Impact on Wins
TEAM_FIELDING_E	Errors	Negative Impact on Wins
TEAM_FIELDING_DP	Double Plays	Positive Impact on Wins
TEAM_PITCHING_BB	Walks allowed	Negative Impact on Wins
TEAM_PITCHING_H	Hits allowed	Negative Impact on Wins
TEAM_PITCHING_HR	Homeruns allowed	Negative Impact on Wins
TEAM_PITCHING_SO	Strikeouts by pitchers	Positive Impact on Wins

Descriptive Statistics

The very first step of my analysis will explore the fundamental descriptive statistics of the dataset. Even though it is still a tabular and numerical representation of the data, it gives a high-level overview of the contents such as range of the values (min, max and quartiles), measures of central location (mean and median) as well as missing data. In the table below, I will mark in red some of the things that stand out at the first glance.

STAT	TARGET_WINS	TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B
1ST QU.	71	1383	208	34
3RD QU.	92	1537	273	72
MAX.	146	2554	458	223
MEAN	80.79	1469	241.2	55.25
MEDIAN	82	1454	238	47
MIN.	0	891	69	0
NA'S	0	0	0	0

STAT	TEAM_BATTING_HR	TEAM_BATTING_BB	TEAM_BATTING_SO	TEAM_BASERUN_SB
1ST QU.	42	451	548	66
3RD QU.	147	580	930	156
MAX.	264	878	1399	697
MEAN	99.61	501.6	735.6	124.8
MEDIAN	102	512	750	101
MIN.	0	0	0	0
NA'S	0	0	102	131

STAT	TEAM_BASERUN_CS	TEAM_BATTING_HBP	TEAM_PITCHING_H	TEAM_PITCHING_HR
1ST QU.	38	50.5	1419	50
3RD QU.	62	67	1682	150
MAX.	201	95	30132	343
MEAN	52.8	59.36	1779	105.7
MEDIAN	49	58	1518	107
MIN.	0	29	1137	0
NA'S	772	2085	0	0

STAT	TEAM_PITCHING_BB	TEAM_PITCHING_SO	TEAM_FIELDING_E	TEAM_FIELDING_DP
1ST QU.	476	615	127	131
3RD QU.	611	968	249.2	164
MAX.	3645	19278	1898	228
MEAN	553	817.7	246.5	146.4
MEDIAN	536.5	813.5	159	149
MIN.	0	0	65	52
NA'S	0	102	0	286

Observations form the table above:

- Maximum number of wins is almost twice as large as the mean which indicates there is a team in our dataset that had at least one amazing year which is worth studying closely.
- Six of the variables contain NA's. We must take that into consideration later when building predictive models. Two of those variables, *TEAM_BASERUN_CS* & *TEAM_BATTING_HBP* are missing a large portion of the data – around 33% and 92% which may be a reason for not using them all-together.
- Some of the max & min values seem to be outliers which may lower the accuracy of our models.
- TEAM_FIELDING_E & TEAM_BASERUN_SB variable may not have a near normal distributions since the median and mean are significantly different. This indicates a need to remove outliers.

Initial visualization

The next step is to visualize the data set.

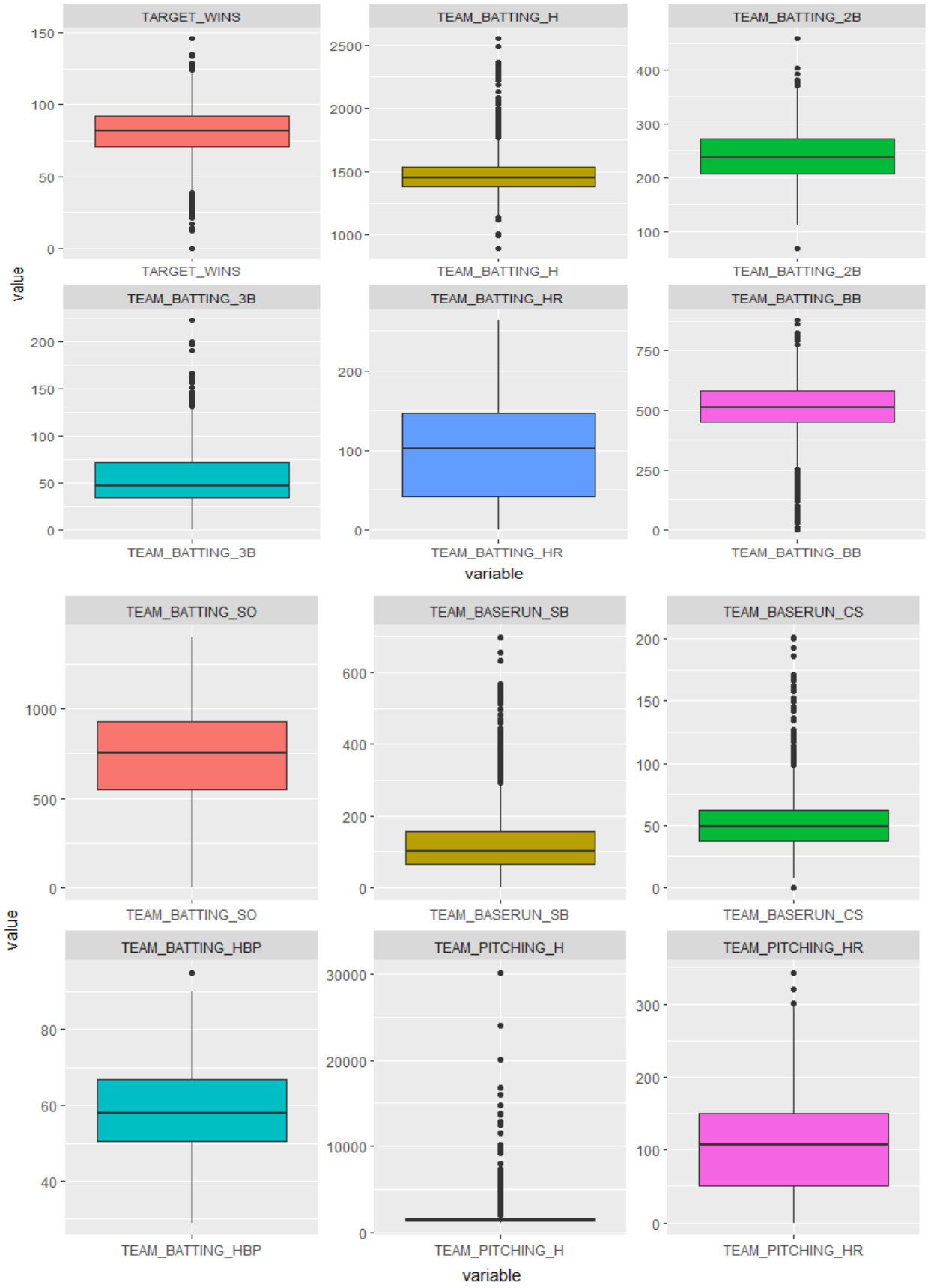
Boxplots are great for painting a picture of the descriptive statistics outlined in the tables in the previous section. Looking at the graphs we can determine the following:

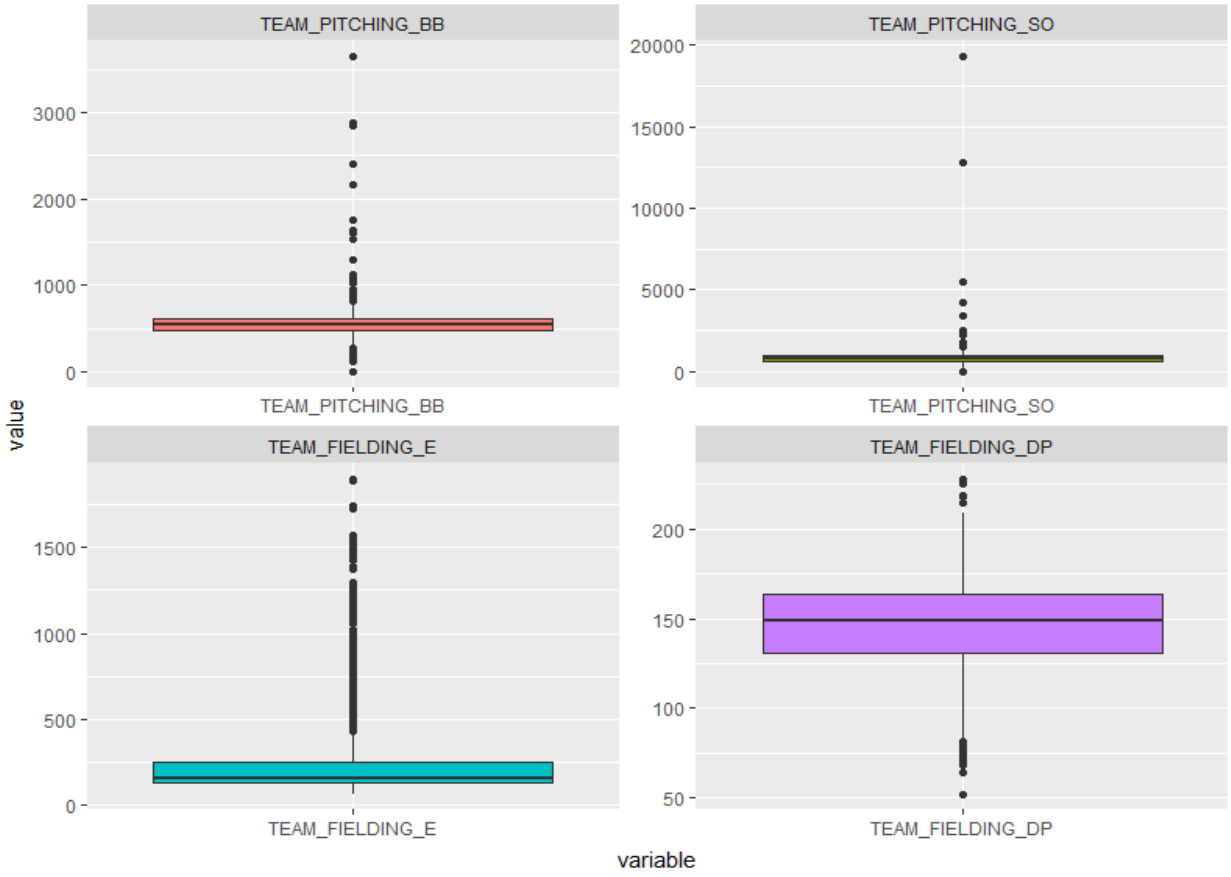
- Upper outliers marked as black dots
- Upper whisker indicating variability outside the upper(3^{rd}) quartile
- A colored box which spans between 3^{rd} and 1^{st} quartiles with a black line which indicates the median value of the variable
- Lower whisker indicating variability outside the lower (1^{st}) quartile
- Lower outliers marked as black dots

To make the visualizations more readable I divided the dataframe into 3 groups, containing 6, 6 and 4 variables respectively. Due to a high variation in the values of each variable I decided to graph each variable with their own Y-scale.

Observations from the boxplots below:

- TARGET_WINS – by looking at the outliers, the graph suggests there are more underperforming teams than there are superstars (upper outliers.) Since this is our target variable we should study the outliers as with more emphasis of what works best and what does not work for the team's success.
- TEAM_BATTING_H – multiple upper outliers and a small interquartile range.
- TEAM_BATTING_HR – No outliers flagged, it appears to be a steady dataset for homeruns
- TEAM_BASERUN_SB – stolen bases reported multiple upper outliers, perhaps a shift in the play itself might have evolved over the years
- TEAM_PITCHING_H – this variable represents *hits allowed* and shows an unusual number of outliers. This variable is assumed to negatively impact the win rate and such a high number of outliers may decrease the accuracy of the final predictive model.
- TEAM_PITCHING_SO & TEAM_FIELDING_E also show significant numbers of outliers. Displaying these variables over time may provide additional insights.


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_BATTING_HBP
TEAM_PITCHING_H
TEAM_PITCHING_HR
value
variable


TEAM_PITCHING_BB
TEAM_PITCHING_SO
TEAM_FIELDING_E
TEAM_FIELDING_DP
value
variable
0
1000
2000
3000
5000
10000
15000
20000
500
1500
50
100
150
200

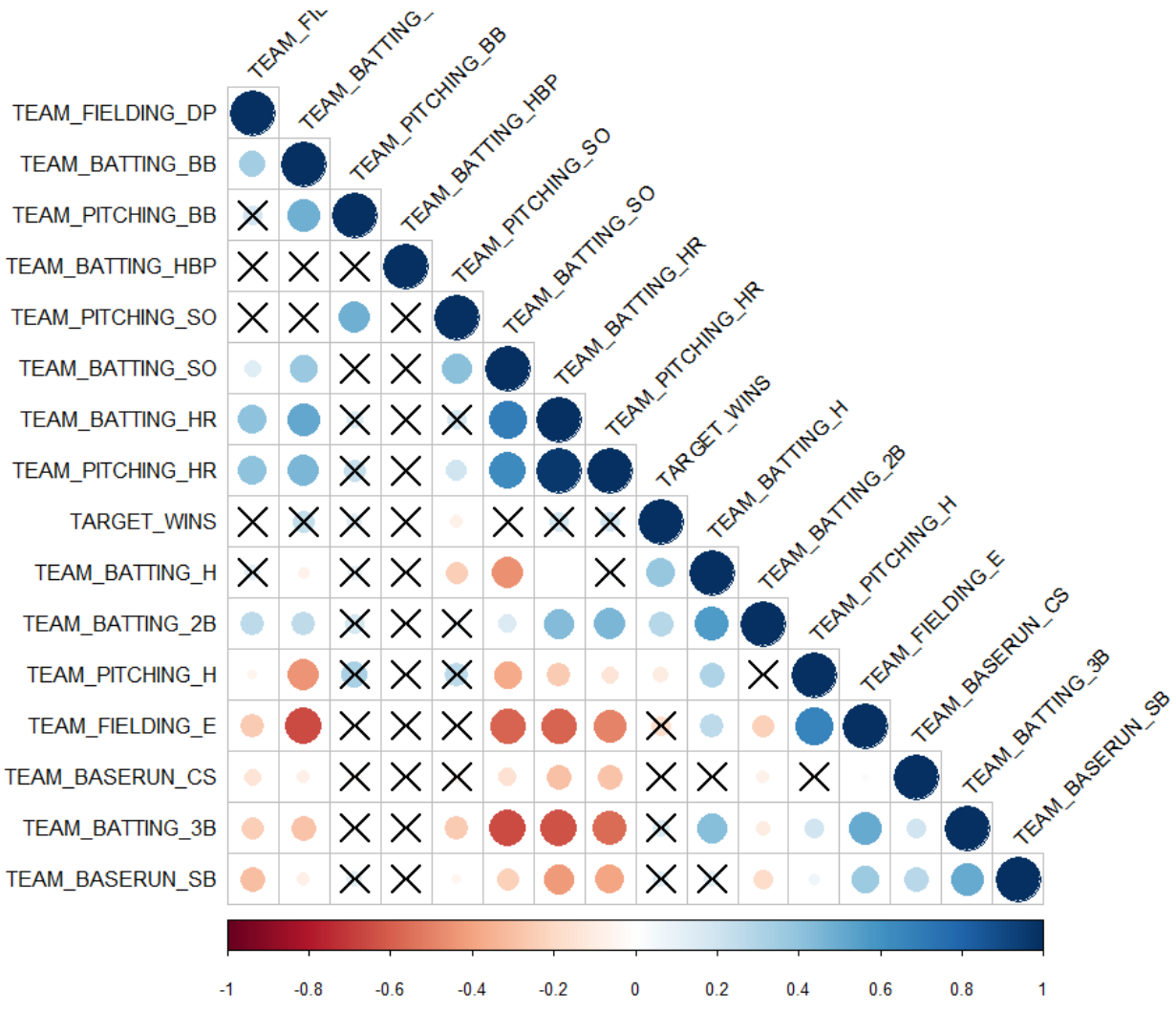
Correlation & Significance Test

The table below shows correlation between the number of won games and other variables. Homeruns have the highest correlation with winning games followed by doubles by batters whereas errors and the number of hits allowed negatively impacts the games.

Variable	Correlation
TARGET_WIN	1
TEAM_BATTING_H	0.38876752
TEAM_BATTING_2B	0.28910365
TEAM_BATTING_BB	0.23255986
TEAM_PITCHING_HR	0.18901373
TEAM_BATTING_HR	0.1761532
TEAM_BATTING_3B	0.14260841
TEAM_PITCHING_BB	0.12417454
TEAM_BASERUN_SB	0.12297192
TEAM_BATTING_HBP	0.01633738
TEAM_BASERUN_CS	0.01556444
TEAM_FIELDING_DP	-0.02884126
TEAM_BATTING_SO	-0.03067847
TEAM_PITCHING_SO	-0.07578725
TEAM_PITCHING_H	-0.10993705
TEAM_FIELDING_E	-0.17648476

The heatmap below makes it easy for us to find the correlation between variables. The color as well as the size of the circles within each box represents the strength of the correlation and whether it impacts it positively or negatively. Red shows negative impact and blue positive. The reason why it's important to look at each variable's correlation to each other and not just the target variable is to determine if they can contribute something unique to the model. This is the case when they are *not* correlated to each other.

Furthermore, the crossed out boxes may be rather insignificant when considering 95% confidence level and 20% significance level.



Data Preparation

Variable Removal

There are two variables TEAM_BATTING_HBP & TEAM_BASERUN_CS with a high portion of their data missing which I decided not to proceed with data fixing or transformations. These variables were removed from the dataframe.

Outlier Removal

The approach to deal with outliers used was to calculate the minimum and maximum 'benchmark' value for each variable based on their interquartile values. The values that were discarded from the dataset fell either below Q1-1.5*IQR or above Q3 + 1.5*IQR. This is so called 1.5 x IQR rule.

I used a function to iterate through each variable and replaced the values detected as outliers with a 'NA'.

Missing data treatment

In the section with descriptive statistics we pointed out that several variables have missing data. There are many ways to deal with this issue and we will some of them. The technique used will impact our model, so it is crucial to run multiple tests and determine the best strategy.

Filling techniques:

- Forward or backward fill – using this method we iterate through a variable and when a missing value is encountered we populate it with either previous value or the next available value. Since every record is independent and the data collection spans over 100 years, the variance may be large to rely on this technique.
- Mean or Median fill – this technique uses the mean or median of the entire variable to populate the missing values.
- Algorithmic - expectation–maximization (EM algorithm) or similar. This is a more advanced technique which uses statistical models to estimate variable's expected values.

The table below indicated the percentage of missing values for both dataframes, one with outliers and one after IQR treatment:

VARIABLE	BEFORE OUTLIER TREATMENT	AFTER OUTLIER TREATMENT
TARGET_WINS	0	0
TEAM_BATTING_H	0	0
TEAM_BATTING_2B	0	0.01
TEAM_BATTING_3B	0	0.01
TEAM_BATTING_HR	0	0.01
TEAM_BATTING_BB	0	0.03
TEAM_PITCHING_H	0	0.04
TEAM_PITCHING_HR	0	0.04
TEAM_PITCHING_BB	0	0.06
TEAM_FIELDING_E	0	0.06
TEAM_BATTING_SO	0.04	0.09
TEAM_PITCHING_SO	0.04	0.11
TEAM_BASERUN_SB	0.06	0.13
TEAM_FIELDING_DP	0.13	0.14

We can note that not only the number of missing data increased in the original dataset but clearing the outliers introduced data gaps in all columns but 2, one of which is our target variable.

Since we already forked our datasets by outlier treatment, this gives us an opportunity to use different approaches for building two or more predictive models.

1. The dataset with outliers contains a small number of missing values, we will simply discard them and capture only 'complete cases'. After removing rows from the bottom 4 variables we end up with a smaller set of 1835 cases but every variable contains full cases.
2. Since we removed outliers from the other set, its mean and median should be much closer as is the case for data with no outliers. We will treat the missing values with it's mean.

Normalization and Scaling

We are now working with two datasets,

1. Trimmed outliers and mean-filled missing data
2. Complete cases only with outliers

There are special normalization and scaling techniques which are adequate for those cases. Min-Max normalization for dataframe without outliers which uses the following formula:

(x - min(x)) / (max(x) - min(x)

For the set with complete cases, so called Z-Score Standardization is more appropriate because it appropriately weighs in the outliers whereas the min-max technique brings the values closer to the mean. There is a built in function in R to easily perform this operation.

Model Building

I started with 5 model ideas - 4 based on the dataset prepared in the previous section and 1 that ‘makes sense’ to me based on my knowledge of baseball. Each dataset used was divided into training and testing with the 80-20 ratio.

Initial composition:

- Model 1 - all variables, no outliers, mean filled, not normalized
- Model 2 - all variables, no outliers, mean filled NA's, min-max normalized
- Model 3 - all variables, complete cases, no transformation
- Model 4 - all variables, complete cases, Z-Score Standardization
- Model 5 - 'make-sense variables' + complete cases

Each of the first 4 models were adjusted by following both-directions **Stepwise Model Path**. The following variables were removed from each:

- Model 1 - TEAM_BATTING_2B, TEAM_PITCHING_HR, TEAM_PITCHING_SO
- Model 2 - TEAM_BATTING_2B, TEAM_BATTING_HR
- Model 3 - TEAM_PITCHING_HR, TEAM_PITCHING_BB, TEAM_BATTING_H
- Model 4 - TEAM_PITCHING_HR, TEAM_BATTING_H, TEAM_PITCHING_BB
- Model 5 – no changes

```
> summary(model1_adjusted)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B +
    TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB +
    TEAM_PITCHING_H + TEAM_FIELDING_E + TEAM_FIELDING_DP, data = train_m1)

Residuals:
    Min      1Q  Median      3Q     Max
-51.316  -8.271   0.187   8.049  58.330

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    54.241780   6.253283   8.674  < 2e-16 ***
TEAM_BATTING_H  0.030060   0.004082   7.363 2.71e-13 ***
TEAM_BATTING_3B 0.104574   0.019047   5.490 4.58e-08 ***
TEAM_BATTING_HR 0.083803   0.010304   8.133 7.67e-16 ***
TEAM_BATTING_BB 0.023587   0.003759   6.275 4.37e-10 ***
TEAM_BATTING_SO -0.016723  0.002246  -7.444 1.50e-13 ***
TEAM_BASERUN_SB 0.043053   0.006487   6.637 4.22e-11 ***
TEAM_PITCHING_H -0.007514  0.002659  -2.826  0.00476 **
TEAM_FIELDING_E -0.046883  0.006532  -7.177 1.04e-12 ***
TEAM_FIELDING_DP -0.108920 0.015184  -7.173 1.06e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.71 on 1810 degrees of freedom
Multiple R-squared:  0.2376,    Adjusted R-squared:  0.2338
F-statistic: 62.67 on 9 and 1810 DF,  p-value: < 2.2e-16
```

```
> summary(model2_adjusted)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B +
    TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB +
    TEAM_PITCHING_H + TEAM_PITCHING_BB + TEAM_FIELDING_E + TEAM_FIELDING_DP,
    data = train_m2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.56818 -0.09917  0.00258  0.09904  0.66873

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.37396    0.03304  11.318  < 2e-16 ***
TEAM_BATTING_H    0.25001    0.03130   7.988 2.42e-15 ***
TEAM_BATTING_3B   0.18783    0.02946   6.377 2.29e-10 ***
TEAM_BATTING_HR   0.23359    0.03324   7.028 2.95e-12 ***
TEAM_BATTING_BB   0.07735    0.03722   2.078  0.03785 *
TEAM_BATTING_SO  -0.20424    0.03796  -5.380 8.42e-08 ***
TEAM_BASERUN_SB   0.12653    0.02246   5.634 2.04e-08 ***
TEAM_PITCHING_H  -0.12181    0.03159  -3.855  0.00012 ***
TEAM_PITCHING_BB  0.07748    0.03483   2.224  0.02626 *
TEAM_FIELDING_E  -0.18712    0.02915  -6.418 1.76e-10 ***
TEAM_FIELDING_DP -0.15848    0.02378  -6.666 3.48e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.153 on 1809 degrees of freedom
Multiple R-squared:  0.2236,    Adjusted R-squared:  0.2194
F-statistic: 52.11 on 10 and 1809 DF,  p-value: < 2.2e-16
```

```
> summary(model3_adjusted)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
    TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_BB + TEAM_PITCHING_SO +
    TEAM_FIELDING_E + TEAM_FIELDING_DP, data = train_m3)

Residuals:
    Min      1Q  Median      3Q     Max
-32.238  -7.359   0.012   6.921  30.120

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      57.330856   6.798732   8.433  < 2e-16 ***
TEAM_BATTING_H   -0.024857   0.017760  -1.400  0.16185
TEAM_BATTING_2B  -0.051939   0.009891  -5.251 1.74e-07 ***
TEAM_BATTING_3B   0.197467   0.021412   9.222  < 2e-16 ***
TEAM_BATTING_HR   0.107379   0.010176  10.552  < 2e-16 ***
TEAM_BATTING_BB   0.093221   0.044956   2.074  0.03829 *
TEAM_BATTING_SO   0.035843   0.019125   1.874  0.06111 .
TEAM_BASERUN_SB   0.069389   0.006175  11.237  < 2e-16 ***
TEAM_PITCHING_H   0.052417   0.016171   3.241  0.00122 **
TEAM_PITCHING_BB -0.057402   0.042657  -1.346  0.17862
TEAM_PITCHING_SO -0.056561   0.018047  -3.134  0.00176 **
TEAM_FIELDING_E  -0.117814   0.007977 -14.769  < 2e-16 ***
TEAM_FIELDING_DP -0.112177   0.013585  -8.258 3.30e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.2 on 1455 degrees of freedom
Multiple R-squared:  0.4144,    Adjusted R-squared:  0.4095
F-statistic: 85.79 on 12 and 1455 DF,  p-value: < 2.2e-16
```

```
> summary(model4_adjusted)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
    TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_BB + TEAM_PITCHING_SO +
    TEAM_FIELDING_E + TEAM_FIELDING_DP, data = train_m4)

Residuals:
     Min       1Q   Median       3Q      Max
-2.45713 -0.55288  0.01808  0.53310  2.25976

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.003317   0.020222  -0.164  0.86974
TEAM_BATTING_H   -0.145239   0.139142  -1.044  0.29674
TEAM_BATTING_2B  -0.187506   0.032039  -5.852 5.98e-09 ***
TEAM_BATTING_3B   0.284239   0.035567   7.992 2.69e-15 ***
TEAM_BATTING_HR   0.396315   0.042332   9.362  < 2e-16 ***
TEAM_BATTING_BB   0.613018   0.287650   2.131  0.03325 *
TEAM_BATTING_SO   0.563500   0.320257   1.760  0.07870 .
TEAM_BASERUN_SB   0.282642   0.024987  11.312  < 2e-16 ***
TEAM_PITCHING_H   0.653880   0.203280   3.217  0.00133 **
TEAM_PITCHING_BB -0.429056   0.311430  -1.378  0.16851
TEAM_PITCHING_SO -0.879203   0.309719  -2.839  0.00459 **
TEAM_FIELDING_E  -0.498620   0.034910 -14.283  < 2e-16 ***
TEAM_FIELDING_DP -0.171088   0.023614  -7.245 6.98e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7743 on 1455 degrees of freedom
Multiple R-squared:  0.3972,    Adjusted R-squared:  0.3922
F-statistic: 79.88 on 12 and 1455 DF,  p-value: < 2.2e-16
```

```
> summary(model5)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + 
    TEAM_BATTING_3B + TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHING_H + 
    TEAM_FIELDING_E + TEAM_FIELDING_DP, data = train_m5)

Residuals:
    Min      1Q  Median      3Q     Max 
-34.710  -7.381  -0.045   7.225  30.074 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      16.2008957  4.9247155   3.290  0.00103 ** 
TEAM_BATTING_H    0.0551376  0.0049588  11.119  < 2e-16 ***
TEAM_BATTING_2B  -0.0761529  0.0099128  -7.682 2.85e-14 ***
TEAM_BATTING_3B   0.1487467  0.0191000   7.788 1.28e-14 ***
TEAM_BATTING_BB   0.0431834  0.0035611  12.126  < 2e-16 ***
TEAM_BASERUN_SB   0.0491334  0.0059200   8.300 2.35e-16 ***
TEAM_PITCHING_H   0.0008721  0.0023641   0.369  0.71227    
TEAM_FIELDING_E  -0.1136045  0.0073610 -15.433  < 2e-16 ***
TEAM_FIELDING_DP -0.0974343  0.0142944  -6.816 1.36e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.64 on 1459 degrees of freedom
Multiple R-squared:  0.3419,    Adjusted R-squared:  0.3383 
F-statistic: 94.76 on 8 and 1459 DF,  p-value: < 2.2e-16
```

Model Selection

First, indicator for the model accuracy was based on the testing data set aside while building and training my models. The following results were noted:

Correlation accuracies

Based on the initial setup with no adjustments in variables:

MODEL	ACCURACY
M1	47%
M2	52%
M3	61%
M4	41%
M5	63%

The stepwise variable adjustments only slightly increased overall performance of the models:

MODEL	ACCURACY
M1	47%
M2	53%
M3	61%
M4	50%
M5	63%

In both cases, the model with hand-picked variables seemed to outperform the other ones but that measure alone is not enough to finalize model selection.

R-Squared:

Models 1 and & 2 had significantly lower R-squared and Adjusted R-squared than the rest – around 0.2, and models 3 & 4 had the highest R-squared values, almost twice as large at 0.4. Model 5 is right in the middle at 0.33. The higher the value the better contribution to the overall variability of the model which puts models 3 & 4 in the lead in explaining the variation.

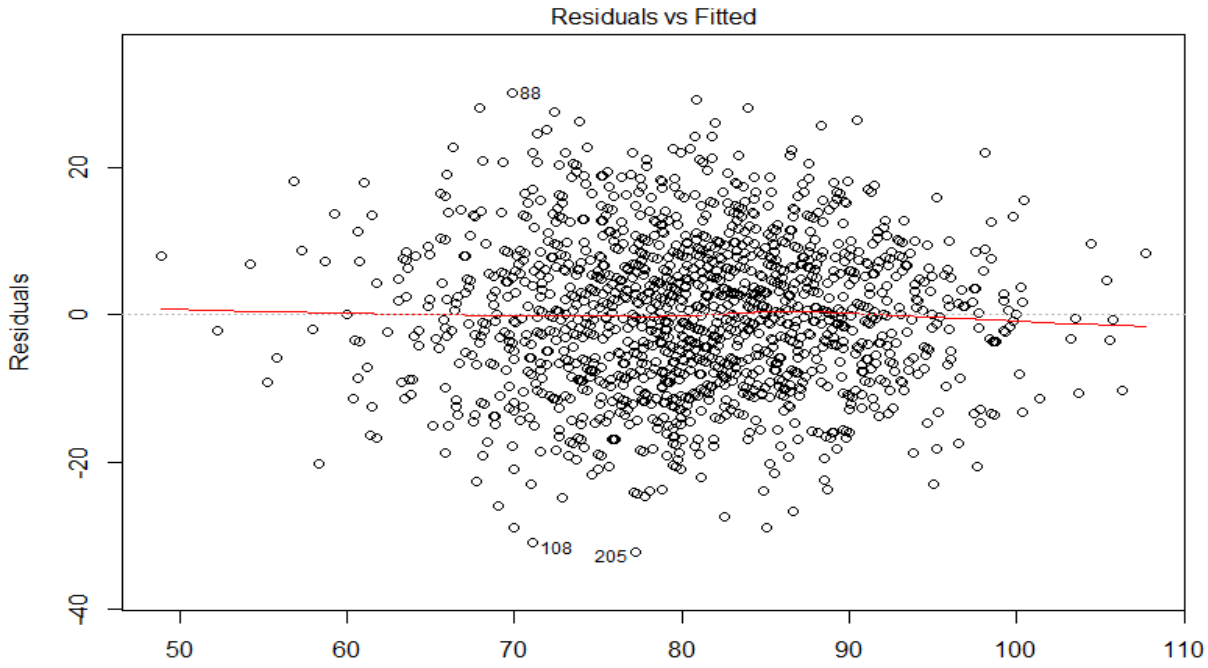
F-Statistic

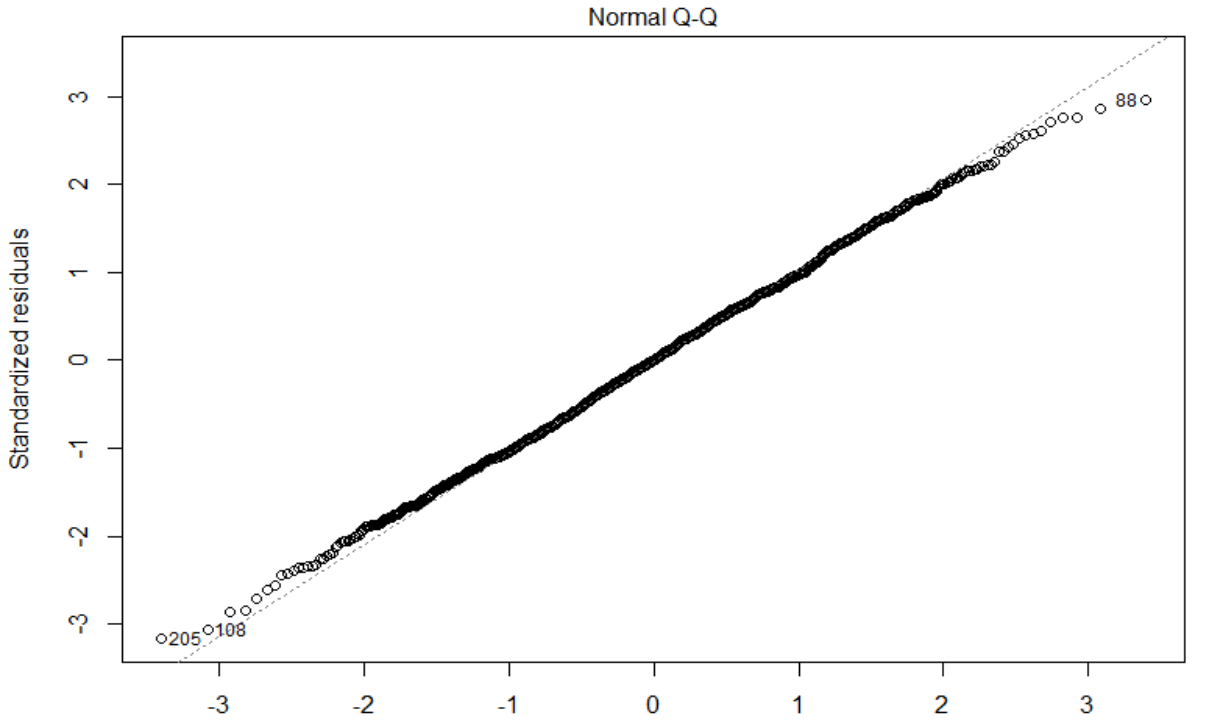
F-Statistic is used for hypothesis testing and the greater the value the better the model is. Models 3 and 5 are once again in the in the lead.

Residuals

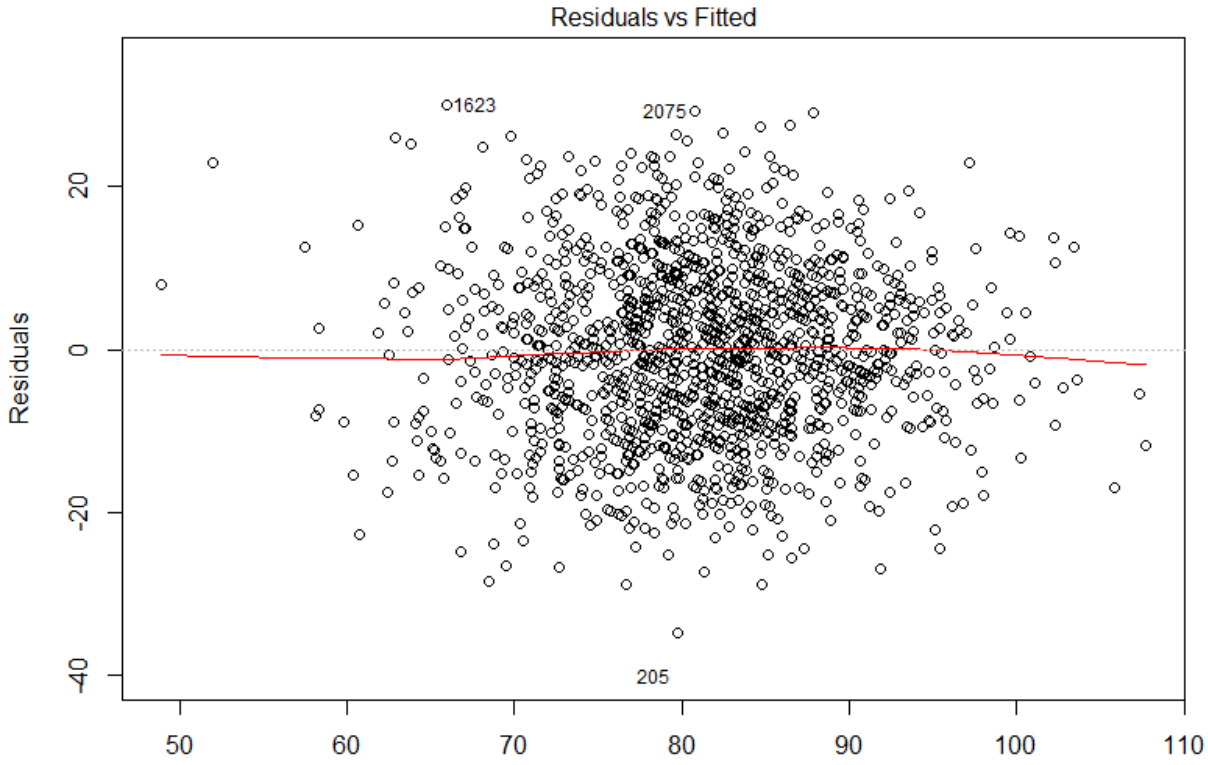
With all the previous indicators, we can safely narrow down to only two models competing for the final round – Model 3 and Model 5. The last step in our selection analysis will be residual plotting and investigation. Based on the graphs below, residuals vs fitted as well as the QQ line for both models show similar tendencies and fall under the linear regression assumptions.

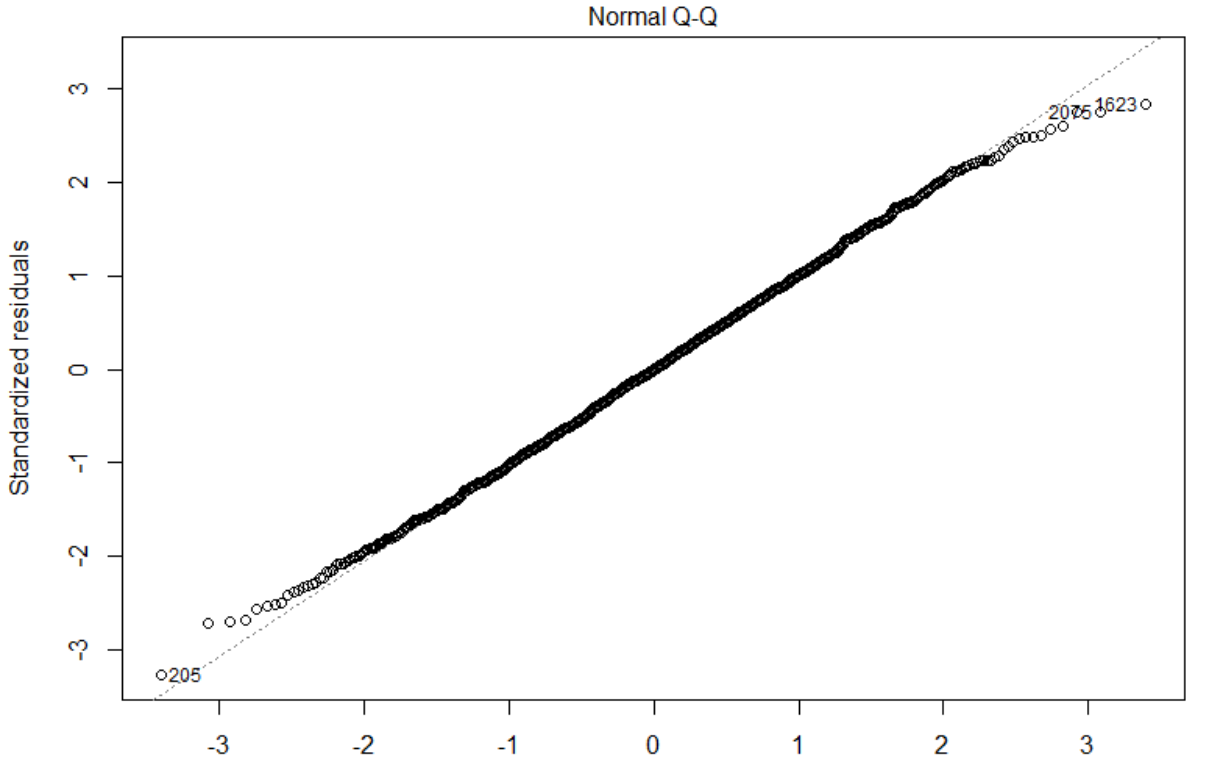
Model 3:





Model 5:


Residuals vs Fitted
Residuals
1623
2075
205
20
0
-20
-40
50
60
70
80
90
100
110


Normal Q-Q
Standardized residuals
1623
2075
205
3
2
1
0
-1
-2
-3

Final Selection

Model 3 seems to be the best model out of all 5. It holds the highest value for explaining the variability, it predicts with approximately 61% accuracy, there are at least 4 variables with fairly high p-value which indicates potential for additional tuning and improvements and most importantly it does not violate any of the regression assumptions.

Model 3 summary: It contains 11 variables; only complete cases were used for training with no additional data transformations and there is clear tuning potential.