Unit 1 Assignment: “Moneyball Baseball Problem”

Alan Kessler, MSDS 411, Section 56

Intro

The purpose of the assignment is to analyze baseball season statistics at the team level to predict the number of wins for a given team. A variety of multivariate linear regression models are evaluated, and a final model is chosen based on model fit statistics and diagnostic plots. The final model considers many of the attributes related to batting, fielding, and baserunning including interactions. Missing and incorrect data is shown to make a significant impact on the ability to use pitching variables and requires transformations of other variables.

Bonus

For bonus points, I used the glm function to show that the results match the lm results for the default glm options. Additionally, I used decision trees to impute missing values and a random forest to assist in variable selection.

Data Exploration

The data for this assignment contains summary information about individual team’s baseball statistics. After viewing a sample of the data and summary statistics by each variable, there are a number of missing values. Figure 1 shows the portion of values missing in the training data for variables with at least one missing value. These missing values are imputed in the Data Preparation section. The large percentage of missing TEAM_BATTING_HBP values indicate that even with imputation, this variable should not be considered in a model.

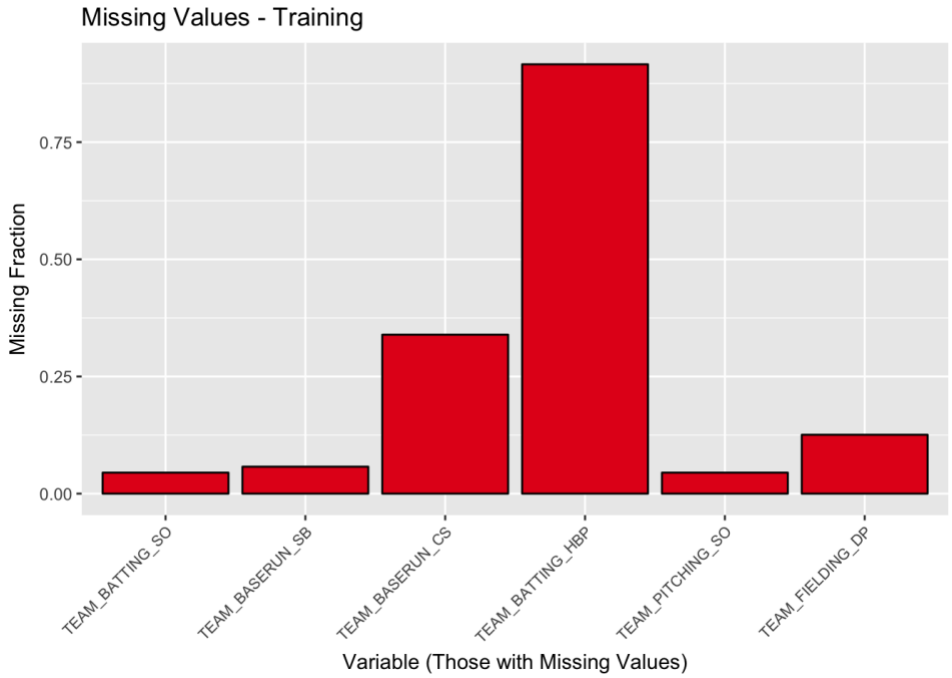

Figure 1 – Missing Value Frequency

Figure 2 shows that the target is approximately normally distributed with the exception of a heavy left-tail. The data lacks information on the number of games played, so this could be a case of shortened seasons having an impact. As a result, it is likely that a regression model will not be able to fit these situations particularly well.

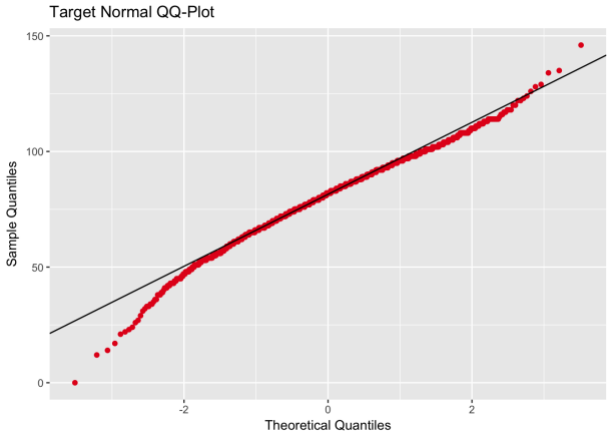

Figure 2 – Normal QQ-Plot of Training Data

Figure 3 shows the distribution of the predictor variables. Based on the box plots, it is clear that the testing data likely comes from the same population as the training data and that several of the variables are skewed. In addition to needing to adjust for skewness, the data presents outliers for which flooring or capping should be considered.

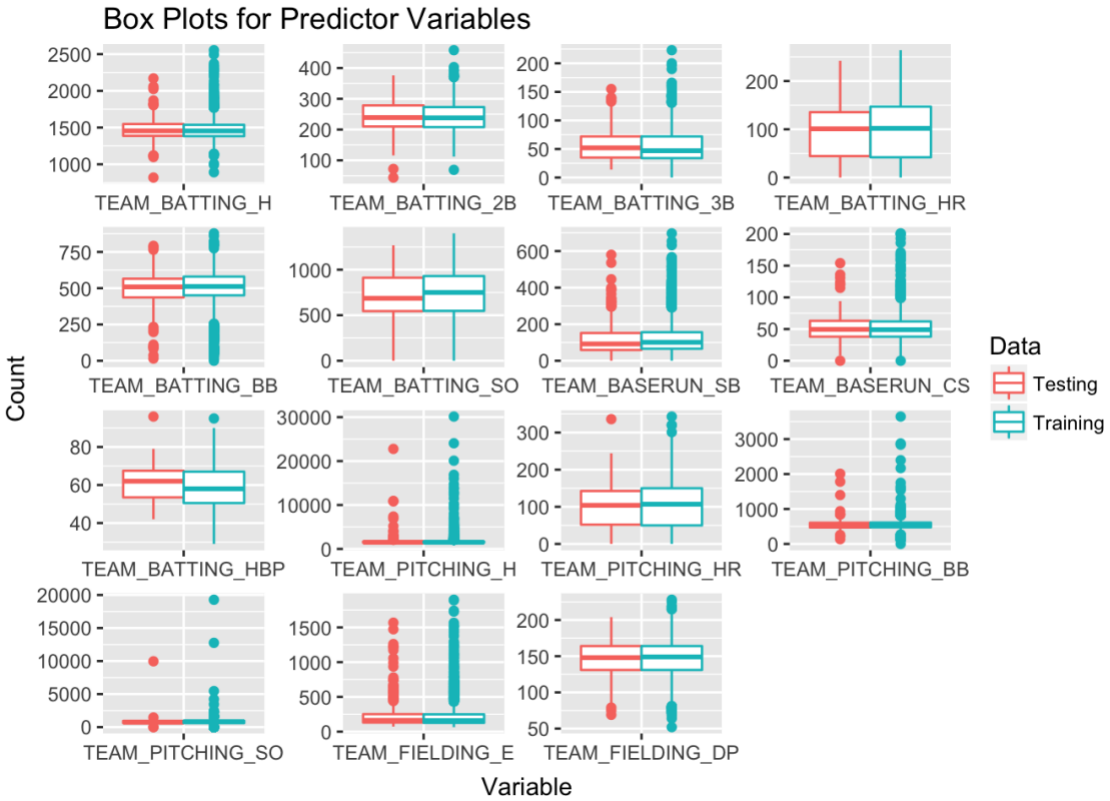

Figure 3 – Box Plots of Training and Testing Predictor Variables

Figure 4 shows the same data in the form of histograms. From this series of charts it appears that pitching data is skewed by outliers. Some of the predictor distributions also appear bimodal. This indicates that important information such as the number of games played in a particular season or era-specific characteristics are missing. If the year of the season was available, that context could adjust for historical baseball trends.

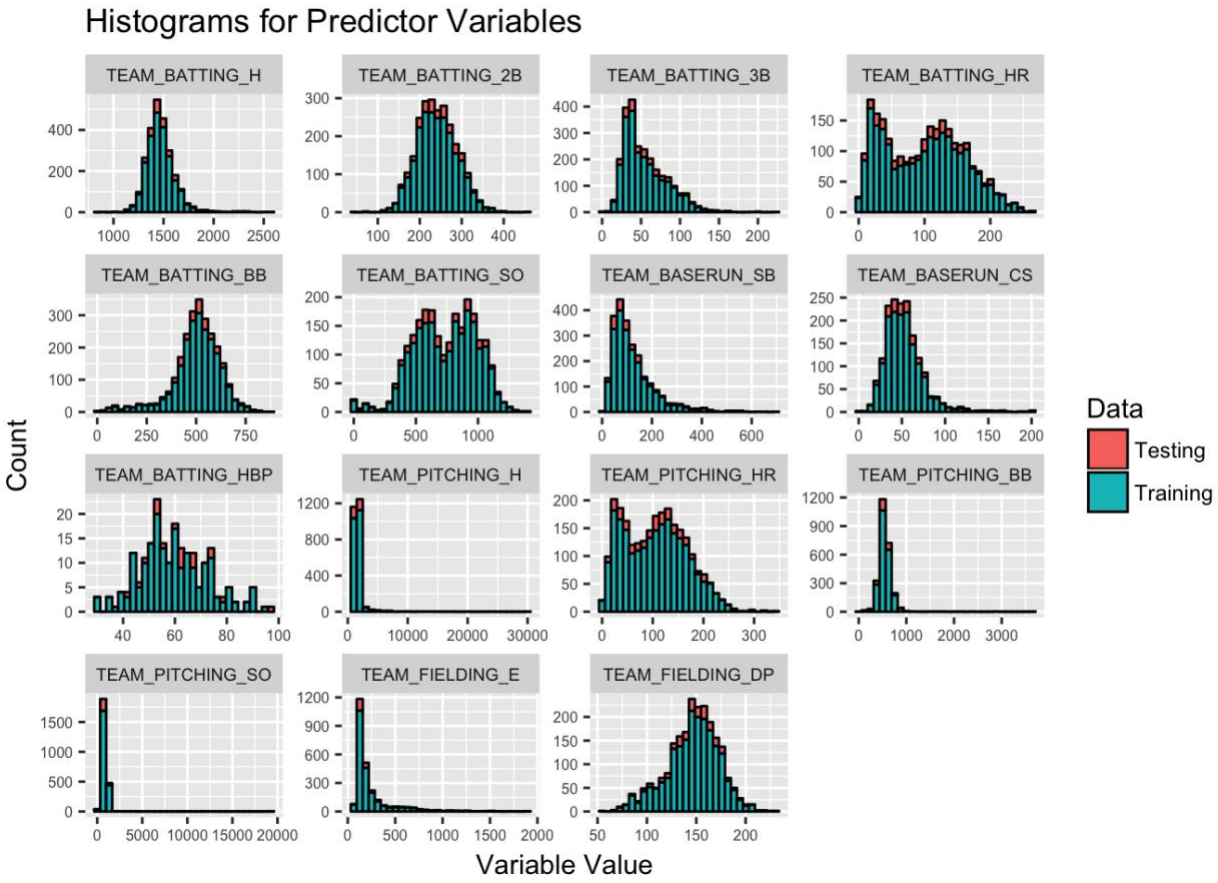

Figure 4 – Histograms of Training and Testing Data

The correlation matrix of target and predictors is shown in Figure 5. It demonstrates that predictors are correlated with the target, showing promise for building a predictive model. Additionally, most predictors are not strongly correlated with each other with the exception of the pitching and hitting variables (hits, home runs, walks, and strikeouts). Based on the relationship with the target, it would appear that the pitching data is inaccurate and should be dropped from consideration. For example, it shows that teams with more pitching walks result in more wins which is unintuitive. Considering this data is public record, it also raises doubts about building a model on this data and a simple model should be strongly considered.

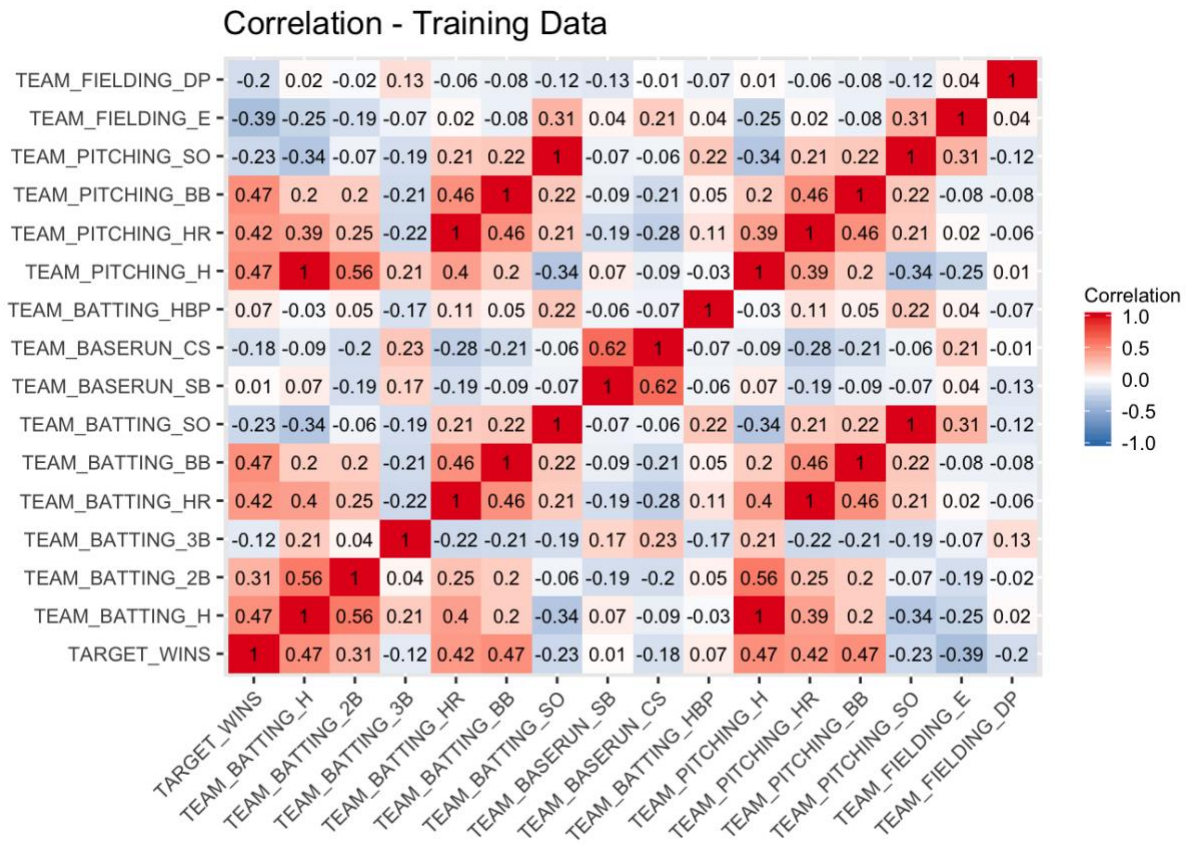

Figure 5 – Correlation in Training Data

Data Preparation

Prior to imputing missing values, observations with over 2,000 pitching hits are removed from the training data. These observations appear to occur when pitching and batting hits already identified as a potential error do not match. The pitching variables with this high correlation are then dropped from the training set.

Next, training observations are only kept for win totals of 20 to 116 as those are major league records and observations outside that range may influence the model undesirably. For the same reason, training data with an excess of 1,800 batting hits are removed as that is approximately a major league record in a season.

An additional variable for singles is created to avoid having a variable like doubles be a subset of hits. As a result, hits are not used in the model. The last step prior to imputation is to create missing value indicator variables. These variables are used in the modeling process to determine if the fact that a value is missing adds predictive value itself.

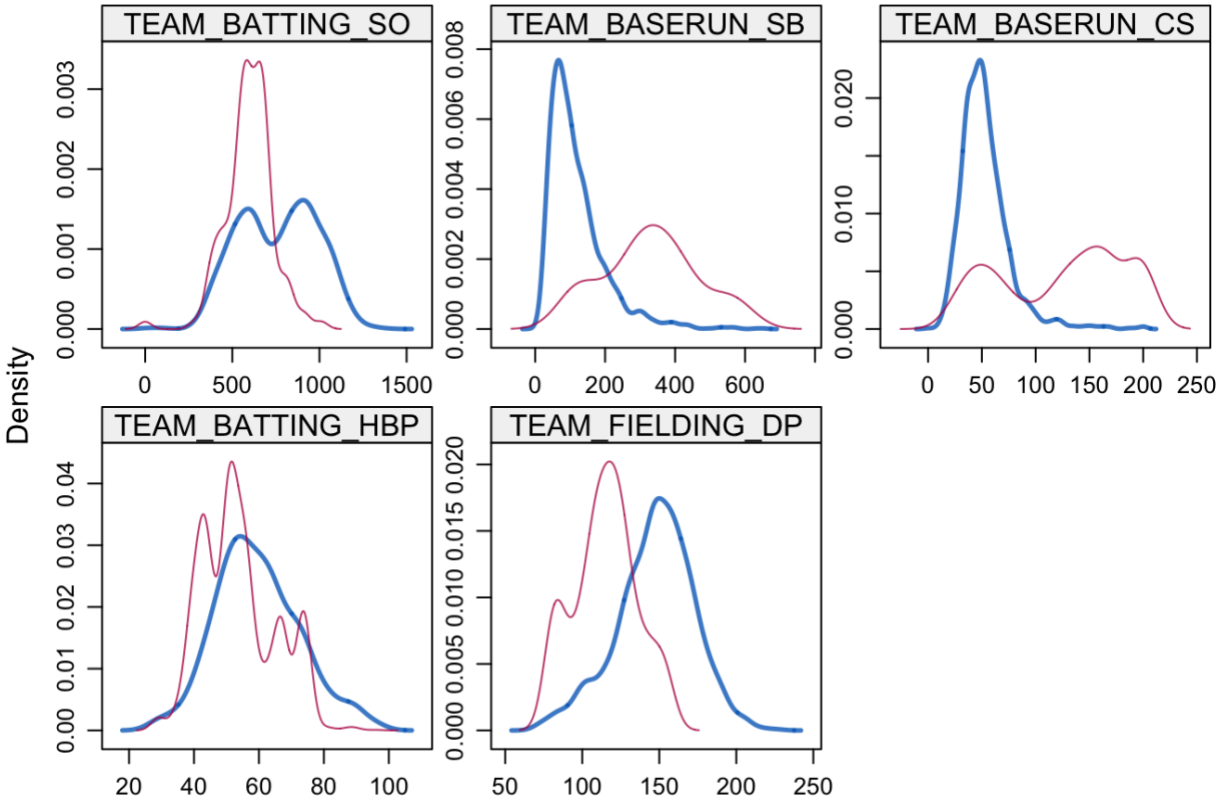

Figure 6 – Imputation Density Plots

The missing values are imputed using decisions trees within the MICE R package. Figure 6 shows the results of the imputation. The non-missing distribution is shown in blue and the values assigned to missing values are shown in red. The difference in these distributions is further evidence that the missing indicators may prove useful in a predictive model. Due to the way that the MICE package works, the imputation formula is not able to be saved for future use. That means that testing data and training data are combined together for imputation and then separated again for further model building. This works for the small size of data in this example but would not be practical for large data sets.

Next, the target and missing value indicator variables are added back to the now imputed data set. A new variable to represent stolen base success percentage is created based in part on the imputed values. This variable has a floor at 30% to effectively cap outliers.

For the other variables, many are skewed. For singles, triples, stolen bases, and errors, the log of the variable is used instead. To account for large outliers in the predictors, singles, doubles, triples, stolen bases, errors, and double plays are capped at the 99.5th percentile of the training data. As a final filter, seasons with zero batting strike outs are excluded from the training data.

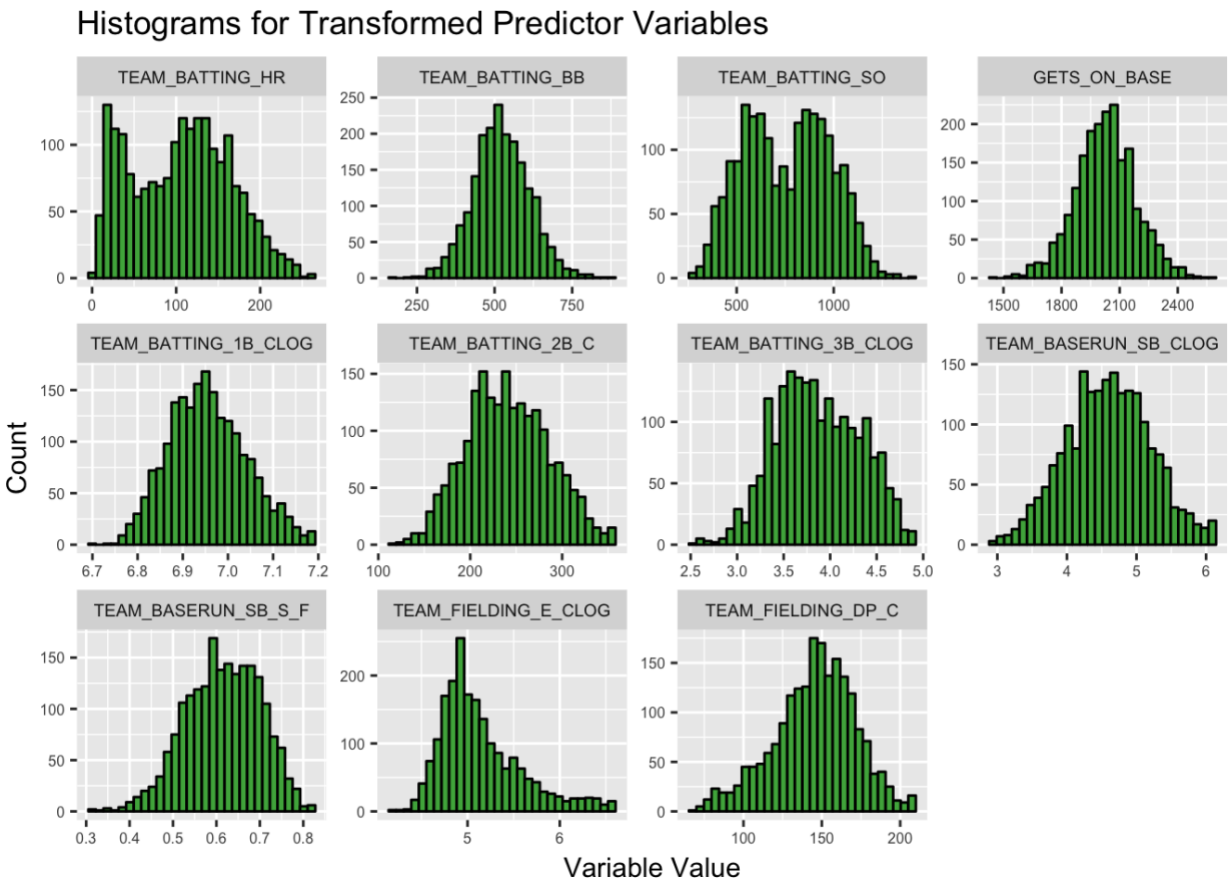

Figure 7 – Histograms of Transformed Training Data

The variable “Gets on Base” is created as the sum of hits, walks, and hit by pitches to create a baseline model discussed in the Build Models Section. The distributions of these final continuous variables are included in Figure 7. The charts show that the transformations do reduce the skewness in the data but the bimodal nature of some of the variables persist.

Figure 8 shows the correlation between these final variables. This check shows that variables highly correlated with each other have been successfully removed, so multicollinearity should be avoided. The correlation with the target also shows promise for predictive models.

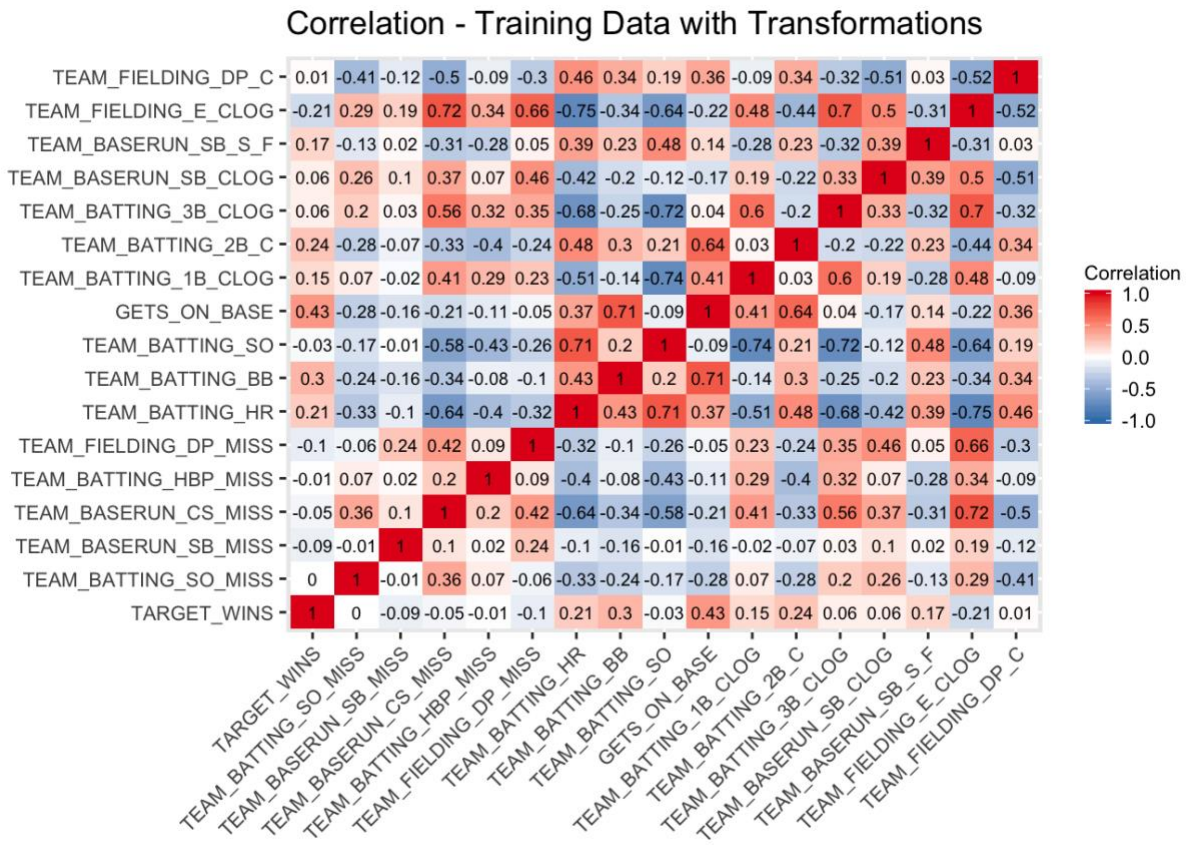

Figure 8 – Correlation in Transformed Training Data

Build Models

The first step in the model building process is to further partition the training set. This creates a new validation data set making up 30% of the total training data. The validation data provides another way to assess model performance and estimate how a model would perform on new data.

The first model, Model 1, is used as the baseline. It has a single predictor variable, “Gets on Base” as an homage to the movie where a player’s ability to get on base was stressed the best predictor of future performance. This variable is significant and the coefficient has an intuitive relationship with the target.

Model 2 contains all of the variables identified in the Data Preparation section and shown in Figure 8. The model appears to overfit as some of the variables are statistically insignificant and have unintuitive relationships with the target. While none of the variance inflation factors are greater than nine, there are variables contained in the model with factors greater than six which raises concerns regarding the stability of the model’s coefficients.

Model 3 applies stepwise variable selection to the variables included in Model 2. Reducing the number of variables retains the performance of the more complicated model but has small VIFs and more intuitive coefficient values.

Model 4 is the same model as Model 3, using the glm function. The default options for that function specify the identity link function and a gaussian error distribution. This matches linear regression exactly. To the point of the bonus problem, linear regression is a special case of a generalized linear model.

Model 5 first uses a random forest model to determine the top eight variables in the data: errors, home runs, walks, singles, triples, doubles, stolen bases, and strike outs. These variables are then used in a linear regression. This method would make sense when there are many variables but a simpler model is required. In this case, some of the coefficient values are still unintuitive.

Model 6 contains the same variables as Model 3 with additional interaction terms between home runs and stolen bases, and errors and strike outs. These interactions were selected by looking at the correlation between these variables and testing the impact of adding them to the model. The result is intuitive relationships between coefficients and the target. Figure 9 shows the model diagnostic plots for Model 6. These plots show that assumptions of linear regression are met, however the model does do a relatively poor job of fitting to extremely low win seasons. When building the stand alone scoring implementation version of the model, the scores should be capped at zero or a small number to avoid negative values.

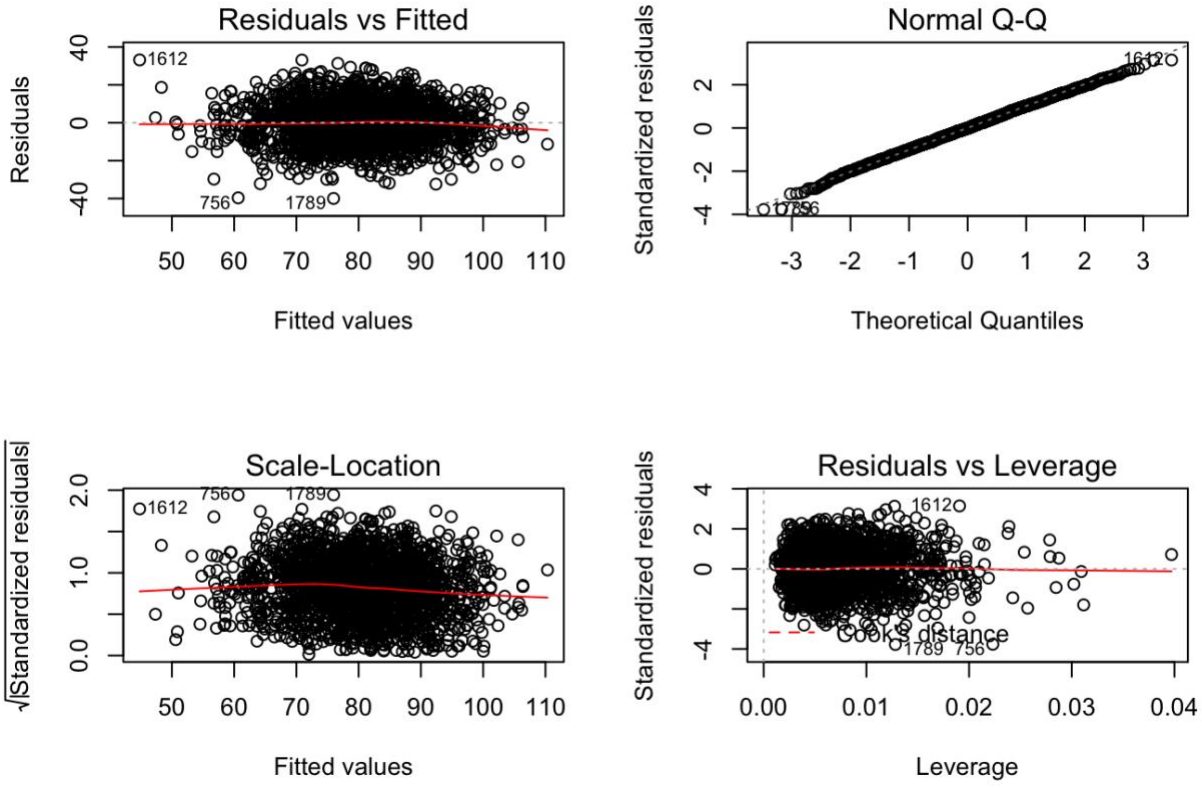

Figure 9 – Final Model Diagnostic Plots

Select Models

Figure 10 displays metrics of model performance. Based on adjusted r-squared, AIC, and mean squared error for both partitions of the training data, Model 6 is the best performing model.

This model contains the transformed versions of the following variables with signs in parentheses:

- TEAM_BATTING_HR (+)
- TEAM_BATTING_BB (+)
- TEAM_BATTING_SO (-)
- TEAM_BATTING_1B_CLOG (+) – Calculated variable for singles, log-transformed, and capped at the 99.5th percentile
- TEAM_BATTING_3B_CLOG (+) – log-transformed and capped at 99.5th percentile
- TEAM_BASERUN_SB_CLOG (+) – log-transformed and capped at 99.5th percentile
- TEAM_FIELDING_E_CLOG (-) – log-transformed and capped at 99.5th percentile
- TEAM_FIELDING_DP_C (-)
- –capped at 99.5th percentile
- TEAM_BATTING_SO_MISS (+) – Missing indicator for TEAM_BATTING_SO
- TEAM_BASERUN_CS_MISS (+) – Missing indicator for TEAM_BASERUN_CS
- TEAM_BATTING_HBP_MISS (+) – Missing indicator for TEAM_BATTING_HBP
- TEAM_BATTING_HR * TEAM_BASERUN_SB_CLOG (-)
- TEAM_BATTING_SO * TEAM_FIELDING_E_CLOG (+)

Once Model 6 is selected, the model is fit to the entire data set. This provides the benefit of having more accurate coefficients but also to check that the addition of other data does not drastically change the coefficients. In this case, the coefficients remain stable.

Model	Adj R-Squared	AIC	Training MSE	Validation MSE
Model 1	0.1936	11,119.09	156.7954	168.7245
Model 2	0.3976	10,737.33	117.2043	127.4816
Model 3	0.3971	10,736.49	117.3015	127.2394
Model 4	0.3971	10,736.49	117.3015	127.2394
Model 5	0.3459	10,839.38	127.2744	145.1014
Model 6	0.4253	10,669.03	111.8132	116.1702

Figure 10 – Model Performance Metrics

Conclusion

Six different models were fit to baseball season statistics data to predict the number of games a given team wins in a year. The data offered a substantial challenge with both missing and unintuitive data. These issues resulting in excluding some variables, transforming others, and dropping observations in the model building process. The final model used many of the available variables including interactions and generated predictions on a test data set where the variables were transformed in the same way. While the final model fits the data well, its coefficients can also illustrate the factors that make an individual team successful. Teams that succeed in common offensive categories such as home runs will likely win more games on average. In the future, accurate pitching data could enhance the model further.