

DATA621_HW_1

Critical group 2: Avery Davidowitz, Josh Iden, Mathew Katz, Tyler Brown, Gabriel Santos, John Ledesma

2023-02-11

```
knitr::opts_chunk$set(echo = FALSE, class.source = "codechunk")
library(RCurl)
library(tidyr)
library(dplyr)
library(RCurl)
library(ggplot2)
library(reshape2)
library(corrplot)
library(mice)
library(car)
library(reshape)
library(mixtools)
library(tidyverse)
library(GGally)
library(MASS)
library(faraway)
suppressPackageStartupMessages(library("visdat"))
```

Problem Statement and Goals

Our objective is to make a linear regression model that can predict how many wins a baseball team will have in a season based on some metrics.

The variables we have been provided theoretically have positive or negative effects on the total number of wins.

The goal is to establish which variables are correlated with wins, as well as finding out if some of the variables can be consolidated using known conventional baseball-stats algorithms like SABER.

Data Exploration

Viewing Data

The data contains 17 columns. The index column will be ignored for analysis purposes, and so that leaves the other 16. TARGET_WINS is the variable we want to investigate with regards to how well it is correlated with the other columns. Every row represents a baseball team and its performance during a particular season. TARGET_WINS is the number of wins, and each column after that represents a particular metric for the season. For example, TEAM_PITCHING_E represents how many times an opposing team made a pitching mistake during the season. 'TEAM_BATTING_H' represents how many base hits by batters occurred for that team during the season. There are four categories of feature types:

- Batting
- Baserunning
- Pitching
- Fielding

```

## TARGET_WINS      TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
## Min. : 0.00    Min. : 891     Min. : 69.0    Min. : 0.00
## 1st Qu.: 71.00  1st Qu.:1383   1st Qu.:208.0   1st Qu.: 34.00
## Median : 82.00  Median :1454   Median :238.0    Median : 47.00
## Mean   : 80.79  Mean   :1469   Mean   :241.2    Mean   : 55.25
## 3rd Qu.: 92.00  3rd Qu.:1537   3rd Qu.:273.0   3rd Qu.: 72.00
## Max.  :146.00  Max.  :2554    Max.  :458.0    Max.  :223.00
##
## TEAM_BATTING_HR TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB
## Min. : 0.00    Min. : 0.0     Min. : 0.0     Min. : 0.0
## 1st Qu.: 42.00  1st Qu.:451.0   1st Qu.: 548.0   1st Qu.: 66.0
## Median :102.00  Median :512.0   Median : 750.0   Median :101.0
## Mean   : 99.61  Mean   :501.6   Mean   : 735.6   Mean   :124.8
## 3rd Qu.:147.00  3rd Qu.:580.0   3rd Qu.: 930.0   3rd Qu.:156.0
## Max.  :264.00  Max.  :878.0    Max.  :1399.0   Max.  :697.0
## NA's   :102      NA's   :102    NA's   :131
##
## TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
## Min. : 0.0      Min. :29.00    Min. : 1137    Min. : 0.0
## 1st Qu.: 38.0   1st Qu.:50.50   1st Qu.: 1419    1st Qu.: 50.0
## Median : 49.0   Median :58.00    Median : 1518    Median :107.0
## Mean   : 52.8   Mean   :59.36    Mean   : 1779    Mean   :105.7
## 3rd Qu.: 62.0   3rd Qu.:67.00   3rd Qu.: 1682    3rd Qu.:150.0
## Max.  :201.0   Max.  :95.00    Max.  :30132   Max.  :343.0
## NA's   :772      NA's   :2085
##
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E  TEAM_FIELDING_DP
## Min. : 0.0      Min. : 0.0     Min. : 65.0    Min. : 52.0
## 1st Qu.: 476.0  1st Qu.: 615.0   1st Qu.: 127.0   1st Qu.:131.0
## Median : 536.5  Median : 813.5   Median : 159.0   Median :149.0
## Mean   : 553.0  Mean   : 817.7   Mean   : 246.5   Mean   :146.4
## 3rd Qu.: 611.0  3rd Qu.: 968.0   3rd Qu.: 249.2   3rd Qu.:164.0
## Max.  :3645.0  Max.  :19278.0  Max.  :1898.0   Max.  :228.0
## NA's   :102      NA's   :286

```

According to the data, we can see that that target variable is roughly normally distributed, with a mean of total wins around 80 games.

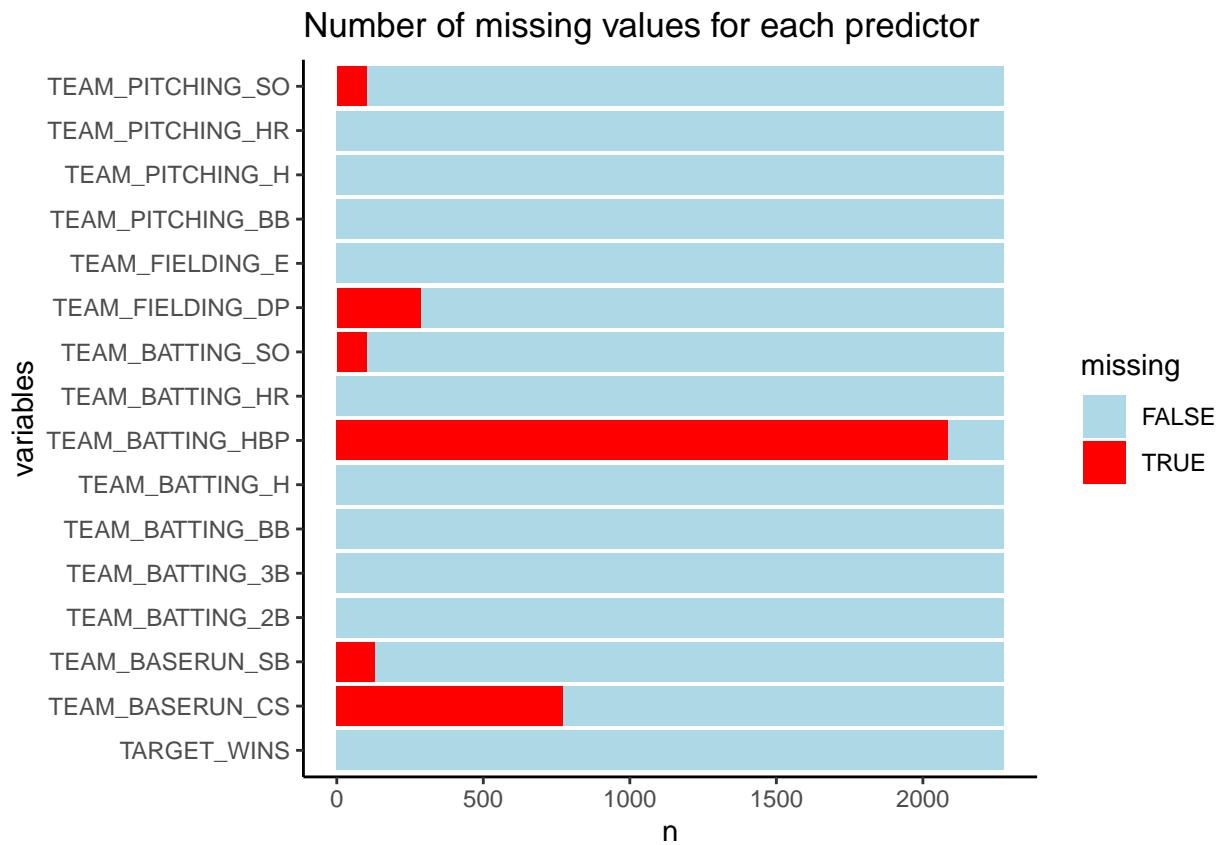
It might make sense, as a standard season is 162 games, we would expect that the average number of wins would be roughly half of this value.

There are a few columns which appear to have outliers, particularly TEAM_PITCHING_H.

Exploration

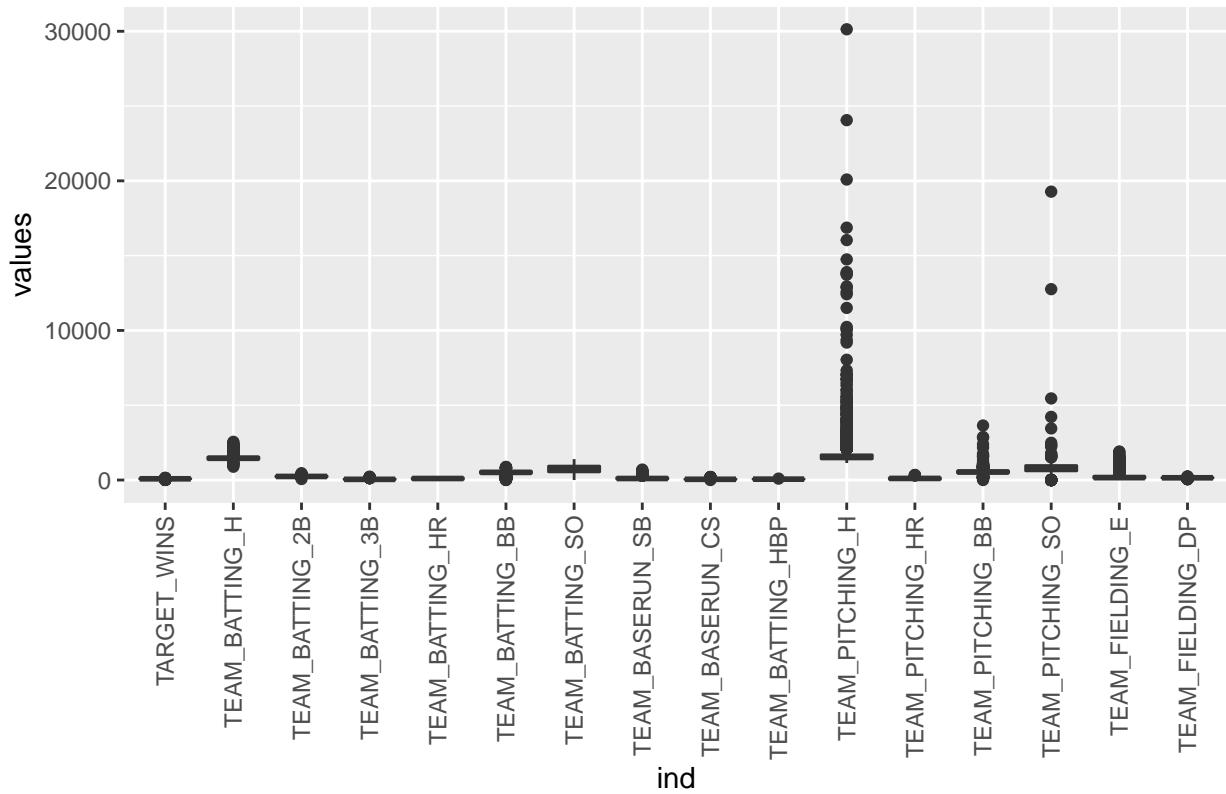
Some of the columns have missing values.

It is possible that an entire season could be played without strikeouts by batters.



We were able to observe some atypical values (outliers), some values are too high. For example, a team with more than 20,000 strikeouts.

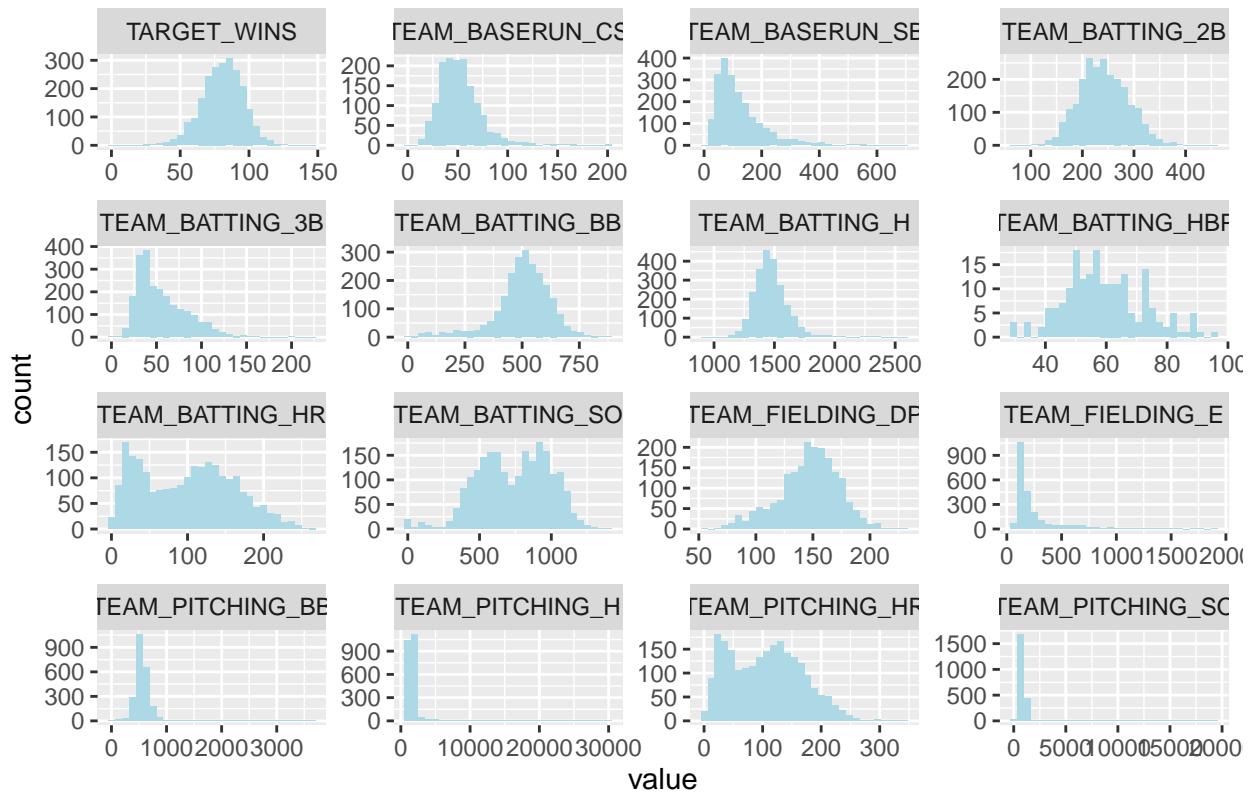
All variables



From the chart above, we can see that the most problematic features include TEAM_PITCHING_H, TEAM_PITCHING_BB, and TEAM_PITCHING_SO. Where available we will employ cutoffs based on third party reference data such as baseball-almanac.com. If there is no available data, we will use other logical imputation methods to replace the outliers with reasonable values more fit to the data.

Data distribution

Histograms All Variables

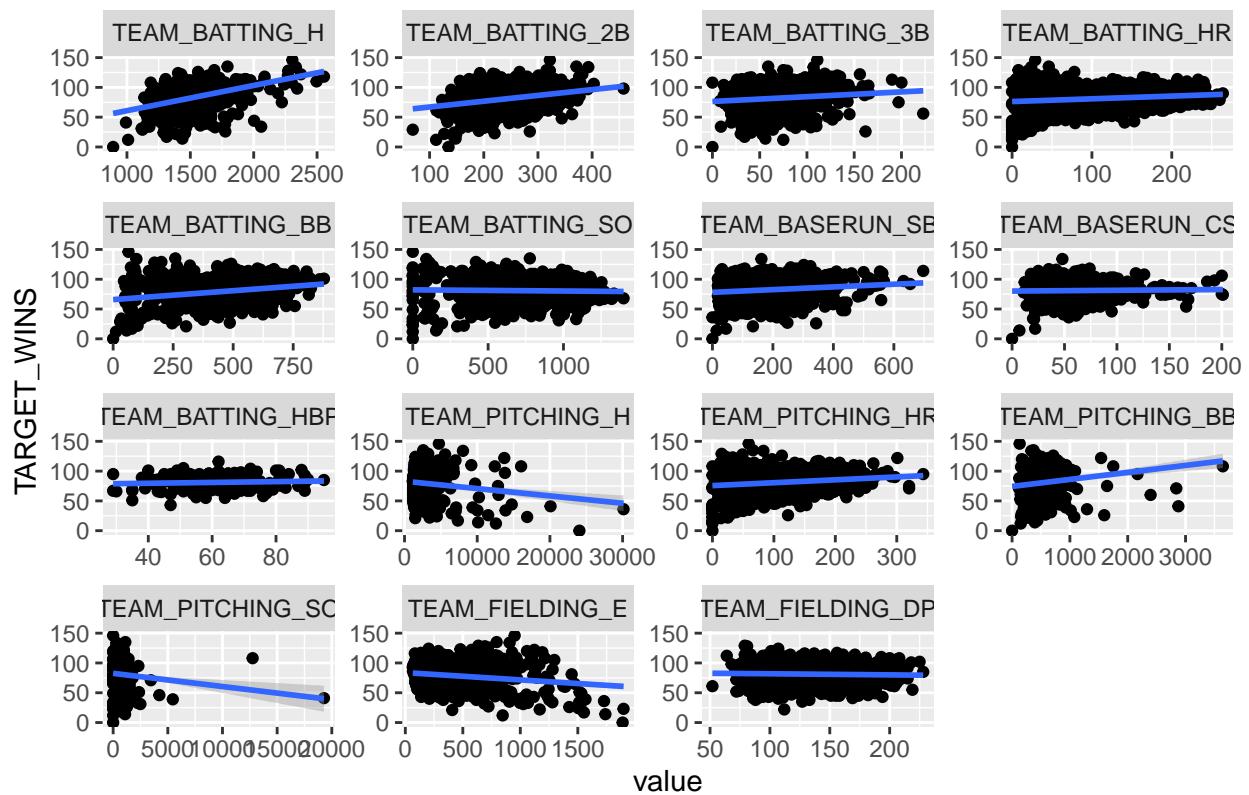


We can see that some of the variables are skewed to the right or the left like TEAM_PITCHING_SO. Some of them even have more than one spike (bimodal) like TEAM_PITCHING_H. Our target feature is very well normally distributed.

Correlation

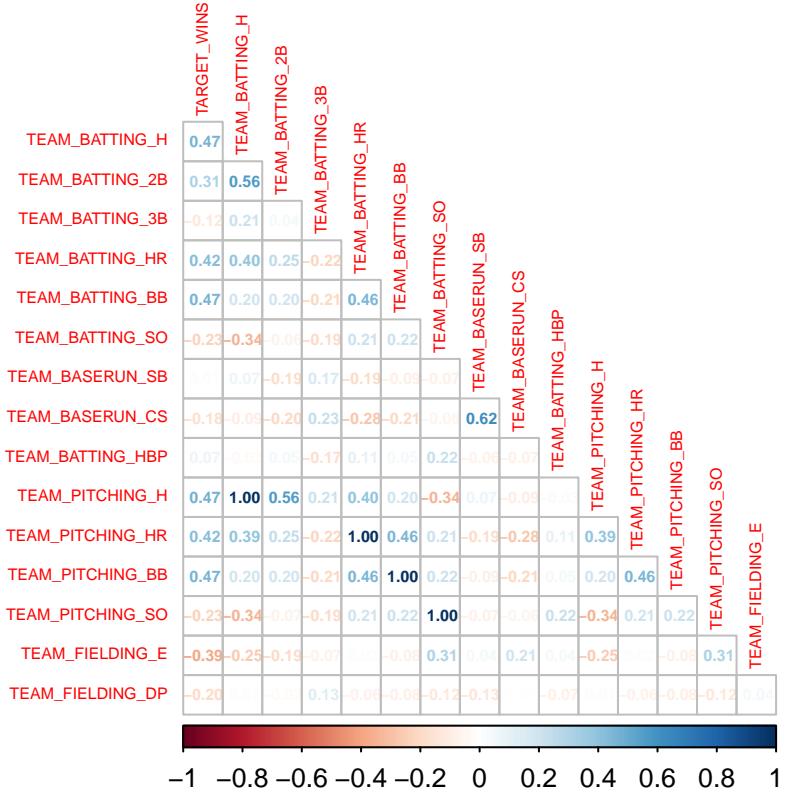
This is exploration of how the variables correlate with wins. In the chart below we can see that some of these variables correlate as we would expect with the number of wins - such as TEAM_BATTING correlating positively with wins. However some of them did not make sense- like TEAM_PITCHING_SO having a negative correlation with wins. We made this chart to get a general idea of how each variable related to the number of wins.

Correlation against the target variable for each predictor



Feature Multicollinearity

We want to know which characteristics are correlated, by means of a correlation graph.



we notice that there are a few features which exhibit very strong positive correlation. In particular:

- TEAM_PITCHING_H & TEAM_BATTING_H == 1.0 correlation
- TEAM_PITCHING_HR & TEAM_BATTING_HR == 1.0 correlation
- TEAM_PITCHING_BB & TEAM_BATTING_BB == 1.0 correlation
- TEAM_PITCHING_SO & TEAM_BATTING_SO == 1.0 correlation

However, we must consider that these initial correlation values could be influenced by the fact that missing values and outliers have yet to be addressed.

Data Preparation

Renaming Column Names

Keeping column names short and readable is important in order to practice “table hygiene”. Therefore, new column names were generated and are shown on Table XX.

Original Column Name	New Column Name
TARGET_WINS	target
TEAM_BATTING_H	bat_h
TEAM_BATTING_2B	bat_2b
TEAM_BATTING_3B	bat_3b
TEAM_BATTING_HR	bat_hr
TEAM_BATTING_BB	bat_bb
TEAM_BATTING_HBP	bat_hbp

Original Column Name	New Column Name
TEAM_BATTING_SO	bat_so
TEAM_BASERUN_CS	bas_cs
TEAM_FIELDING_E	f_e
TEAM_FIELDING_DP	f_dp
TEAM_PITCHING_BB	p_bb
TEAM_PITCHING_H	p_h
TEAM_PITCHING_HR	p_hr
TEAM_PITCHING_SO	p_so

Dealing with Missing Values

There are 6 features that have missing values:

- Strikeouts by batters (5%): Should use median or regression model for imputation
- Stolen bases (6%): Stolen bases weren't tracked officially until 1887, which means some of the missing data could be from 1871-1886. These values could be imputed.
- Caught stealing (34%): Stolen bases weren't tracked officially until 1887, so some of the missing data could be from 1871-1886. These values could be imputed.
- Batter hit by pitch (92%): This predictor will be removed from the analysis as too many of its values are missing.
- Strikeouts by pitchers (4%): Should use median or regression model for imputation
- Double plays (12%): Should use median or regression model for imputation

Tabachnick and Fidell

In general, imputations by the means/medians is acceptable if the missing values only account for 5% of the sample. Peng et al.(2006) However, should the degree of missing values exceed 20% then using these simple imputation approaches will result in an artificial reduction in variability due to the fact that values are being imputed at the center of the variable's distribution.

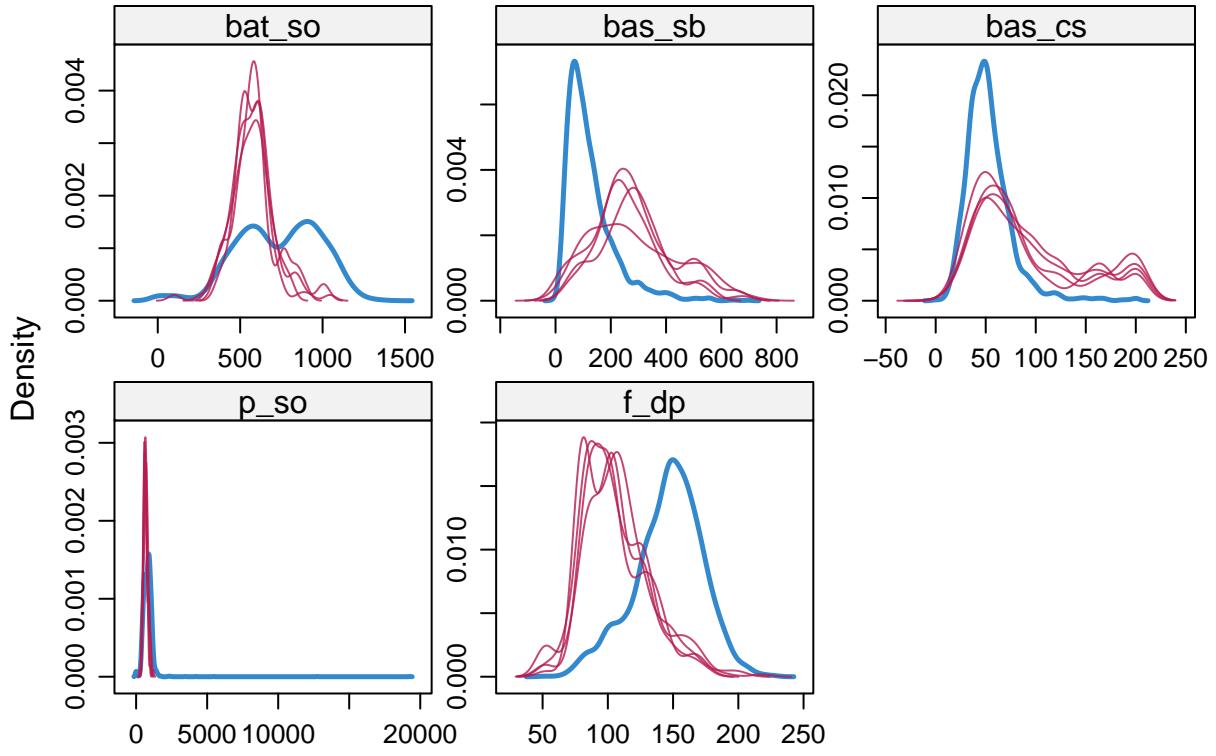
Our team decided to employ another technique to handle the missing values: Multiple Regression Imputation using the MICE package.

The MICE package in R implements a methodology where each incomplete variable is imputed by a separate model. Alice points out that plausible values are drawn from a distribution specifically designed for each missing datapoint. Many imputation methods can be used within the package. The one that was selected for the data being analyzed in this report is PMM (Predictive Mean Matching), which is used for quantitative data.

Van Buuren explains that PMM works by selecting values from the observed/already existing data that would most likely belong to the variable in the observation with the missing value. The advantage of this is that it selects values that must exist from the observed data, so no negative values will be used to impute missing data. Not only that, it circumvents the shrinking of errors by using multiple regression models. The variability between the different imputed values gives a wider, but more correct standard error. Uncertainty is inherent in imputation which is why having multiple imputed values is important. Not only that. Marshall et al. 2010 points out that:

“Another simulation study that addressed skewed data concluded that predictive mean matching ‘may be the preferred approach provided that less than 50% of the cases have missing data...’

Removal of bat_hbp- NA



Following use of the MICE package, we can visualize the distributions of the imputed versus existing data points as shown on Figure 6. The density of the imputed data for each imputed dataset is shown in magenta. The density of the observed data is shown in blue. For the MICE algorithm, the number of multiple imputations was set to five. The imputed distribution for `bas_sb` and `p_so` look close to the original data distribution which is good. The imputed data distributions for the other variables do not match so closely to the original data. Reasons include:

- Some of the variables are bimodal in nature (which is why in `bas_cs` for example, there is bimodality in the imputed distributions).
- 34% of the data for `bas_cs` is missing, which is above 5%, while the missing data for `p_so` only makes up 4% of the total amount of missing data for that predictor.
- 12% of the data for `f_dp` is missing, which is above 5%, while the missing data for `p_so` only makes up 4% of the total amount of missing data for that predictor.

Analysis of Outliers

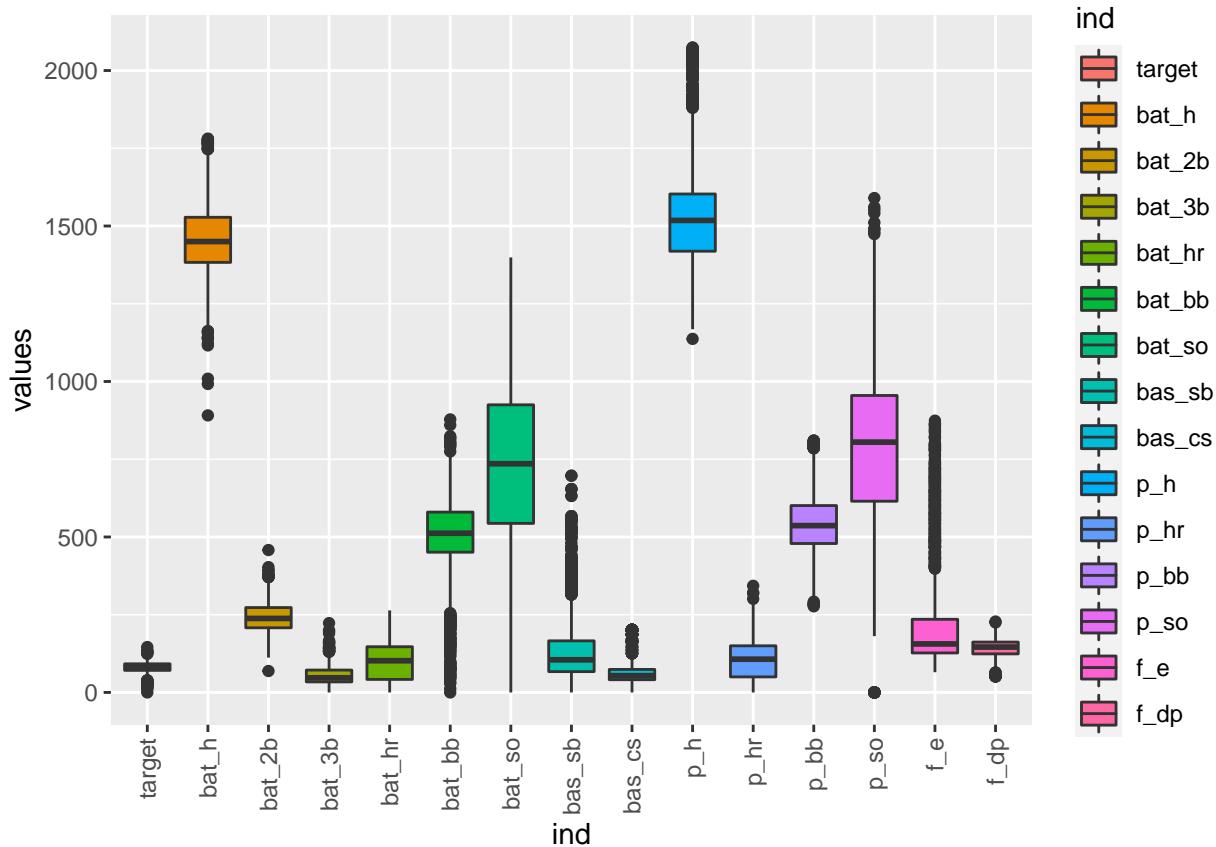
Several predictors contained outliers that contradicted with existing baseball statistics or fell out of an “acceptable” range given the feature’s inherent distribution. These features are:

- `bat_h`: The most hits by team in a season is 1783. Therefore, any values above 1,783 were replaced with the median for the predictor (Source).
- `p_h`: We could not find any suitable statistics from outside sources for this feature. However, we can apply interquartile outlier analysis. By analyzing a given feature, those datapoints which fall above or below an “acceptable” range can be identified given the features inherent distribution.

- p_so: The record for most strikeouts in a season is 1595. Anything above this should be removed or imputed (Source).
- f_e: The record for most errors in a season is 886. Anything above this should be removed or imputed (Source).
- p_bb: We could not find any suitable statistics from outside sources for this feature. However, we can apply interquartile outlier analysis. By analyzing a given feature, those datapoints which fall above or below an “acceptable” range can be identified given the features inherent distribution.

After replacing the above outliers, we can visualize the improved distributions by use of a boxplot.

Distributions after outlier analysis and imputing NA Values

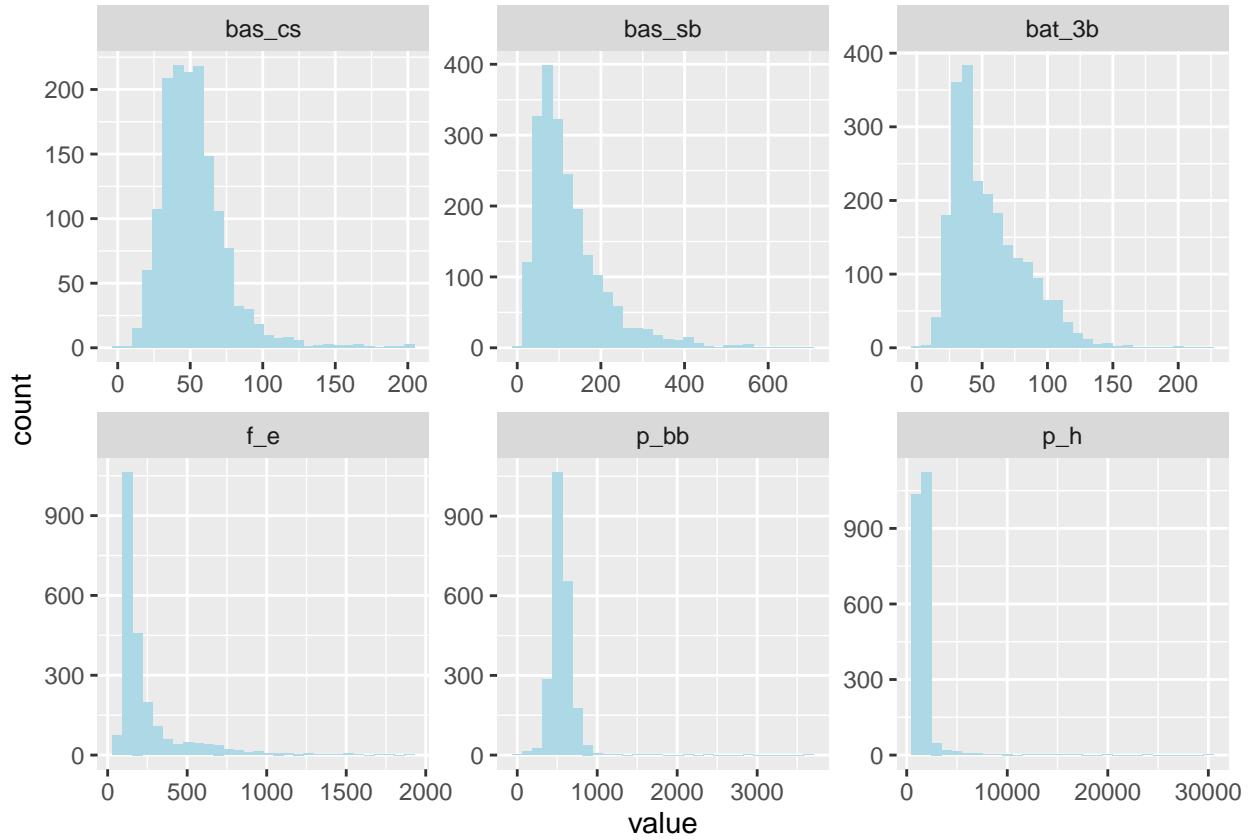


While there are still outliers present in the dataset, particularly for bas_sb and f_e, we can see a large improvement from before. All features are within the range 0-2500. We can attempt to further deal with outliers should the need arise, but for now we will accept this distribution.

Box-Cox Transformation for skewed variables

Based on the previous distribution plot (using histograms) we noticed that a select group of columns exhibited non-normal skew. In particular, the following columns showed signs of left-skew:

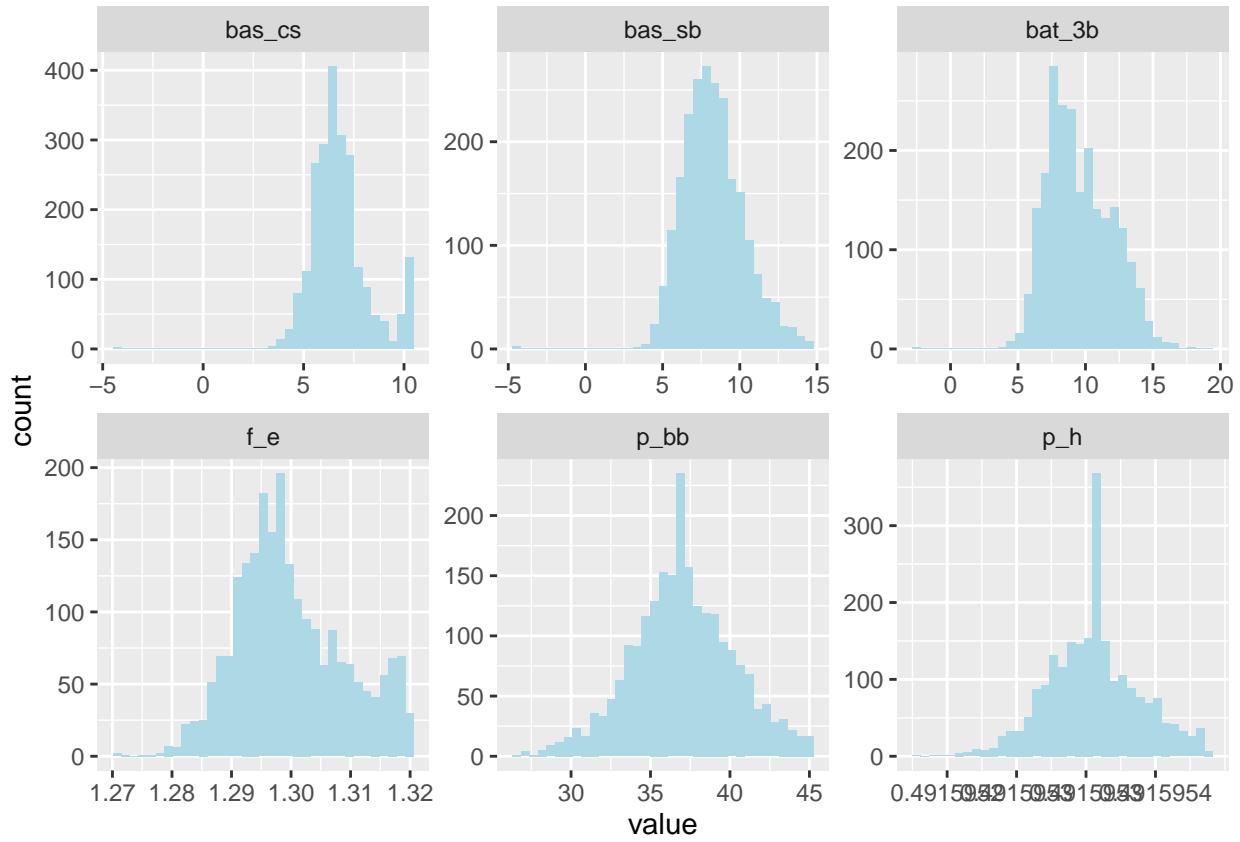
Skewed variables



In order to address this skewness and attempt to normalize these features for future modeling, we will employ box-cox transformations. Because some of these values include 0, we will need to replace any zero values with infinitesimally small, non-zero values.

The λ 's that were used to transform the skewed variables are shown on Table 2.

Histograms for transformed variables



λ 's for skewed variables

Column Name	λ
bat_3b	0.400
bas_sb	0.220
bas_cs	0.232
f_e	-0.753
p_bb	0.460
p_h	-2.034

As we can see from the above, the boxcox transformations on the selected features performed extremely well. We can see that all features included now exhibit normal or near-normal distributions around their respective centers.

Dealing with Bimodal Variables

Bimodal distributions in data are interesting, in that they represent features which actually contain multiple (2) inherent systems resulting in separated distributional peaks. Figure XX shows that bimodality is present in `bat_so`, `p_hr`, `bat_hr`. While a Box-Cox transformation could have been undertaken in order to transform the bimodal variables to a normal distribution. However, this throws away important information that is

inherent in the bimodal variable itself. The fact that the variable is bimodal in the first place is essentially ignored, and the predicted values in the linear multiple regression model will not reflect this bimodality.

Our approach to solving this is to create dummy variables representing which side of the local minimum each data point falls with respect to its original bimodal distribution. First, two histograms were fit to these variables using the `mixtools` package. Then, the intersection point between the two histograms was determined by solving for c . Where

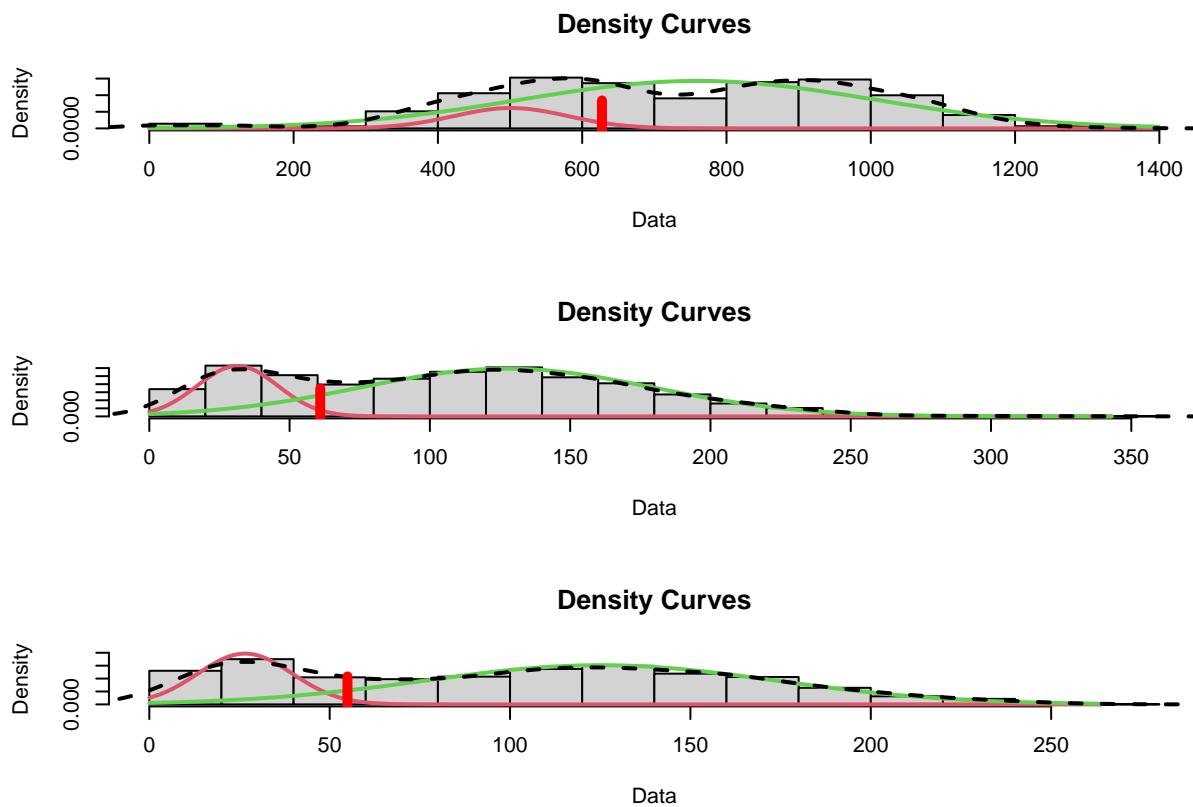
$$c = \frac{\mu_2\sigma_1^2 - \sigma_2(\mu_1\sigma_2 + \sigma_1)\sqrt{(\mu_1 - \mu_2)^2 + 2(\sigma_1^2 - \sigma_2^2)\log\frac{\sigma_1}{\sigma_2}}}{\sigma_1^2 - \sigma_2^2}$$

Where μ_1 and σ_1 are the mean and standard deviation for the left distribution and μ_2 and σ_2 are the mean and standard deviation for the right distribution.

A new variable was created for each bimodal predictor, where any observed values below c would be assigned a value of 0, while any observed values above c would be assigned a value of 1. For example, c for `bat_so` was calculated to be 806.39. `bi_bat_so` is a new dummy variable that was created where any values above 806.39 in the original `bat_so` data were assigned a value of 0, while values below 806.39 were assigned a value of 1. The λ 's for the three bimodal variables are shown in Table XX. The counts for the unique values are shown in each dummy variable are shown on the barcharts on Figure XXX.

Finds where two histograms intersect and Creates dummy variables based on bimodal data:

Density curves for each bimodal predictor with two normal distributions fit to each peak.

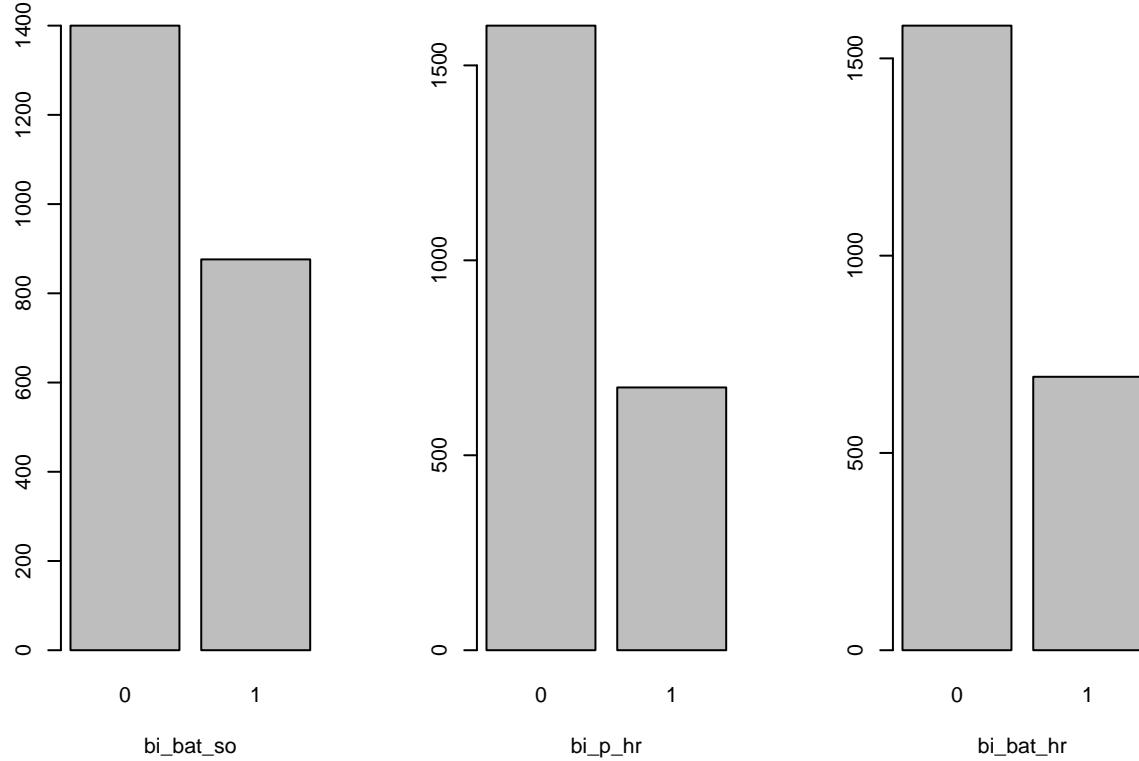


Summary of bimodal dummy variable generation

Column	μ_1	μ_2	σ_1	σ_2	c	Count of 0's
bat_so	606.31	972.61	199.88	114.06	806.38	969
p_hr	31.43	127.37	14.39	52.08	60.93	1602
bat_hr	26.55	125.06	13.10	48.72	54.93	1583

Bar graphs for each of the bimodal dummy variables.

0 represents the amount of observations for the original variable where the value was above c , while 1 represents the amount of observations below c^*



Saber Model

We would like to employ outside analysis in order to engineer new, potentially powerful features. Popularized in the movie “Money ball”, the SABERMETRICS model for baseball analysis includes a feature known as BsR (base runs). This statistic estimates the amount of runs a team should score.

Base Runs (BsR) is a run estimator developed by David Smyth in the early 1990s. It was inspired by Bill James’ Runs Created, and like Runs Created it is a dynamic model of team run scoring. Many different versions of Base Runs have been introduced by several sabermetricians, to accommodate different data sets and philosophies. However, all Base Runs formulas take the form:

$$A \cdot B / (B + C) + D$$

$$BSR = AB/(B + A) + C$$

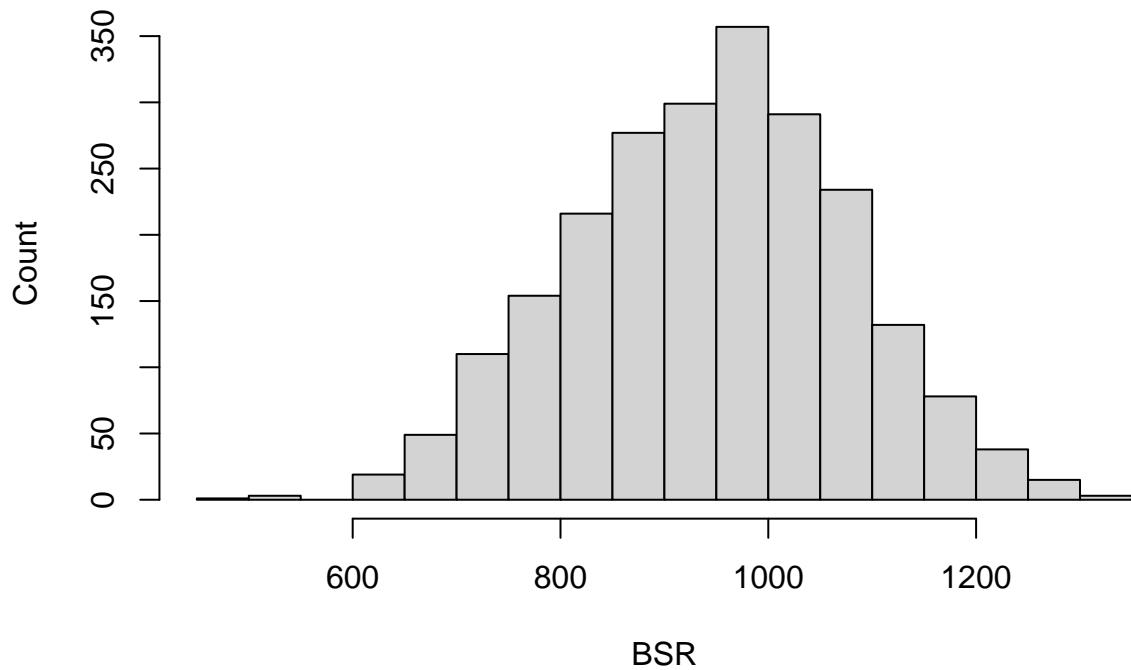
where:

$$A = TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_BB$$

$$B = 1.02(1.4TEAM_TOTAL_BASES - 0.6TEAM_BATTING_H + 0.1TEAM_BATTING_BB)$$

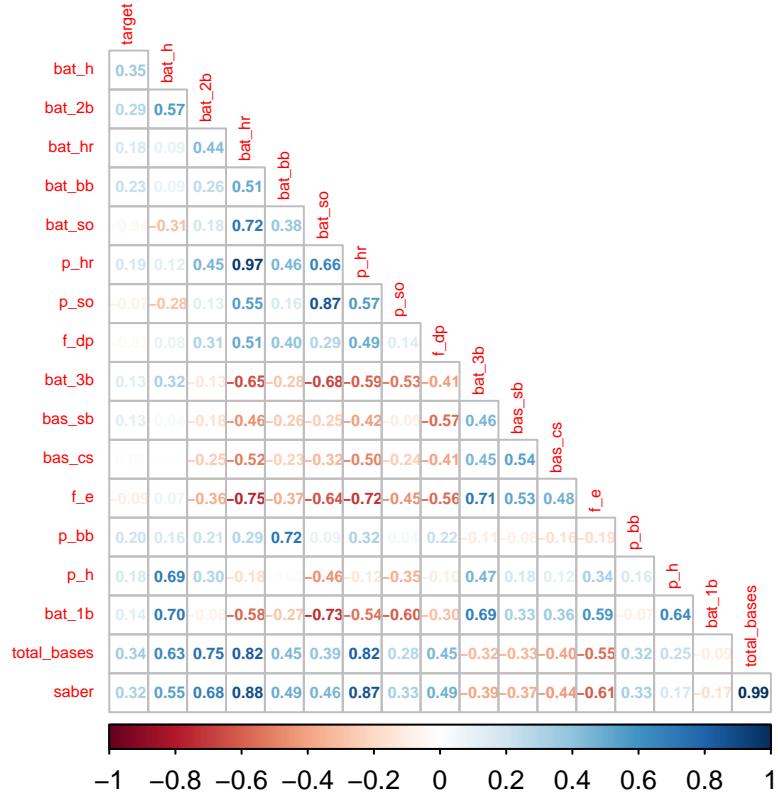
$$C = TEAM_BATTING_HR$$

Histogram of BSR Predictor



Reviewing the correlations

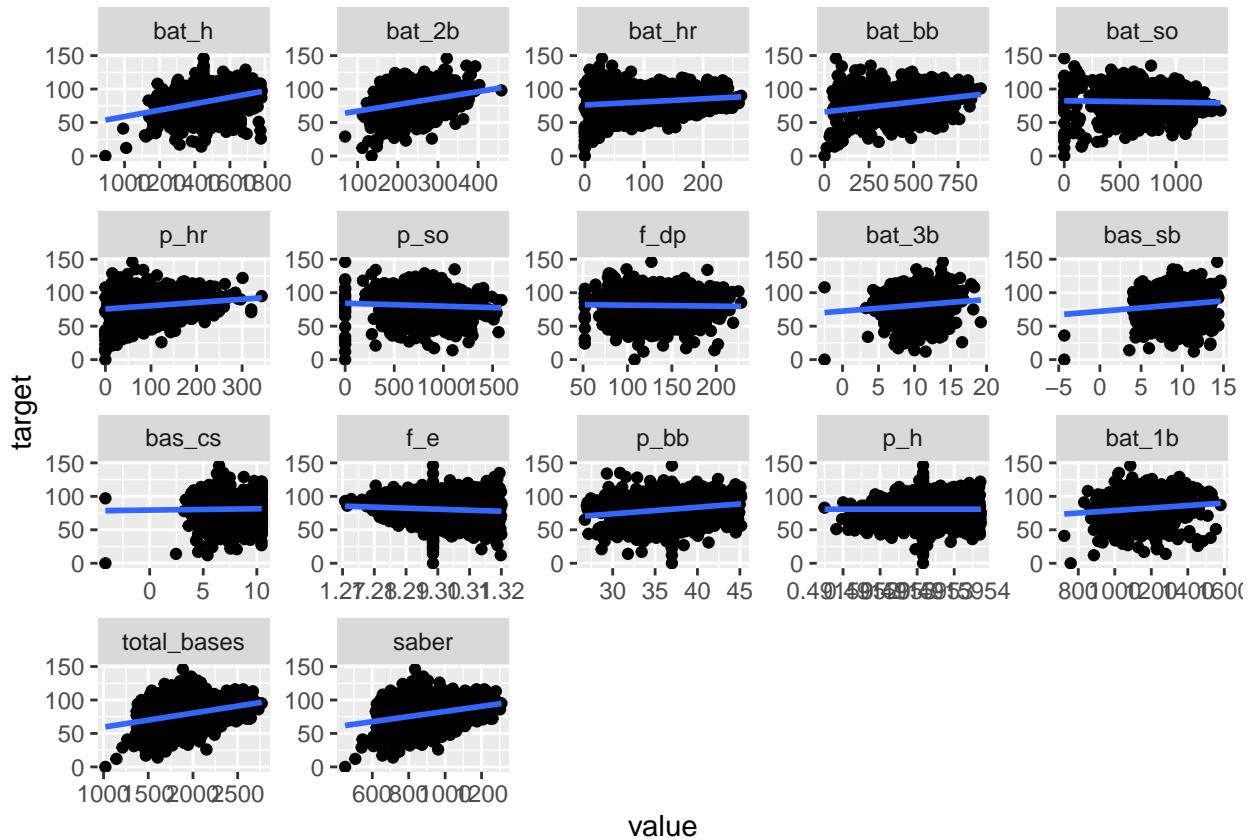
After performing multiple cleaning and imputation steps, we would like to visualize again the correlations between features and their target, as well as between features themselves.



These correlation values make much more sense than before. We can see that features no longer have 1.0 correlations, which in general are highly unlikely to occur naturally. The new most correlated (and least correlated) features are as follows:

- p_hr & bat_hr (0.97): This is an interesting correlation, as we would not have initially expected the amount of home runs allowed to be correlated with the number of home runs achieved from a team. However, one could make the argument that a team which focuses on offense would similarly be lacking in defense.
- bat_1b & bat_so (-0.73): These features are negatively correlated, which makes intuitive sense. If a team has many players making it to base, then conversely we would expect that this team would have less strikeouts at bat.
- bat_so & p_so (0.87): These features intuitively should not have such high correlation. Similar to above, we would not expect the performance of batter strikeouts to have any relationship to the performance of pitching strikeouts on the same team.

Target correlation plot after cleaning



After applying all transformations and imputations, we can see that the feature correlation with the target variable has also improved. Features predicted to have positive correlations (as provided by the assignment guide) do tend to have positive correlations. Similarly, features with expected negative correlations behave as described. This provides us some level of validation as we take the next steps with model building.

Build Models

Create data frame with 0 rows and 3 columns and Create function to update the tracker

Examine base model, no transformations, no engineering

Our first model (Base model) will use all of the initially provided columns, after cleaning and imputation. We will use the results of this model to understand a baseline for our future model development.

```
## 
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 6.9253e+02 7.4378e+01 9.3110 < 2.2e-16
## bat_h       1.8255e-02 3.8878e-03 4.6955 2.819e-06
## bat_2b      2.7989e-02 8.6740e-03 3.2268 0.0012697
## bat_hr      7.1466e-02 2.8495e-02 2.5080 0.0122103
## bat_bb      3.9873e-02 4.5933e-03 8.6807 < 2.2e-16
## bat_so     -1.9114e-02 4.1948e-03 -4.5566 5.476e-06
```

```

## p_hr      1.5501e-02 2.4562e-02 0.6311 0.5280260
## p_bb     -5.1417e-01 1.4920e-01 -3.4461 0.0005791
## p_so      1.4940e-03 3.2302e-03 0.4625 0.6437681
## f_dp     -1.0332e-01 1.3232e-02 -7.8084 8.784e-15
## bat_3b    1.4218e+00 2.0473e-01 6.9449 4.930e-12
## bas_sb    1.8485e+00 2.0479e-01 9.0267 < 2.2e-16
## bas_cs    1.3926e-01 2.5226e-01 0.5520 0.5809747
## f_e       -5.0564e+02 5.6774e+01 -8.9062 < 2.2e-16
##
## n = 2276, p = 14, Residual SE = 13.37938, R-Squared = 0.28

```

Based on the above output, we can see that this model performs relatively poorly against the training data. However, as this is our base model, we will assess the performance of all future models against this value. Moving forward, if we can lift the Adjusted r^2 to above 0.3, we will consider it a general improvement.

Evaluate SABER model

The next model we would like to evaluate is the SABER model. Here we will use all original features, and additionally we will include the engineered SABER metrics. Hopefully we will see a lift in performance after utilizing these industry-derived features.

```

##
## Coefficients: (3 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.8848e+02 7.4296e+01 9.2667 < 2.2e-16
## bat_h      -2.1195e-01 5.0614e-02 -4.1875 2.928e-05
## bat_2b     -1.0044e-01 2.9629e-02 -3.3899 0.0007111
## bat_hr     -7.1592e-01 1.7701e-01 -4.0446 5.416e-05
## bat_bb      2.5793e-02 5.7484e-03 4.4870 7.586e-06
## bat_so     -2.5209e-02 4.4272e-03 -5.6942 1.401e-08
## p_hr      -2.3380e-02 2.5598e-02 -0.9134 0.3611534
## p_so       4.6712e-03 3.2620e-03 1.4320 0.1522772
## f_dp      -1.1571e-01 1.3448e-02 -8.6043 < 2.2e-16
## bat_3b     1.0546e+00 2.1917e-01 4.8115 1.597e-06
## bas_sb     1.9034e+00 2.0605e-01 9.2376 < 2.2e-16
## bas_cs     2.5214e-01 2.5180e-01 1.0014 0.3167610
## f_e       -5.0468e+02 5.6876e+01 -8.8733 < 2.2e-16
## p_bb      -3.8029e-01 1.5221e-01 -2.4984 0.0125459
## bi_bat_so1 -1.7174e+00 1.0639e+00 -1.6143 0.1065933
## bi_p_hri   -3.4765e+00 1.9107e+00 -1.8195 0.0689662
## bi_bat_hri  5.8917e+00 2.0429e+00 2.8839 0.0039649
## saber      4.8640e-01 1.0613e-01 4.5829 4.837e-06
##
## n = 2276, p = 18, Residual SE = 13.28237, R-Squared = 0.29

```

As expected, we did see a lift in performance after including SABER metrics. However, the lift was hardly significant. We are still below 0.3 Adjusted R^2 .

SABER reduced

Here we will test out a more parsimonious version of the above SABER model. In the spirit of simplifying the model for human use and understanding, we will select only the features that have high significance

from the above SABER model. Additionally, we will exclude any features which were included as part of the construction of SABER, in order to reduce inherent multicollinearity.

```
##           Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 4.8049e+02 6.8429e+01 7.0217 2.884e-12  
## saber      5.2232e-02 3.1766e-03 16.4425 < 2.2e-16  
## bi_bat_hri  3.0012e-02 1.0873e+00 0.0276    0.978  
## f_e        -3.4661e+02 5.1743e+01 -6.6987 2.643e-11  
## bas_sb     2.1296e+00 1.9113e-01 11.1425 < 2.2e-16  
## f_dp       -1.0406e-01 1.3536e-02 -7.6882 2.208e-14  
## bat_so     -2.0973e-02 1.6315e-03 -12.8553 < 2.2e-16  
## bat_bb     2.7673e-02 2.8093e-03  9.8504 < 2.2e-16  
##  
## n = 2276, p = 8, Residual SE = 13.56093, R-Squared = 0.26
```

While the Adjusted R² has been slightly reduced to 0.26, we have also significantly reduced the complexity of the model. This provides value in itself, as the model can be more easily distributed to players and coaches.

Step AIC

Step AIC works by deselecting features that negatively affect the AIC, which is a criterion similar to the R-squared. It selects the model with not only the best AIC score but also a model with less predictors than the full model, since the full model may have predictors that do not contribute or negatively contribute to the model's performance. The direction for the Step AIC algorithm was set to `both`, because this implements both forward and backward elimination in order to decide if a predictor negatively affects the model's performance.

```
##           Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 6.8681e+02 7.2984e+01 9.4104 < 2.2e-16  
## bat_h      1.8176e-02 3.8290e-03 4.7469 2.195e-06  
## bat_2b     2.8459e-02 8.5675e-03 3.3218 0.0009086  
## bat_hr     8.6889e-02 9.9994e-03 8.6895 < 2.2e-16  
## bat_bb     3.7782e-02 3.9872e-03 9.4760 < 2.2e-16  
## bat_so     -1.7546e-02 2.2134e-03 -7.9270 3.491e-15  
## p_bb      -4.5775e-01 1.3582e-01 -3.3703 0.0007635  
## f_dp      -1.0287e-01 1.3187e-02 -7.8004 9.342e-15  
## bat_3b     1.4629e+00 2.0035e-01 7.3020 3.905e-13  
## bas_sb     1.8969e+00 1.9331e-01 9.8128 < 2.2e-16  
## f_e        -5.0186e+02 5.5816e+01 -8.9913 < 2.2e-16  
##  
## n = 2276, p = 11, Residual SE = 13.37488, R-Squared = 0.28
```

Square Root Step AIC

The following model was generated using the same AIC methodology, except that the `target` variable was square rooted.

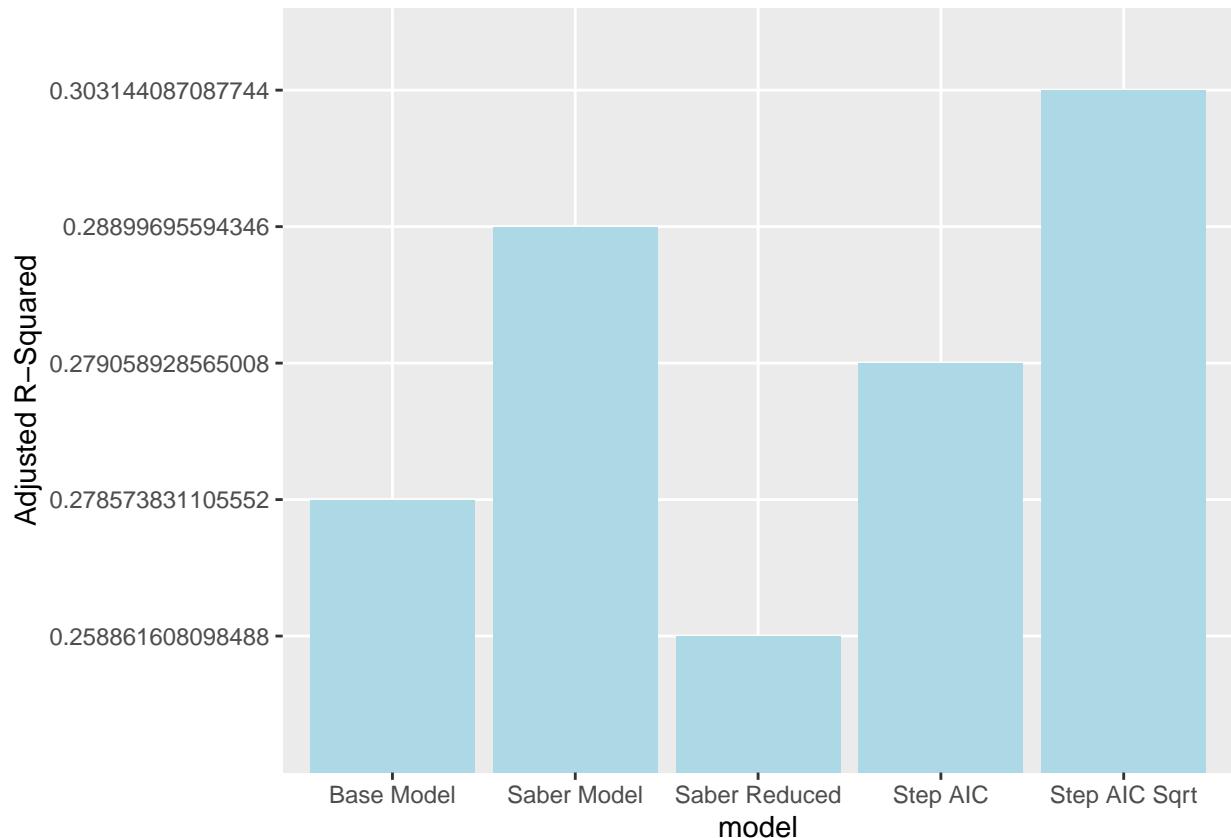
```
##           Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 4.2631e+01 4.2723e+00 9.9785 < 2.2e-16  
## bat_h      -1.2958e-02 2.9130e-03 -4.4484 9.072e-06  
## bat_2b     -6.5019e-03 1.7158e-03 -3.7894 0.0001550
```

```

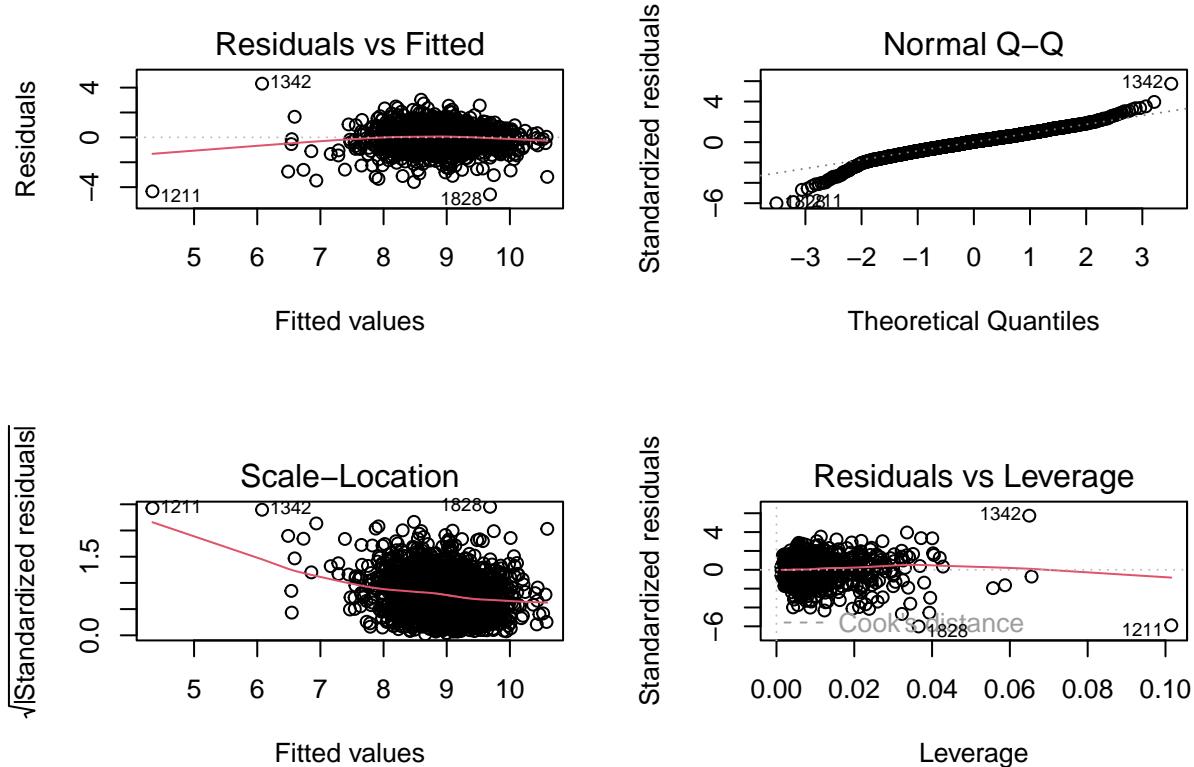
## bat_hr      -4.6426e-02  1.0289e-02 -4.5123  6.741e-06
## bat_bb      1.7533e-03  3.1617e-04  5.5454  3.275e-08
## bat_so     -9.7840e-04  1.5165e-04 -6.4517  1.349e-10
## f_dp       -6.4855e-03  7.8615e-04 -8.2498  2.661e-16
## bat_3b      6.3764e-02  1.2722e-02  5.0123  5.796e-07
## bas_sb      1.1348e-01  1.2044e-02  9.4223 < 2.2e-16
## bas_cs      2.7857e-02  1.4665e-02  1.8996  0.0576168
## f_e        -2.8383e+01  3.2744e+00 -8.6683 < 2.2e-16
## p_bb       -3.0901e-02  8.2592e-03 -3.7414  0.0001876
## bi_bat_so1 -9.6903e-02  6.2247e-02 -1.5567  0.1196707
## bi_p_hr1   -1.8528e-01  1.0787e-01 -1.7176  0.0860074
## bi_bat_hr1  3.1635e-01  1.1674e-01  2.7099  0.0067814
## saber      3.0081e-02  6.1195e-03  4.9156  9.484e-07
##
## n = 2276, p = 16, Residual SE = 0.77728, R-Squared = 0.31

```

Model Selection



The model that is ultimately chosen for this analysis is Step AIC Square Root. We were able to increase over the base model by 3%. AIC is a measure of multicollinearity so the selection process parsed out variables that were highly colinear with other variables, giving us a model that has the lowest AIC values based on a select number of predictors. This is important because this model needs to be used and understood by professionals in the industry; the step AIC model ensures that only the most prominent features are included.



The QQ plot shows that the data is centered in the middle, but there is significant amount of residuals in the middle of the distribution. This is known as the “thin tail” phenomenon. Normal distributions with “thin tails” correspond to the first quantiles occurring at larger than expected values and the last quantiles occurring at less than expected values. Notice that the “thin tailed” Q-Q plot is a reflection of a “fat tailed” Q-Q plot, which is the opposite phenomenon, across the X-Y diagonal. The Residuals vs. Fitted, and scale-location plots show that the residuals are mostly centered around the zero line. However there is some skewness as a result of outliers that are marked numerically on the plot itself. The Residuals vs. Leverage plot shows that there are no residual values that exceed Cook’s distance, which is good.

Important Metrics for Step AIC Square Root Model

Important metrics for the Step AIC Square Root model are:

- R-squared: 0.3027
- F-statistic: 71.54
- RSS: 1366.88
- MSE: 0.6
- RMSE: 0.774

Predictions on Evaluation Set

The predictions were generated using the evaluation set on the Square Root Step AIC model. These predictions are provided in the `predictions.csv` file.

Appendix

The R code that was used in the analyses shown in this report can be viewed here: “RMD code”.

- Thank you.