Andrea Bruckner
Kaggle: abruckner

Moneyball

Introduction

In this report we are analyzing professional baseball team data from the years 1871 to 2006, inclusive, in order to predict the number of wins a baseball team can expect in a standard 162 game season. We will be using the Moneyball data set to build linear regression models to predict baseball game wins. In order to create a model that most accurately predicts the number of wins, we will prepare the data using imputation, truncation, and transformation, and we will select the variables using a variety of techniques, such as automated variable selection, correlation matrices, and decision trees. We will present several models, discuss the merits and shortcomings of each, and ultimately select one model that best predicts the number of baseball game wins a team has in a season.

Data Exploration

The Moneyball data set contains 2276 observations with 16 numerical variables and 1 index variable concerning the performance of professional baseball teams for a given year from the years 1871 to 2006, inclusive. Although the number of games in a season was not standardized until the 1960s, all of the statistics in this data set are adjusted to represent a team's performance during a standard 162 game season.

Variable	Label	Mean	Median	N	N Miss
INDEX		1268.46	1270.50	2276	0
TARGET_WINS		80.7908612	82.0000000	2276	0
TEAM_BATTING_H	Base Hits by batters	1469.27	1454.00	2276	0
TEAM_BATTING_2B	Doubles by batters	241.2469244	238.0000000	2276	0
TEAM_BATTING_3B	Triples by batters	55.2500000	47.0000000	2276	0
TEAM_BATTING_HR	Homeruns by batters	99.6120387	102.0000000	2276	0
TEAM_BATTING_BB	Walks by batters	501.5588752	512.0000000	2276	0
TEAM_BATTING_SO	Strikeouts by batters	735.6053358	750.0000000	2174	102
TEAM_BASERUN_SB	Stolen bases	124.7617716	101.0000000	2145	131
TEAM_BASERUN_CS	Caught stealing	52.8038564	49.0000000	1504	772
TEAM_BATTING_HBP	Batters hit by pitch	59.3560209	58.0000000	191	2085
TEAM_PITCHING_H	Hits allowed	1779.21	1518.00	2276	0
TEAM_PITCHING_HR	Homeruns allowed	105.6985940	107.0000000	2276	0
TEAM_PITCHING_BB	Walks allowed	553.0079086	536.5000000	2276	0
TEAM_PITCHING_SO	Strikeouts by pitchers	817.7304508	813.5000000	2174	102
TEAM_FIELDING_E	Errors	246.4806678	159.0000000	2276	0
TEAM_FIELDING_DP	Double Plays	146.3879397	149.0000000	1990	286

Table 1: Overview of Moneyball Data Set

In **Table 1**, we can see that most of the variables contain all 2276 data points. Of the 17 variables in the data set, only 6 variables have missing data: TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_SO, and TEAM_FIELDING_DP. Most of the variables with missing data can be fixed by imputing values for the missing values. Given that the means and medians are very similar within each variable with missing values, it will likely not matter much if we impute the missing values with either the mean or the median. If, however, the variables with missing values demonstrated a large difference between mean and median similar the differences exhibited by TEAM_PITCHING_H (1779.21 mean, 1518 median) or TEAM_FIELDING_E (246.48 mean, 159 median), we would more closely analyze the statistical output before deciding whether to use the mean or median. We will discuss imputation in greater depth in the next section. The only variable we cannot salvage from the data

set is the number of batters hit by pitches, TEAM_BATTING_HBP. More than 90 percent of its data is missing, so we will drop the variable from consideration when creating our models.

Pearson Correlation Coefficients, N = 2276							
Prob > |r| under H0: Rho=0							
	TARGET_WINS	**BATTING_H**	**BATTING_2B**	**BATTING_HR**	**BATTING_BB**	**PITCHING_HR**	**FIELDING_E**
TARGET_WINS	1	0.38877 <.0001	0.2891 <.0001	0.17615 <.0001	0.23256 <.0001	0.18901 <.0001	-0.17648 <.0001
BATTING_H Base Hits by batters	0.38877 <.0001	1	0.56285 <.0001	-0.00654 0.755	-0.07246 0.0005	0.07285 0.0005	0.2649 <.0001
BATTING_2B Doubles by batters	0.2891 <.0001	0.56285 <.0001	1	0.4354 <.0001	0.25573 <.0001	0.45455 <.0001	-0.23515 <.0001
BATTING_HR Homeruns by batters	0.17615 <.0001	-0.00654 0.755	0.4354 <.0001	1	0.51373 <.0001	0.96937 <.0001	-0.58734 <.0001
BATTING_BB Walks by batters	0.23256 <.0001	-0.07246 0.0005	0.25573 <.0001	0.51373 <.0001	1	0.45955 <.0001	-0.65597 <.0001
PITCHING_HR Homeruns allowed	0.18901 <.0001	0.07285 0.0005	0.45455 <.0001	0.96937 <.0001	0.45955 <.0001	1	-0.49314 <.0001
FIELDING_E Errors	-0.17648 <.0001	0.2649 <.0001	-0.23515 <.0001	-0.58734 <.0001	-0.65597 <.0001	-0.49314 <.0001	1

Table 2: Variables with Strongest Pearson Correlation Coefficients to TARGET_WINS

In order to develop a linear regression model that accurately predicts the number of baseball game wins, we created a correlation matrix of all the variables to discover which variables most strongly correlated with the number of wins. (With the exception of TARGET_WINS, the variable names in the table above are all abbreviated by omitting “TEAM_” for the sake of minimizing table size.) **Table 2** shows that TEAM_BATTING_H has the strongest correlation with TARGET_WINS at 0.38877. TEAM_FIELDING_E has the strongest negative correlation with TARGET_WINS at -0.17648. This means that the number of team game wins will likely increase with an increase in base hits by batters but decrease with an increase in the number of fielding errors. The variables shown in the table above are ones we may want to consider including in our model. However, we should be wary of including variables that highly correlate with other predictive variables. For instance, we can see that TEAM_BATTING_HR and TEAM_PITCHING_HR have a 0.96937 correlation. This is logical since the number of homeruns made by batters directly relates to how many homeruns are allowed by the pitchers. **Figure 1** illustrates the strong linear correlation between these two variables. Including both of these variables in a predictive model may result in the model being overfitted to the data it was created on, which means that the model would likely result in poor predictive performance for any new data introduced to the model. While creating our models, we will probably include just one of these two variables to avoid multicollinearity.

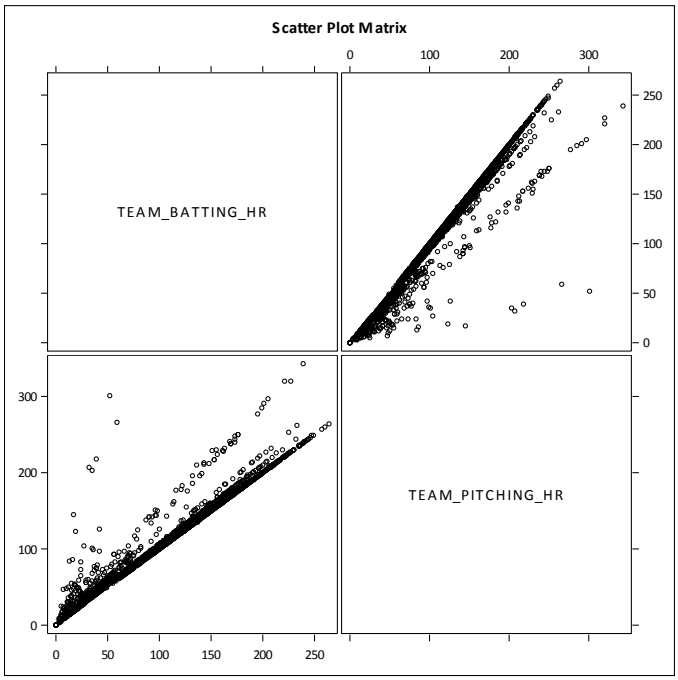

Figure 1: Scatter Plot Matrix of Homerun Variables

In addition to examining the correlation between variables in the Moneyball data set, we also examined the frequencies of the values for each variable to see if there were any unusual patterns. In particular, we were looking for how often 0 appeared in a given variable's data, and where in relation the 0s fell with the other values. Most of the variables had just one or two 0s that were near other low values, such as single digits or low teens. These 0s seemed reasonable. However, a few variables had suspicious 0s. For instance, TEAM_PITCHING_BB had one 0, but the next lowest value was 119. This raised some red flags that we should probably treat this value as an outlier when preparing the data for modeling.

TEAM_BATTING_HR and TEAM_PITCHING_HR each had 15 counts of 0s. While we were wary that they each had the same number of 0s and were both statistics about homeruns, we decided to not investigate these 0s further for several reasons. The statistics for these variables had many low values, so the 0s didn't appear as outliers, and both of these variables had no missing values. Though we are not baseball experts, it seems plausible that a team might not score a single homerun in a season, or that a team might not allow a homerun all season. Consulting an expert would be wise for deciding whether to impute these 0s in future analysis.

Of all the variables, only two variables contained suspicious 0s: TEAM_BATTING_SO and TEAM_PITCHING_SO. As **Tables 3 and 4** depict, each of these variables contained 20 counts of 0s, or approximately 1 percent of the data values were 0. The tables below show only the first three data values of each variable, but we can clearly see that next lowest values for each of these variables are considerably higher than 0. This leads us to believe that the more likely minimum for TEAM_BATTING_SO is in the upper 60s, while the likely minimum for TEAM_PITCHING_SO is closer to 200. Given that each of these variables was also missing 102 values, or approximately 4 percent of their data, we thought it was realistic that these 0s were not accurate. We are not baseball experts, but it seems unlikely that a team would go an entire 162 game season without striking out even once. We will deal with these suspicious 0s later while preparing the data for modeling.

Strikeouts by batters				
TEAM_BATTING_SO	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Frequency Missing = 102				
0	20	0.92	20	0.92
66	1	0.05	21	0.97
67	1	0.05	22	1.01

Table 3: Suspicious Zeros for TEAM_BATTING_SO

Strikeouts by pitchers				
TEAM_PITCHING_SO	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Frequency Missing = 102				
0	20	0.92	20	0.92
181	1	0.05	21	0.97
205	1	0.05	22	1.01

Table 4: Suspicious Zeros for TEAM_PITCHING_SO

After examining the data using correlation matrices and frequency tables, we continued our data exploration by looking at the statistics and graphical distributions of values for each variable. The only variable that exhibited a mostly normal distribution was TEAM_BATTING_2B. **Figure 2** shows how the data fall nearly perfectly straight on the 45-degree line in the Q-Q plot, and the histogram and box plot in **Figure 2** shows that the data fall in a mostly centered distribution with no tails and only a couple potential outliers. Because TEAM_BATTING_2B also had a relatively strong correlation of 0.2891 with TARGET_WINS in the correlation matrix, we thought we should strive to transform the other variables to resemble the normality inherent in TEAM_BATTING_2B.

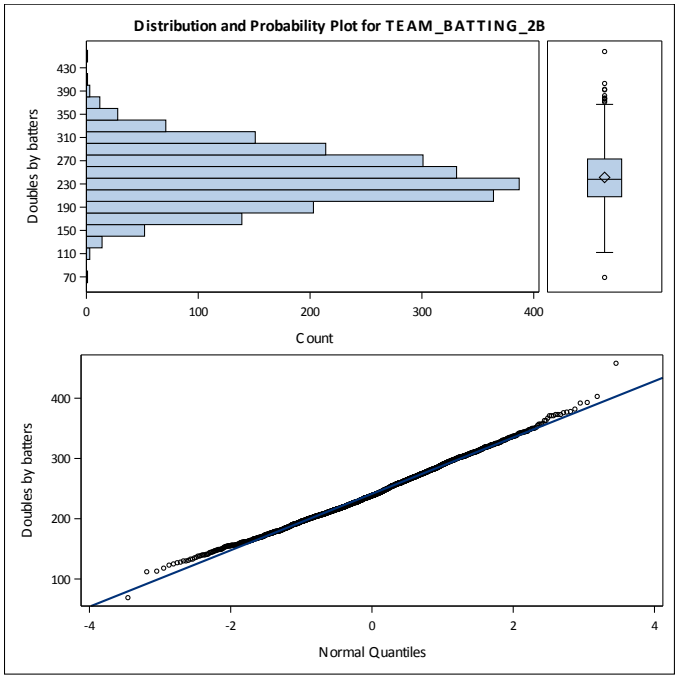

Figure 2: Plots for TEAM_BATTING_2B

We examined the distribution and probability plots for all of the variables and determined that most of the variables would require some sort of transformation, imputation, or truncation in order to achieve even a semblance of a normal distribution. For instance, one of the variables we thought most in need of some work was TEAM_PITCHING_SO. As **Table 1** revealed, TEAM_PITCHING_SO was missing 102 variables, and thus needed some imputed values. **Table 4** also showed that this variable contained several suspicious 0s that might benefit from imputation as well. **Figure 3** shows the distribution of data for TEAM_PITCHING_SO prior to any imputation for missing values or for suspicious 0s.

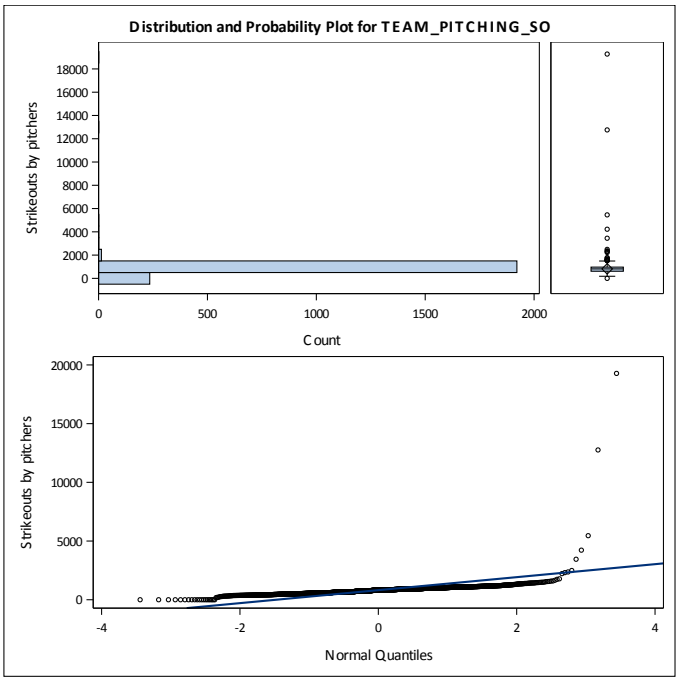

Figure 3: Plots for TEAM_PITCHING_SO

Because TEAM_BATTING_H showed the strongest correlation with TARGET_WINS in the correlation matrix (0.38877), we hypothesized that capping some of the high and low values might improve the predictive potential of this attribute. **Figure 4** shows that the median falls around 1450, so capping the values might make this variable more representative of the typical base hits by batters in a season. Capping might also help correct the slight right-skewedness by reducing the tail containing the super high values.

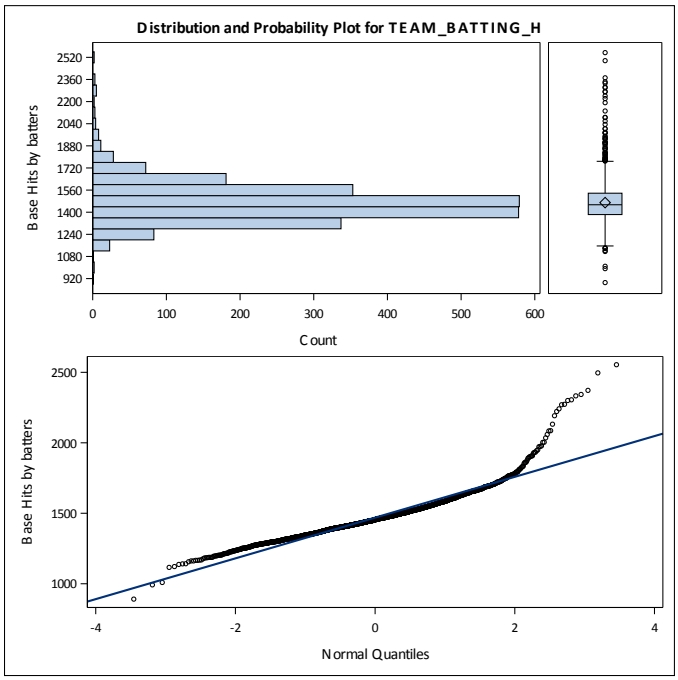

Figure 4: Plots for TEAM_BATTING_H

In the next section we will discuss how we used the observations we made during our data exploration in order to prepare our data for modeling.

Data Preparation

The first step we took in preparing the data was to fix the missing values. As we mentioned in the previous section we decided to drop the variable TEAM_BATTING_HBP because it was missing too many values to prove useful. For the other variables with missing values, we decided that since the means and medians for each variable were similar (i.e., TEAM_PITCHING_SO had mean 817.73 and median 813.5), we could safely use either the mean or median for imputation. We chose to impute the missing values with the median for each since this was generally the more conservative value and could better temper the effects of major outliers that might drive up or down the mean value. The imputations for the missing values were as follows:

TEAM_BATTING_SO = 750
TEAM_BASERUN_SB = 101
TEAM_BASERUN_CS = 49
TEAM_PITCHING_SO = 813.5
TEAM_FIELDING_DP = 149

We also decided to transform the suspicious 0s that we found in TEAM_BATTING_SO and TEAM_PITCHING_SO with the medians for each. We thought that changing the suspicious 0s would make the variables more representative of the data as a whole.

After looking at the distribution and probability plots during our data exploration phase, we made several changes to the variables.

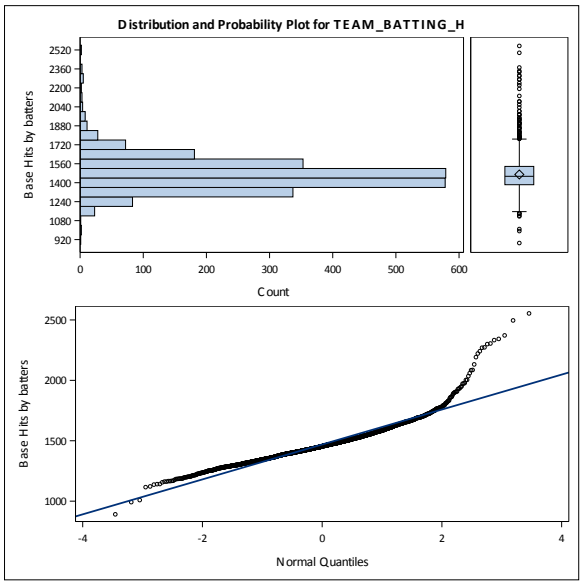

Figure 5: TEAM_BATTING_H

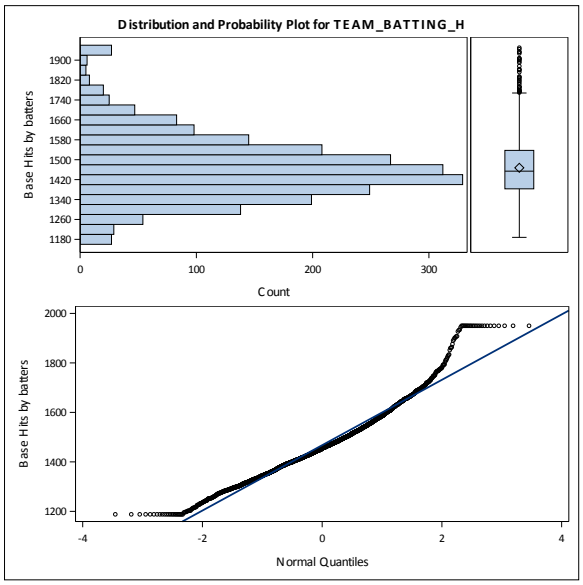

Figure 6: TEAM_BATTING_H, Capped

For TEAM_BATTING_H, we capped the data at the 1st and 99th percentiles. **Figures 5 and 6** show how capping the data affected the distribution. The data is still slightly right-skewed with some very large values, but the histogram more closely follows a normal distribution in **Figure 6** than in **Figure 5**. The boxplot in **Figure 6** shows that there are fewer outliers in the data after truncation as well. In **Tables 5 and 6**, we can see that capping the values has significantly reduced the data range from 1663 to 762, while only slightly changing the mean from approximately 1469 to 1467. This indicates that the data retains many of the same characteristics but without the distraction outliers might have on a predictive model.

Basic Statistical Measures			
Location		**Variability**	
Mean	1469.270	**Std Deviation**	144.59120
Median	1454.000	**Variance**	20907
Mode	1458.000	**Range**	1663
		Interquartile Range	154.50000

Table 5: Statistics for TEAM_BATTING_H

Basic Statistical Measures			
Location		**Variability**	
Mean	1467.433	**Std Deviation**	132.09499
Median	1454.000	**Variance**	17449
Mode	1188.000	**Range**	762.00000
		Interquartile Range	154.50000

Table 6: Statistics for TEAM_BATTING_H, Capped

In addition to capping TEAM_BATTING_H, we fixed the two outliers in TEAM_BATTING_2B by capping the values at the values closest to the outliers. We set the lower cap to 112 to handle the low outlier value 69, and we set the upper cap to 403 to handle the high outlier value 458. The distribution and probability plots did not exhibit drastic changes. For TEAM_BATTING_3B, we fixed the highest outlier, 223, by capping it at the next highest value, 200. In hindsight, we think we should have been more rigorous in our capping, perhaps by setting the low and high caps to the 1st and 99th percentiles.

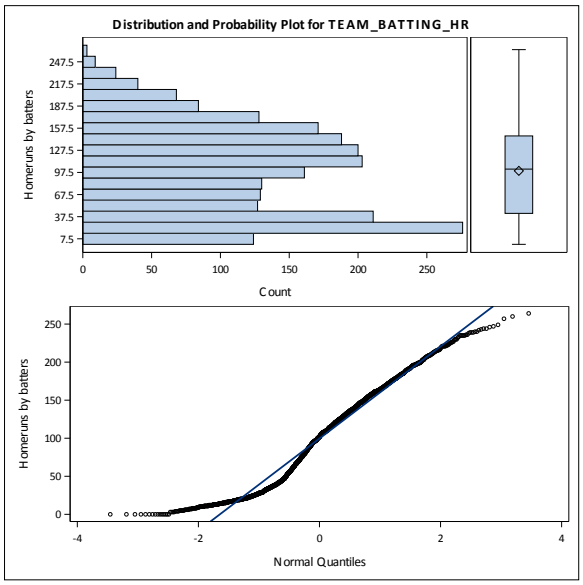

Figure 7: TEAM_BATTING_HR

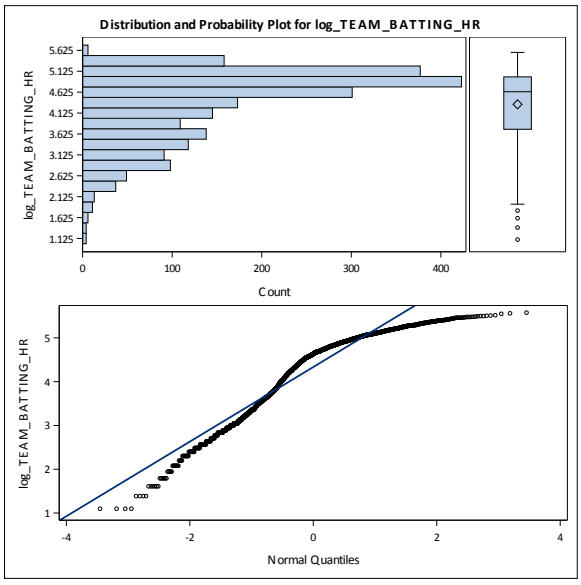

Figure 8: log_TEAM_BATTING_HR

For TEAM_BATTING_HR, we attempted to get a more normal distribution by performing a logarithmic transformation of the data. As **Figures 7 and 8** reveal, the logarithmic transformation did not appear to fix the distribution. Instead, **Figure 8** appears worse off than **Figure 7** because the data is more left-skewed in the histogram, the boxplot shows new low outliers, and the Q-Q plot has fewer data points on the line than in **Figure 7**. These changes all indicate that perhaps a logarithmic transformation of TEAM_BATTING_HR is not appropriate.

In addition to imputing values for the missing values in TEAM_BASERUN_SB, we capped the values at 439, the 99th percentile, because there were many high outliers. **Figure 9** shows a heavily skewed distribution, but **Figure 10** does not show much of an improvement. In the future, we might be more aggressive in capping to the 95th percentile or try binning the data.

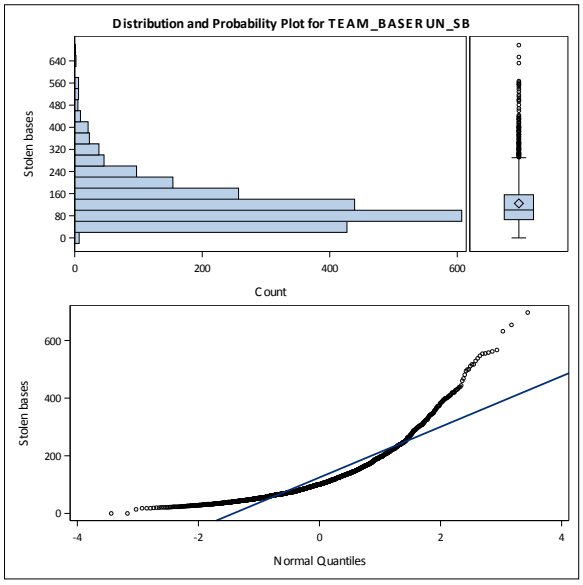

Figure 9: TEAM_BASERUN_SB

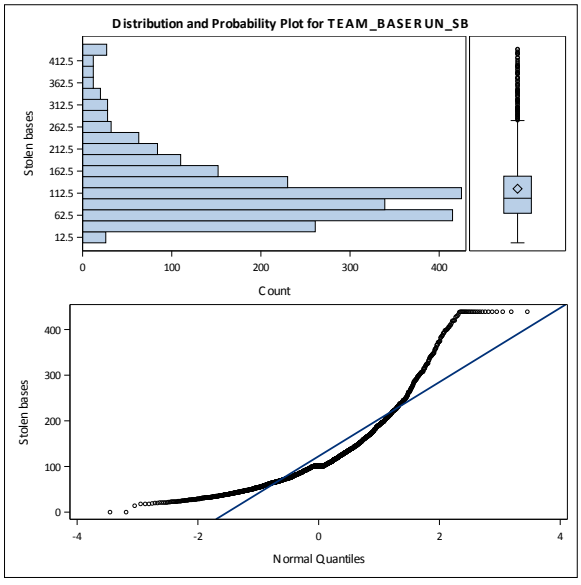

Figure 10: TEAM_BASERUN_SB, Capped

For TEAM_PITCHING_H, we performed a logarithmic transformation of the variable. As **Figures 11 and 12** show, this did not greatly improve the distribution of the data. There are many high values, as demonstrated by the boxplot in **Figure 12**. Capping the values at the 5th and 95th percentiles after performing the logarithmic transformation might normalize the distribution best.

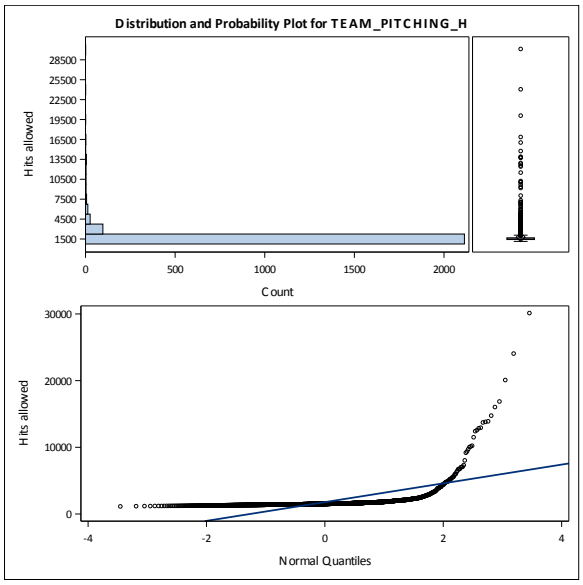

Figure 11: TEAM_PITCHING_H

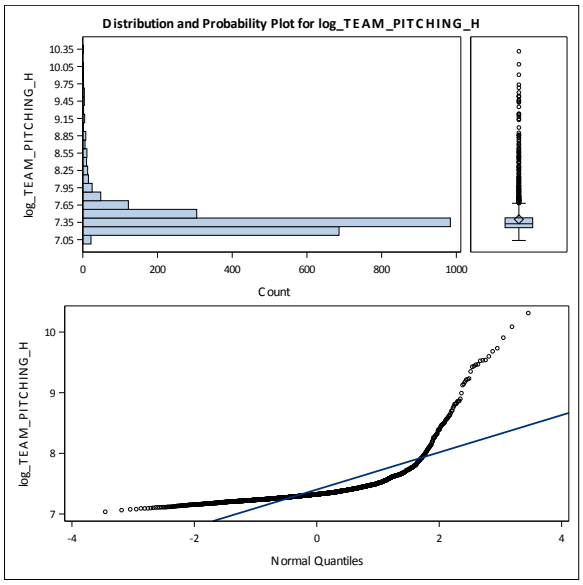

Figure 12: log_TEAM_PITCHING_H

The plots for TEAM_PITCHING_BB in **Figure 13** reveal a right-skewed distribution with many outliers. In order to try to normalize this variable's data, we capped the values at the 1st and 99th percentile. **Figure 14** reveals that capping the data did improve the distribution, but that even more aggressive capping, such as using the 5th and 95th percentiles might further improve the distribution and lead to greater predictive capabilities.

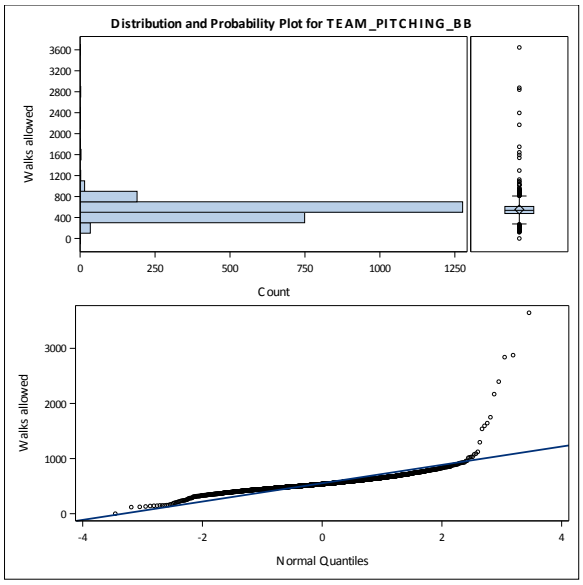

Figure 13: TEAM_PITCHING_BB

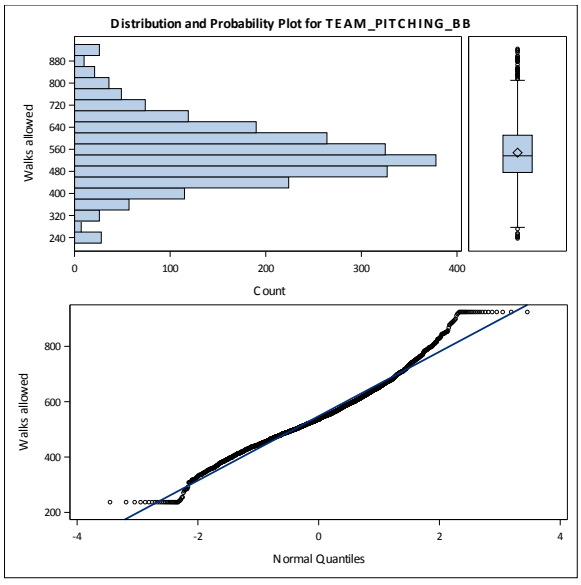

Figure 14: TEAM_PITCHING_BB, Capped

As we mentioned in the previous section, we thought that TEAM_PITCHING_SO required a lot of work before it could be viable for modeling. First we imputed the missing values using the median.

Next we imputed the suspicious 0s with the median as well, and labeled the new variable IMP_TEAM_PITCHING_SO. Then we capped the values at the 1st and 99th percentiles in order to deal with the outliers that could affect the predictive accuracy of the model. As **Figure 16** depicts, these imputations and capping resulted in a more normal distribution than the data in **Figure 15**.

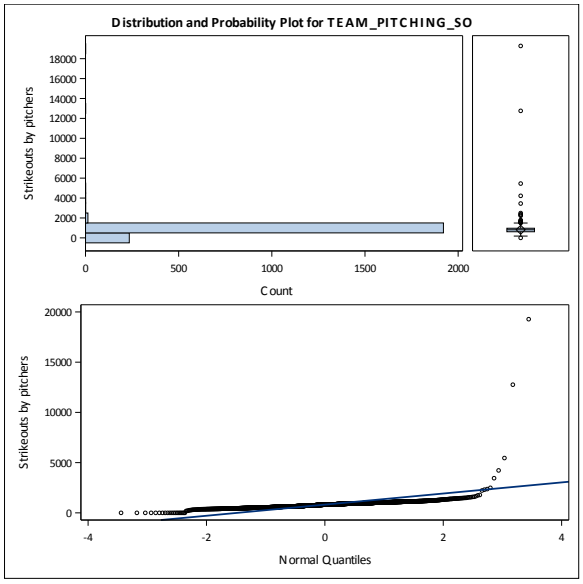

Figure 15: TEAM_PITCHING_SO

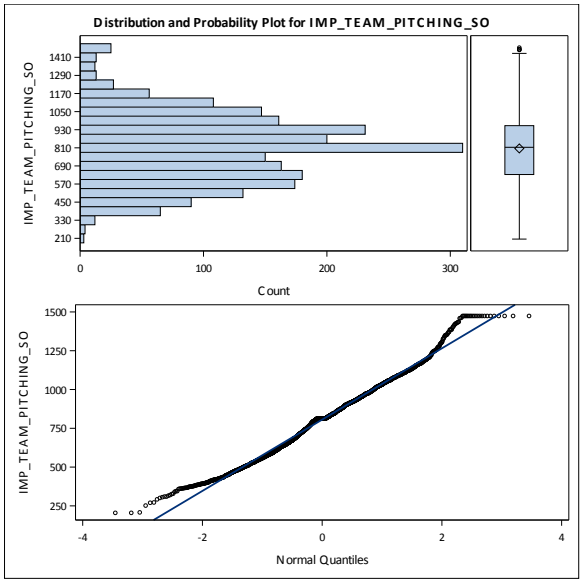

Figure 16: IMP_TEAM_PITCHING_SO

Because TEAM_FIELDING_E had a relatively strong negative correlation, -0.17648, with TARGET_WINS, we wanted to transform this variable in order to normalize it and strengthen its predictive potential. We chose to perform a logarithmic transformation on the variable. As **Figures 17 and 18** show, the transformation helped reduce the number of high outliers as well as help lessen the right-skewedness of the data. We thought that the transformation of this variable would help create a strong model.

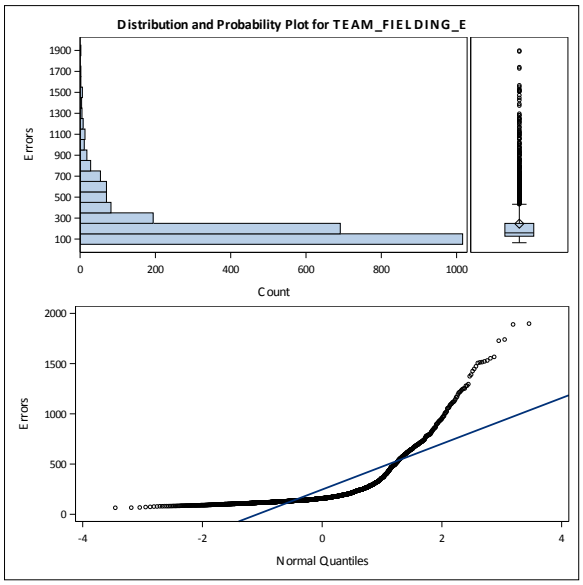

Figure 17: TEAM_FIELDING_E

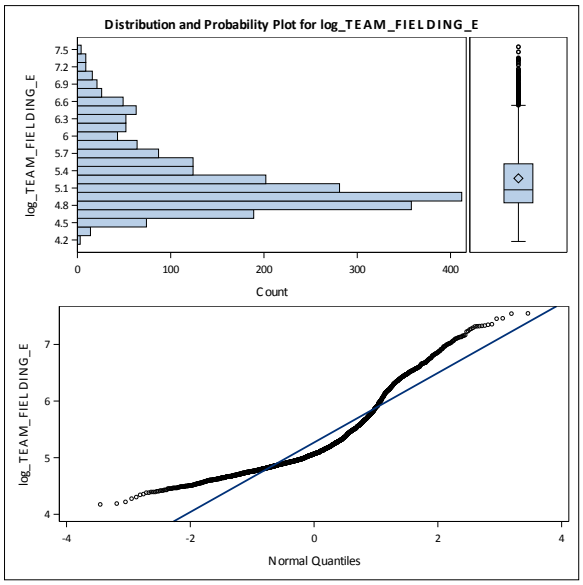

Figure 18: log_TEAM_FIELDING_E

The data changes described above are all in reference to how we prepared the data that resulted in our best model. While our first attempt at preparing the data contained some different transformations, several data preparation steps were the same. We still imputed the missing values, as well as the suspicious 0s, with the medians; we performed a logarithmic transformation of TEAM_BATTING_HR, TEAM_PITCHING_H, and TEAM_FIELDING_E; and we capped TEAM_PITCHING_BB and TEAM_PITCHING_SO with imputed values at the 1st and 99th percentiles. In our first attempt at preparing the data for the model, we performed a logarithmic transformation of TEAM_BATTING_H; we deleted the outliers for TEAM_BATTING_2B and TEAM_BATTING_3B rather than cap them; we capped TEAM_BATTING_BB and TEAM_BASERUN_CS at the 1st and 99th percentiles; and we performed a logarithmic transformation of TEAM_PITCHING_HR, TEAM_BASERUN_SB, and TEAM_BATTING_SO after we had imputed the missing values. The probability and normal plots for the variables after the changes did not look as good, and the first model performed poorly. After preparing the data and running the code for the best model, we tried to improve upon the model during the data preparation stage by imputing using means instead of medians, performing some additional logarithmic transformations, and changing how we capped some variables, as well as adding caps to variables that previously did not have any. Although we tried to improve upon our best model, we were unable to do so. The data preparation we described throughout this section is what we believe leads to the best predictive model.

Model Creation

After exploring the data and preparing them so that they would be ready for modeling, we began selecting variables to include in the models. In total we built 14 models, but we will focus on just three in this report as several of the models used similar variable selection techniques with only slight differences in data preparation or variable inclusion, which thus led to very similar levels of predictive accuracy.

The first model we created included variables that we selected from the correlation matrix. We based our selection on the variables that most strongly correlated with TARGET_WINS prior to any of the data preparation. **Table 2** is reprinted below for ease of reference.

Pearson Correlation Coefficients, N = 2276 Prob > |r| under H0: Rho=0	TARGET_WINS	BATTING_H	BATTING_2B	BATTING_HR	BATTING_BB	PITCHING_HR	FIELDING_E
TARGET_WINS	1	0.38877 <.0001	0.2891 <.0001	0.17615 <.0001	0.23256 <.0001	0.18901 <.0001	-0.17648 <.0001
BATTING_H Base Hits by batters	0.38877 <.0001	1	0.56285 <.0001	-0.00654 0.755	-0.07246 0.0005	0.07285 0.0005	0.2649 <.0001
BATTING_2B Doubles by batters	0.2891 <.0001	0.56285 <.0001	1	0.4354 <.0001	0.25573 <.0001	0.45455 <.0001	-0.23515 <.0001
BATTING_HR Homeruns by batters	0.17615 <.0001	-0.00654 0.755	0.4354 <.0001	1	0.51373 <.0001	0.96937 <.0001	-0.58734 <.0001
BATTING_BB Walks by batters	0.23256 <.0001	-0.07246 0.0005	0.25573 <.0001	0.51373 <.0001	1	0.45955 <.0001	-0.65597 <.0001
PITCHING_HR Homeruns allowed	0.18901 <.0001	0.07285 0.0005	0.45455 <.0001	0.96937 <.0001	0.45955 <.0001	1	-0.49314 <.0001
FIELDING_E Errors	-0.17648 <.0001	0.2649 <.0001	-0.23515 <.0001	-0.58734 <.0001	-0.65597 <.0001	-0.49314 <.0001	1

Table 2, Reprinted: Variables with Strongest Pearson Correlation Coefficients to TARGET_WINS

We deliberately chose to exclude one of the homerun variables because they were so highly correlated that we thought including both would overfit the model to the training data. We chose to include TEAM_PITCHING_HR rather than TEAM_BATTING_HR because it had a slightly higher correlation with TARGET_WINS. For any variables that we had transformed or imputed during the data preparation phase, we used that variable instead of the one in the correlation matrix. We also decided to include the strikeouts by pitchers variable, IMP_TEAM_PITCHING_SO, where we had imputed the median for the missing values as well as for the suspicious 0s. Although TEAM_PITCHING_SO did not show as strong of a correlation with TARGET_WINS as the other variables, we thought that it might be a good predictor since the changes we made to it showed some of the most drastic graphical improvements, as evident in **Figures 15 and 16**. The formula for Model 1 for the number of wins in a season is:

P_TARGET_WINS =	- 605.14800	
	+ 97.64009*log_TEAM_BATTING_H	(Base hits)
	- 0.01845*TEAM_PITCHING_HR	(Homeruns allowed)
	- 0.04802*TEAM_BATTING_2B	(Double base hits)
	+ 0.02377*TEAM_BATTING_BB	(Walks by batters)
	- 5.51309*log_TEAM_FIELDING_E	(Fielding errors)
	+ 0.00626*IMP_TEAM_PITCHING_SO	(Strikeouts by pitchers)

For the most part this formula makes sense. The negative coefficients for TEAM_PITCHING_HR and log_TEAM_FIELDING_E are intuitive since a pitcher giving up homeruns to the other team or the team making mistakes in the field are both more likely to decrease the number of wins. On the other hand, when batters score base hits (log_TEAM_BATTING_H) or get walks (TEAM_BATTING_BB), or when pitchers strike out batters (IMP_TEAM_PITCHING_SO), these actions are more likely to increase the team's wins in a season.

The only coefficient that does not make sense is TEAM_BATTING_2B. Hitting doubles should increase a team's number of wins, not decrease it. The fact that TEAM_BATTING_2B had a 0.2891 correlation to TARGET_WINS—the second highest only to TEAM_BATTING_H—in the correlation matrix makes the negative coefficient even more perplexing. Due to the inexplicable coefficient for TEAM_BATTING_2B, we decided to continue tweaking our data to create more models with more intuitive and interpretable formulas.

After several small modifications to the data and a few more practice models, we ended up with our fourth model. We employed forward, backward, and stepwise selection in order to determine which variables to include in the model. We did not include any logarithmically transformed variables in the variable selection techniques. Also, while we included the variables where we imputed the missing values or capped values to deal with outliers, we did not include the strike out variables where we had imputed the suspicious 0s as medians (IMP_TEAM_BATTING_SO and IMP_TEAM_PITCHING_SO). We decided to leave out logarithmically transformed variables and the strike out variables with imputed 0s from the automated variable selection process because these variables did not strengthen previous models. They were also consistently overlooked by the automated variable selection techniques when we included them in prior models. All three variable selection techniques resulted in the same variables. The formula for Model 4 is:

P_TARGET_WINS = 26.18982

+ 0.04334*TEAM_BATTING_H	(Base hits)
+ 0.07215*TEAM_BATTING_3B	(Triple base hits)
+ 0.01052*TEAM_BATTING_BB	(Walks by batters)
- 0.00875*TEAM_BATTING_SO	(Strikeouts by batters)
+ 0.02676*TEAM_BASERUN_SB	(Stolen bases)
- 0.00072629*TEAM_PITCHING_H	(Hits allowed)
+ 0.06455*TEAM_PITCHING_HR	(Homeruns allowed)
+ 0.00208*TEAM_PITCHING_SO	(Strikeouts by pitchers)
- 0.01895*TEAM_FIELDING_E	(Fielding errors)
- 0.12028*TEAM_FIELDING_DP	(Double plays)

It makes sense that a team that successfully gets base hits, hits triples, gets base walks, steals bases, and pitches strikeouts would have an increase in game wins over a 162 game season. The positive coefficients for these variables indicate a logical relationship. When teams strike out at bat, allow base hits while pitching, and make fielding errors, they are more likely to lose games. Thus, the negative coefficients for these variables make sense. However, there are two variables in the above formula that do not make intuitive sense: TEAM_PITCHING_HR and TEAM_FIELDING_DP. We are not baseball experts, but it seems like allowing home runs or making double plays would have the opposite effect on winning as suggested in the formula above. The Pearson correlation coefficient matrix that we explored earlier in this report did, however, suggest that TEAM_PITCHING_HR has a 0.18901 correlation with TARGET_WINS. **Figure 19** depicts the positive, funnel-like correlation between the two variables. Although this seems counterintuitive, perhaps there is some explanation for this relationship that a baseball expert could provide. If no plausible explanation exists, then we may have to revisit our data or remove the variable from the model.

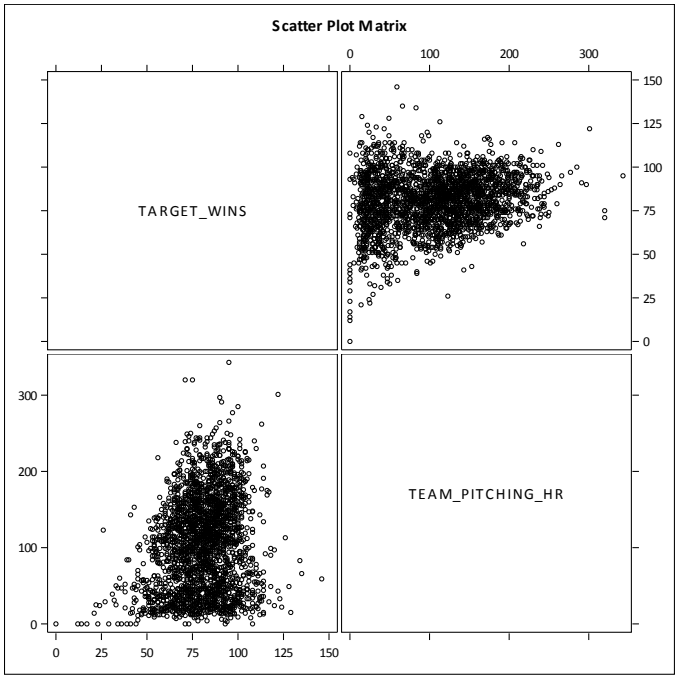

Figure 19: Scatter Plot Matrix between TARGET_WINS and TEAM_PITCHING_HR

For the third model we will discuss in this report, we used a mixture of techniques to select the variables. This model was the sixth model we created, and we were aiming for a very

interpretable model that had all intuitive coefficients. We created a decision tree using R (See Bruckner_DecisionTree.pdf) to see which variables the decision tree software deemed most important. We incorporated all of the variables present in the decision tree, with the exception of TEAM_PITCHING_HR and TEAM_FIELDING_DP, into an automated stepwise variable selection technique. We purposefully eliminated TEAM_PITCHING_HR and TEAM_FIELDING_DP from the automated variable selection process since they tended to have counterintuitive coefficients in previous models. By this same logic we also excluded TEAM_BATTING_2B. We did manually feed in other variables, such as TEAM_BATTING_HR, TEAM_BATTING_SO, TEAM_PITCHING_BB, TEAM_PITCHING_SO, and TEAM_BASERUN_CS, since these variables had generally shown intuitive coefficients in previous models. After running the automated stepwise variable selection technique, we ended up with the following formula for Model 6:

P_TARGET_WINS =	16.31604	
	+ 0.03988*TEAM_BATTING_H	(Base hits)
	+ 0.09003*TEAM_BATTING_3B	(Triple base hits)
	+ 0.05940*TEAM_BATTING_HR	(Homeruns hit)
	- 0.00691*TEAM_BATTING_SO	(Strikeouts by batters)
	+ 0.03855*TEAM_BASERUN_SB	(Stolen bases)
	+ 0.00146*TEAM_PITCHING_SO	(Strikeouts by pitchers)
	- 0.02374*TEAM_FIELDING_E	(Fielding errors)

Of the three models presented in this report, Model 6 has the most intuitive formula. The coefficients are positive for all the events that improve a team's chances of winning a game, while the events that decrease a team's chances of winning, such as when batters strike out or when players in the field make errors, all have negative coefficients. In the next section we will discuss which of the three models explored above we consider the best for predicting wins.

Model Selection

While examining all the models we created, we chose Model 4 (the second model we discussed in the section above), as the best model. The formula for the best model is reprinted below for convenient reference:

P_TARGET_WINS =	26.18982	
	+ 0.04334*TEAM_BATTING_H	(Base hits)
	+ 0.07215*TEAM_BATTING_3B	(Triple base hits)
	+ 0.01052*TEAM_BATTING_BB	(Walks by batters)
	- 0.00875*TEAM_BATTING_SO	(Strikeouts by batters)
	+ 0.02676*TEAM_BASERUN_SB	(Stolen bases)
	- 0.00072629*TEAM_PITCHING_H	(Hits allowed)
	+ 0.06455*TEAM_PITCHING_HR	(Homeruns allowed)
	+ 0.00208*TEAM_PITCHING_SO	(Strikeouts by pitchers)
	- 0.01895*TEAM_FIELDING_E	(Fielding errors)
	- 0.12028*TEAM_FIELDING_DP	(Double plays)

Because the model does contain a couple of counterintuitive coefficients, we would like to consult a baseball expert prior to deploying this model. If the expert determines that this model is flawed, we will re-analyze the data and try to make improvements to change the two coefficients to the correct signs, or we will remove the variables from the model. Our course of action will depend on the timeframe available before the model must be deployed.

Although the coefficients for TEAM_PITCHING_HR and TEAM_FIELDING_DP are counterintuitive, we found that this model performed better than the other two models we discussed in this report for all the criteria we used to determine the best predictive model. We did not use the residual plots from the models to determine the best model since most of the plots looked very similar and thus made for poor comparison between models. The criteria we used for selecting the best model are, in order of importance, the following:

- Mean Squared Error (MSE)
- Variance Inflation Factors (VIFs)
- Adjusted R-Squared

We chose MSE as the most important statistic in selecting a best model because a model with a low error rate is more likely to produce the most accurate predictions. **Table 7** reveals that Model 4 has the lowest MSE. Model 4 also had a considerably lower median squared error than the other two models, though we did not factor this into our criteria for selecting the best model.

Analysis Variable : ERROR		
Model	**Mean**	**Median**
4	173.9630931	64.0000000
6	179.7741652	81.0000000
1	188.4648506	81.0000000

Table 7: Error Statistics for the Three Models

When selecting the best model, we also considered VIFs. We did this because high VIFs indicate the presence of multicollinearity in a model, which can cause the model to perform better with the training data and than with new data. In **Tables 8, 9, and 10**, we can see that none of the models we discussed in this report have very high VIFs. If Model 4 had high VIFs, especially VIFs greater than 10, then we would consider altering the model to reduce the risks of an multicollinearity, or we might choose a different model that appeared more stable.

Parameter Estimates							
Variable	**Label**	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	Intercept	1	26.18982	5.32865	4.91	<.0001	0
TEAM_BATTING_H	Base Hits by batters	1	0.04334	0.00305	14.21	<.0001	2.11204
TEAM_BATTING_3B	Triples by batters	1	0.07215	0.01642	4.40	<.0001	2.72760
TEAM_BATTING_BB	Walks by batters	1	0.01052	0.00338	3.12	0.0019	2.23225
TEAM_BATTING_SO	Strikeouts by batters	1	-0.00875	0.00240	-3.64	0.0003	4.42531
TEAM_BASERUN_SB	Stolen bases	1	0.02676	0.00446	6.00	<.0001	1.70828
TEAM_PITCHING_H	Hits allowed	1	-0.00072629	0.00032590	-2.23	0.0259	2.73690
TEAM_PITCHING_HR	Homeruns allowed	1	0.06455	0.00853	7.57	<.0001	3.55978
TEAM_PITCHING_SO	Strikeouts by pitchers	1	0.00208	0.00066912	3.10	0.0019	1.70317
TEAM_FIELDING_E	Errors	1	-0.01895	0.00239	-7.92	<.0001	3.86632
TEAM_FIELDING_DP	Double Plays	1	-0.12028	0.01309	-9.19	<.0001	1.34244

Table 8: Model 4 Statistics, Including VIFs

Parameter Estimates							
Variable	**Label**	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	Intercept	1	16.31604	4.81880	3.39	0.0007	0
TEAM_BATTING_H	Base Hits by batters	1	0.03988	0.00309	12.89	<.0001	2.10669
TEAM_BATTING_3B	Triples by batters	1	0.09003	0.01668	5.40	<.0001	2.73894
TEAM_BATTING_HR	Homeruns by batters	1	0.05940	0.00928	6.40	<.0001	3.98681
TEAM_BATTING_SO	Strikeouts by batters	1	-0.00691	0.00240	-2.88	0.0041	4.29112
TEAM_BASERUN_SB	Stolen bases	1	0.03855	0.00409	9.43	<.0001	1.53268
TEAM_PITCHING_SO	Strikeouts by pitchers	1	0.00146	0.00060482	2.41	0.0160	1.34835
TEAM_FIELDING_E	Errors	1	-0.02374	0.00171	-13.91	<.0001	1.90717

Table 9: Model 6 Statistics, Including VIFs

Parameter Estimates							
Variable	**Label**	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	Intercept	1	-605.14800	37.22661	-16.26	<.0001	0
log_TEAM_BATTING_H		1	97.64009	5.45810	17.89	<.0001	3.03883
TEAM_PITCHING_HR	Homeruns allowed	1	-0.01845	0.00870	-2.12	0.0340	3.43893
TEAM_BATTING_2B	Doubles by batters	1	-0.04802	0.00968	-4.96	<.0001	2.43624
TEAM_BATTING_BB	Walks by batters	1	0.02377	0.00317	7.49	<.0001	1.76356
log_TEAM_FIELDING_E		1	-5.51309	0.81795	-6.74	<.0001	2.98699
IMP_TEAM_PITCHING_SO		1	0.00626	0.00192	3.26	0.0011	2.34902

Table 10: Model 1 Statistics, Including VIFs

The last factor we used in selecting the best model was the Adjusted R-Squared value. While creating various models, we discovered that a high Adjusted R-Squared value was not always indicative of a good model. For instance, when we ran a stepwise variable selection technique on a set of variables including both TEAM_BATTING_SO and IMP_TEAM_BATTING_SO, both variables appeared in the model. The Adjusted R-Squared value was much higher for this model than any of the values in **Table 11** below, but the VIFs were also very high and suggested a high level of multicollinearity. We chose not to discuss this model in depth in this report since its design was flawed.

Adjusted R-Squared	
Model	**Adj R-Sq**
4	0.2958
6	0.2732
1	0.2374

Table 11: Adjusted R-Squared for the Three Models

Because a higher Adjusted R-Squared did not always indicate a better model, we did not want to select the best model on this metric alone. When models resulted in a relatively high Adjusted R-Squared value but had low VIFs, we interpreted this as a sign that the model might prove to be a

good at predicting baseball wins. We then calculated the MSE for the model in order to determine if the model was better than the previous ones we built. We think that both Adjusted R-Squared and VIFs are helpful metrics for narrowing down which models might be useful, but ultimately we found that MSE was the best indicator of how good a model was at predicting baseball wins.

Conclusion

Our goal with this project was to create a model that best predicted baseball game wins. We created several models based on baseball data from the years 1871 to 2006, inclusive. We prepared the data using various methods, including imputation, truncation, and logarithmic transformations. We then built the models using a variety of techniques, such as stepwise automated variable selection, correlation matrices, and decision trees. Our best model surpassed all the other models in having a low MSE, high Adjusted R-Squared, and low VIFs. Although a couple coefficients in our best model were counterintuitive, the model performed well. We would like to further investigate the counterintuitive coefficients, but doing so requires consulting a baseball expert, which is beyond the scope of the project. In the meantime, we are confident that the model we created is suitable for predicting baseball wins.