**Moneyball OLS Regression Project**

**Bingo Bonus**

PROC REG vs.PROC GLM & PROC GENMOD is in Appendix A (page 12) – 20 Points
Recreate This assignment in R is in Appendix B (page 16) – 20 Points
Scored file was turned in as a SAS Data Set – 10 Points

## Introduction

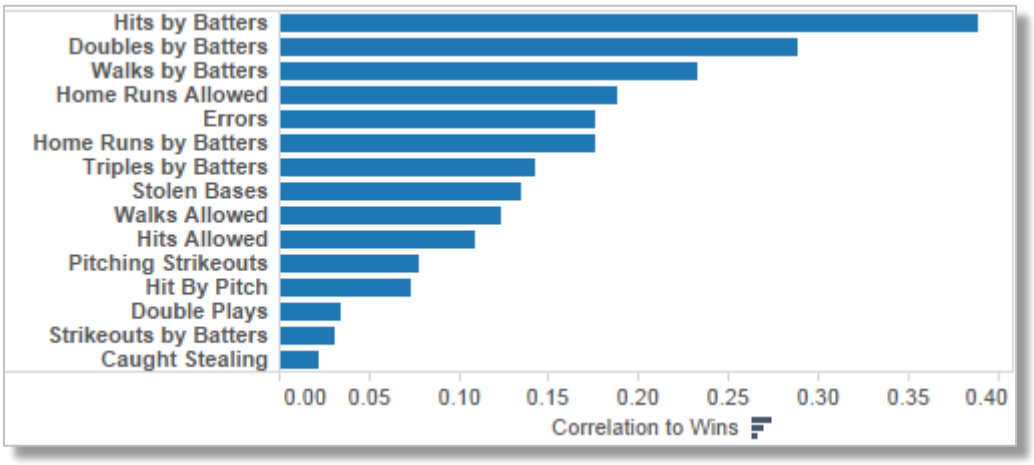
The purpose of this project is to develop a model that will predict the number of wins a professional baseball team will achieve in a season. This will be accomplished by analyzing data from baseball teams between the years of 1871 and 2006. This data will be used to generate a series of a multiple regression models. Techniques such as Forward, Backward, and Stepwise Regression will be used during this process. Evaluation measures including adjusted $r^2$, AIC, BIC, Mallow's Cp will be used to identify the best model and determine its adequacy.

## Data

The data used in the project consists for 2276 records. Each record represents a statistical year for a professional baseball team. In addition to the total number of wins the team had in a given year, each record contains various offensive and defensive statistics including number of hits, homeruns, stolen bases, hits allowed, strikeouts and errors.

Several of the individual variables show a strong correlation to Target Wins. The chart below shows the absolute value of the correlation for each variable. As the chart shows, the fields with the strongest correlation to wins are Hits by Batters, Doubles by Batters, and Walks by Batters. Conversely, the variables that had the weakest correlation to wins are double plays, strikeouts and the number of times a player was caught stealing a base.

**Absolute Correlation to Wins**



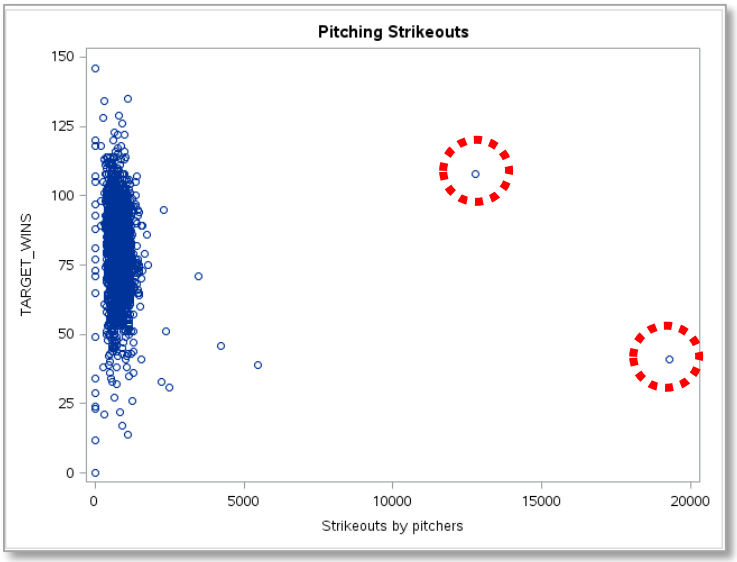

It is important to note that the dataset used in this project has some flaws. Some of the records have missing value for one or more variables. The following table shows these variables. The third column shows how many records have values. The fourth column shows the number of records with missing values.

**Variables With Missing Values**

The MEANS Procedure

| Variable | Label | N | N Miss |
|---|---|---|---|
| TEAM_BATTING_SO | Strikeouts by batters | 2174 | 102 |
| TEAM_BASERUN_SB | Stolen bases | 2145 | 131 |
| TEAM_BASERUN_CS | Caught stealing | 1504 | 772 |
| TEAM_BATTING_HBP | Batters hit by pitch | 191 | 2085 |
| TEAM_PITCHING_SO | Strikeouts by pitchers | 2174 | 102 |
| TEAM_FIELDING_DP | Double Plays | 1990 | 286 |

In addition to the challenge of missing data, several of the variables contain outliers that have the potential to distort predictive models. In some cases these outliers are extreme, such as the picture below. This shows that there are two values, circled in red, for the Strikeouts by Pitchers variable that are drastically higher than all of the others. The highest value is more than 23 times larger than the overall average. It is possible that these values were input incorrectly when the dataset was created.



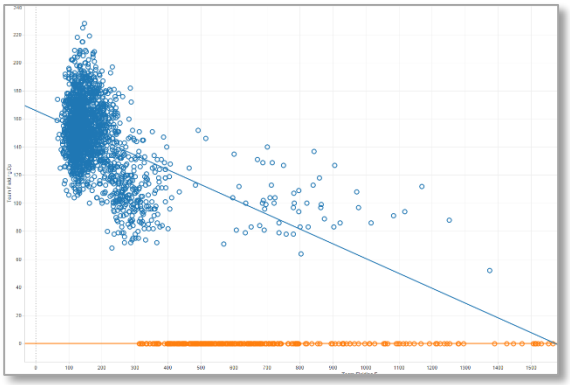

## Data Prep

In order to prepare the data for modeling I performed several steps including imputing missing values, transforming data to eliminate outliers, and creating several derived variables. Records with missing values are skipped during the linear regression modeling process so it is important to fix them. For each variable with missing values I attempted numerous ways of fixing them. Methods attempted included deleting records with missing values, avoiding using variables with a large number of missing values, using business rules, and replacing the missing value with a variable's mean, median, or mode.

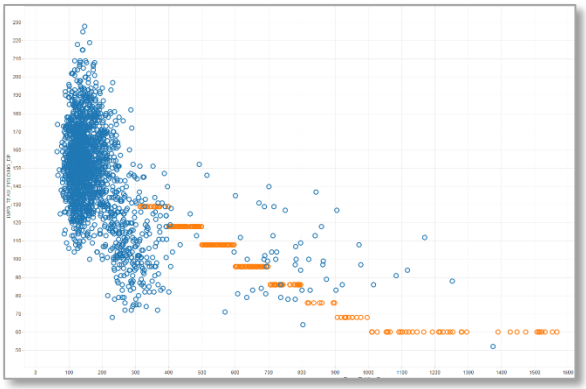
I tried various methods on each variable to see which would increase overall accuracy the most. In the end, I replaced the missing values of Strikeouts by Batters, Stolen Bases, Caught Stealing, and Strikeouts by Pitchers with the mean value of their respective variables. I deleted one record from the dataset because it was either missing or had a value of 0 for the majority of its variables. I avoided using the Batters Hit by Pitch variable all together because over 90% of the records were missing this value. Finally, I noticed that the Double Plays variable has a strong correlation to Errors. The data showed that

teams who make fewer errors record more double plays. This makes sense intuitively because one would expect higher skilled players to make fewer errors and turn more double plays. For this variable, instead of replacing the missing value with the mean, I used a business rule based on a team's fielding errors to impute the missing values. The images below show a scatterplots of Errors and Double Plays before and after the transformation. The missing Double Play values are in orange.

**Double Plays vs. Errors Before Transformation**



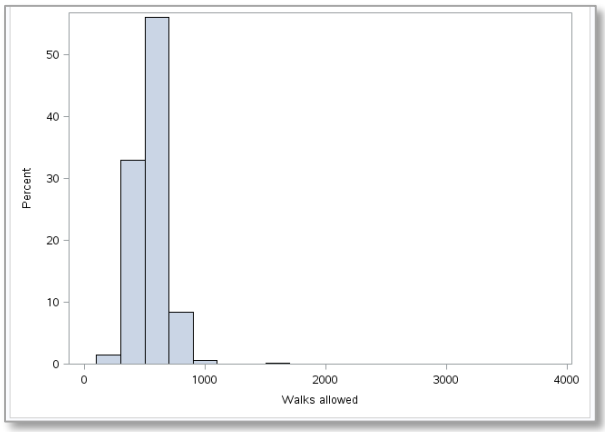
**Double Plays vs Errors After Transformation**



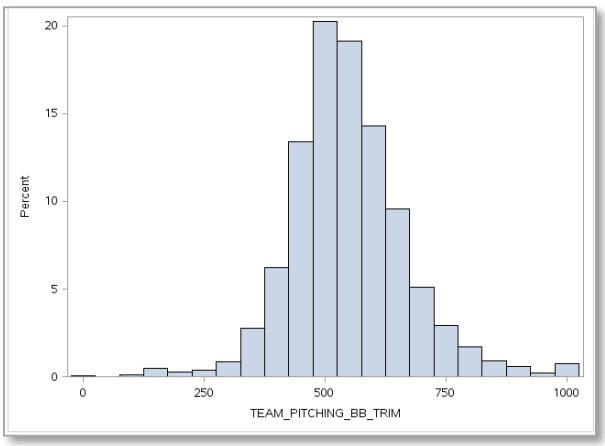
Once the missing values were imputed the next step was to eliminate outliers. Outliers have the potential to significantly influence model accuracy in a negative way. As with the missing values, I attempted a variety of methods to eliminate outliers on the necessary variables and selected the one that had the most positive effect on model accuracy. Methods used include trimming, binning, and using the logaritm function.

One method used to eliminate outliers on several variables is trimming. Trimming takes all of the values higher than a specified threshold and reassigns their value as the threshold number. This helps to normalize the data yet keeps the outliers at the top of the scale. This transformation method was used on Pitching Strikouts and Walks Allowed. The images below show the distribution of the Walks Allowed variable before and after it was trimmed.

**Walks Allowed Before Trim**

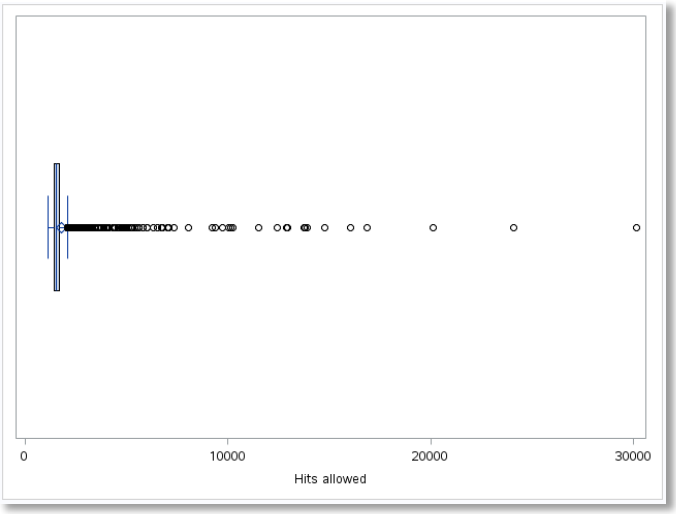



**Walks Allowed After Trim**

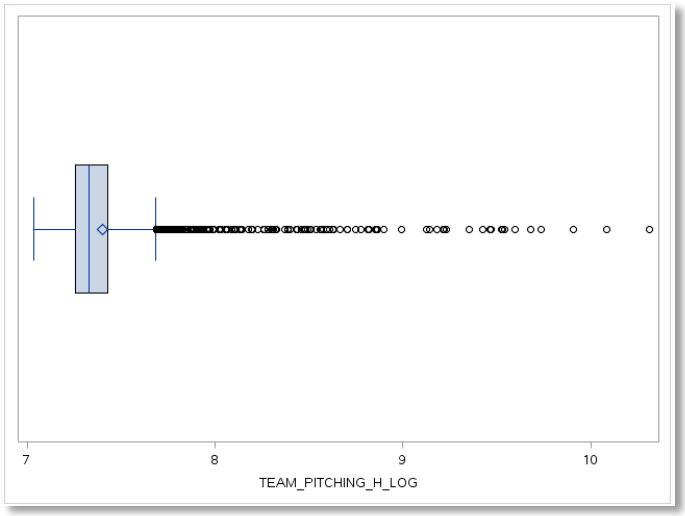


The logarithm function takes the logarithmic value of a variable. This helps to reduce the distance between values. After trying this method on all fields containing outliers I determined that it is the most useful technique for the Hits Allowed variable. The images below show the distribution of Hits Allowed before and after the logarithm function was applied. The images show that even though the data after the transformation still displays a positive skew, the data points are closer together and the scale has been drastically reduced. The scale in the "Before" chart goes up to 30,000. The scale in the "After" tops out around 10.

**Hits Allowed Before Log Transformation**




**Hits Allowed After Log Transformation**




In order to achieve better predictability, I created several new variables derived from the original dataset. I created a flag field for each variable that I imputed data for, including Strikeouts by Batters, Stolen Bases, Caught Stealing, Strikeouts by Pitchers, and Double Plays. The value of the flag field is "0" when the corresponding variable contains an original value and "1" when its value was imputed. This is done because in some cases, the fact that a value is missing can be predictive.

Finally, I wanted to create variables that better capture a team's overall offensive and defensive performance. The image on the first page shows that the variables related to hitting had the highest correlation to wins. With this in mind, I created two extra variables concerned with hitting, and a third to capture a cumulative pitching performance. The new variables are listed below.

- Total Bases Touched: Hits + (Doubles * 2) + (Triples * 3) + (Homeruns * 4) + Walks + Stolen Bases
- Extra Base Hits: Doubles + Triples + Homeruns
- Total Bases Given Up: Hits Allowed + Walks Allowed + (Homeruns * 4)

**Build Models: Model v1**

The first model is comprised of only the original variables in the dataset. It does not include any of the derived fields or flag fields for imputed values. The variables in the model were chosen with the

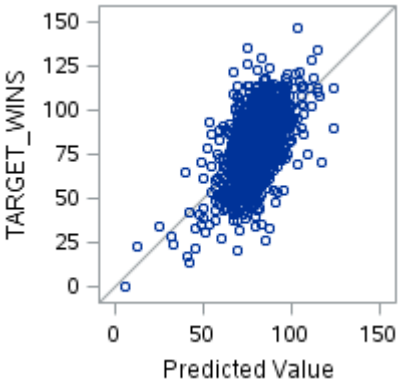
forward selection method. The only variables out of the original set that were not selected for the model were Caught Stealing and Batters Hit by Pitch, which was excluded from all models due to its high number of missing values.

According to the model v1, the variables that have the strongest positive impact on target wins are Homeruns by Batters, Triples by Batters, and Walks by Batters. This seems to agree with conventional wisdom that more hits and base runners lead to more wins.

The variables that have the strongest negative impact on wins are Double Plays, Homeruns Allowed, and Fielding Errors. Two of those three are understandable. Allowing homeruns and committing fielding errors are widely regarded as bad. It is easy to believe that these two things would lead to fewer wins. The negative coefficient of the Double Plays variables suggests that achieving a double play would hurt a team's chances at winning. This is counter-intuitive and will need to be investigated further.

In addition to the Double Plays variable, there are two other variables with coefficients that are counter-intuitive. According to this model, Doubles by Batters lead to fewer wins. Also, since the Hits Allowed coefficient is positive, the model also implies that allowing more hits leads to more wins. Each of these are opposite of what is expected. While this may seem wrong at first, there may be a reasonable explanation. This idea will be discussed in the Model Selection section of this paper.

| Variable | Parameter Estimate |
|---|---|
| Intercept | -104.46878 |
| TEAM_BATTING_H | 0.03893 |
| TEAM_BATTING_2B | -0.03056 |
| TEAM_BATTING_3B | 0.09502 |
| TEAM_BATTING_HR | 0.15094 |
| TEAM_BATTING_BB | 0.05111 |
| IMP_TEAM_BATTING_SO | -0.01447 |
| IMP_TEAM_BASERUN_SB | 0.02926 |
| IMP_TEAM_BASERUN_CS | -0.01085 |
| TEAM_PITCHING_H_LOG | 20.00236 |
| IMP_TEAM_PITCHING_HR | -0.09569 |
| TEAM_PITCHING_BB_TRIM | -0.03639 |
| TEAM_PITCHING_SO_TRIM | 0.01200 |
| TEAM_FIELDING_E | -0.04439 |
| IMP3_TEAM_FIELDING_DP | -0.13995 |




| | |
|---|---|
| Observations | 2275 |
| Parameters | 15 |
| Error DF | 2260 |
| MSE | 166.28 |
| R-Square | 0.3287 |
| Adj R-Square | 0.3245 |

The scatterplot above on the right shows the comparison of actual wins to predicted wins. If the model were able to flawlessly predict a team's wins, the data points would follow the gray diagonal regression line perfectly. As the image shows, this is not the case for model v1. The chart displays quite a bit of variation from the regression line. The following models will attempt to improve on this

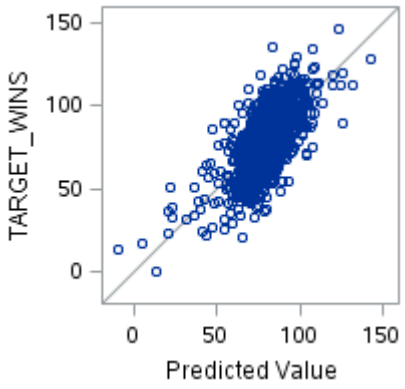
**Model v2**

The second model includes the flag variables that show whether or not the value was imputed as well as the derived variables Total Bases Touched and Total Bases Given. To choose the variables in this model I allowed SAS to build the best model for each possible quantity of variables. Based on the adjusted $r^2$ value, the model with 16 variables scored the highest and was chosen as model v2. The variables excluded by this model include Hits by Batters, Triples by Batters, Strikeouts by Batters, Homeruns Allowed, Extra Base Hits, and the flag variables for imputed Strikeouts by Batters.

Model v2 shows that the variables with the strongest positive impact on wins are the flag variables which show if a value has been imputed. Teams that had missing values for Stolen Bases have their prediction boosted by almost 40 wins. Teams that were missing values for Pitching Strikeouts have their win total boosted by 7.59. As for the non-flag variables, Total Bases Touched, Stolen Bases, and Walks by Batters had the strongest positive impact on wins. The log value of Hits Allowed had the strongest negative impact on wins.

Out of the 16 variables in model v2, 5 of them had coefficients that effect wins differently than conventional wisdom would dictate. According to the model, Doubles by Batters, Homeruns by Batters, Double Plays and Pitching Strikeouts lead to fewer wins. The positive coefficient of Total Bases Given means that a team is more likely to win if they allow the other team to get on base. Most baseball experts would disagree with this notion. This idea will be discussed in later in this paper.

| Variable | Parameter Estimate |
|---|---|
| Intercept | 93.67577 |
| TEAM_BATTING_2B | -0.08857 |
| TEAM_BATTING_HR | -0.10669 |
| TEAM_BATTING_BB | 0.02175 |
| IMP_TEAM_BASERUN_SB | 0.02794 |
| IMP_TEAM_BASERUN_CS | -0.02446 |
| TEAM_PITCHING_H_LOG | -8.62949 |
| TEAM_PITCHING_BB_TRIM | -0.03581 |
| TEAM_PITCHING_SO_TRIM | -0.01342 |
| TEAM_FIELDING_E | -0.07625 |
| IMP3_TEAM_FIELDING_DP | -0.11826 |
| TOTAL_BASES_TOUCHED | 0.03150 |
| TOTAL_BASES_GIVEN | 0.01240 |
| m_TEAM_FIELDING_DP | 2.47435 |
| m_TEAM_BASERUN_SB | 39.83820 |
| m_TEAM_PITCHING_SO | 7.59126 |
| m_TEAM_BASERUN_CS | 1.20322 |




| | |
|---|---|
| Observations | 2275 |
| Parameters | 17 |
| Error DF | 2258 |
| MSE | 138.28 |
| R-Square | 0.4422 |
| Adj R-Square | 0.4382 |

The scatter plot and model fit statistics, shown on the above right, show that the predictions made by model v2 are significantly improved over model v1. Specifically, this is displayed by the higher R-Square and Adjusted R-Square numbers and the lower Mean Square Error (MSE) value.

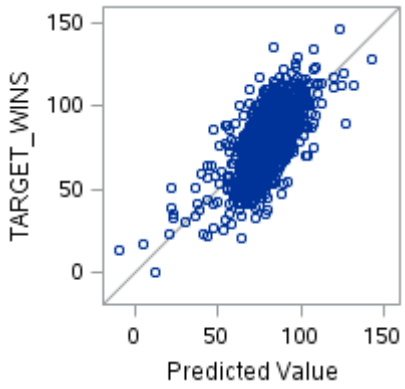
**Model v3**

The third model includes all flag variables and derived variables. The variables in the model were chosen using the backwards selection method. This method chose 16 of the 22 potential variables to be in the model. The fields excluded were Batting Strikeouts, Homeruns Allowed, Total Bases Touched, and Extra Base Hits. The flag variables that mark imputed values for Pitching Strikeouts and Caught Stealing were also excluded.

Similar to model v2, the variables that had the most effect on wins were the flag variables that represented imputed values for Stolen Bases and Pitching Strikeouts. The coefficients for these variables were slightly higher compared to model v2 meaning teams with missing values for these variables win more games. Out of the non-flag variables, model v3 shows that Triples by Batters, Stolen Bases, and Walks by Batters have the largest positive effect on wins. Hits Allowed, Double Plays, and Errors have the strongest negative effect on wins.

This model has 4 coefficients that raise concern. According to model v3, Doubles Hit, Double Plays, and Strikeouts by Pitchers lead to fewer wins, while Total Bases Given leads to more wins. The presence of counter-intuitive coefficients has occurred in each of the first three models and will be discussed in the Model Selection section of this paper.

| Variable | Parameter Estimate |
|---|---|
| Intercept | 91.84478 |
| TEAM_BATTING_H | 0.03265 |
| TEAM_BATTING_2B | -0.02673 |
| TEAM_BATTING_3B | 0.09201 |
| TEAM_BATTING_HR | 0.01751 |
| TEAM_BATTING_BB | 0.05098 |
| IMP_TEAM_BASERUN_SB | 0.06068 |
| IMP_TEAM_BASERUN_CS | -0.03090 |
| TEAM_PITCHING_H_LOG | -8.23266 |
| TEAM_PITCHING_BB_TRIM | -0.03332 |
| TEAM_PITCHING_SO_TRIM | -0.01368 |
| TEAM_FIELDING_E | -0.07595 |
| IMP3_TEAM_FIELDING_DP | -0.12137 |
| TOTAL_BASES_GIVEN | 0.01192 |
| m_TEAM_BATTING_SO | 7.88521 |
| m_TEAM_FIELDING_DP | 2.48453 |
| m_TEAM_BASERUN_SB | 40.15057 |




| | |
|---|---|
| Observations | 2275 |
| Parameters | 17 |
| Error DF | 2258 |
| MSE | 138.4 |
| R-Square | 0.4417 |
| Adj R-Square | 0.4378 |

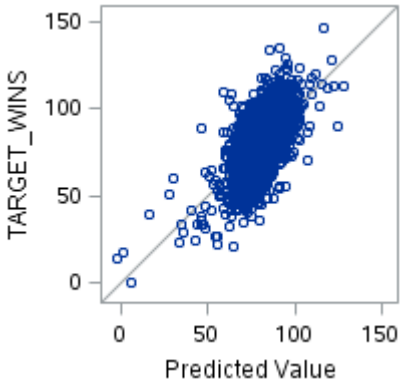
The model fit statistics for model v3, shown on the above right, indicate that the accuarcy of this model is very close to that of model v2. The lower Adjusted R-Square value indicates that this model is slightly less predictive than model v2, but still a significant improvement over model v1.

**Model v4**

In each of the first three models there are multiple coefficients that are the opposite of what one would most likely expect. I built a fourth model that only includes variables with intuitive coefficients. Using the backward elimination method, I started with all of the possible variables in models v2 and v3 and gradually eliminated them until all of the signs on the coefficients matched what one would expect. This process caused me to manually remove Doubles Hit by Batters, Hits Allowed, Homeruns Allowed, Pitching Strikeouts, Double Plays, and all of the derived fields that I created. In the end, this model ended up with only 9 variables.

Similar to models v2 and v3, the flag variables had the strongest effect on wins. Out of the non-flag variables, Triples Hit by Batters, Homeruns by Batters, and Stolen Bases had the strongest positive effect on wins. Fielding Errors, base runners caught stealing, and Strikeouts by Batters had the largest negative effect on wins.

| Variable | Parameter Estimate |
|---|---|
| Intercept | 10.13693 |
| TEAM_BATTING_H | 0.04141 |
| TEAM_BATTING_3B | 0.06809 |
| TEAM_BATTING_HR | 0.06852 |
| TEAM_BATTING_BB | 0.01873 |
| IMP_TEAM_BATTING_SO | -0.01196 |
| IMP_TEAM_BASERUN_SB | 0.05641 |
| TEAM_FIELDING_E | -0.04370 |
| m_TEAM_BATTING_SO | 11.95759 |
| m_TEAM_BASERUN_SB | 31.10365 |




| | |
|---|---|
| Observations | 2275 |
| Parameters | 10 |
| Error DF | 2265 |
| MSE | 151.67 |
| R-Square | 0.3863 |
| Adj R-Square | 0.3838 |

The model fit statistics are shown above on the right. The Adjusted R-Square value is lower for this model compared to the previous two. While this is only one measure and a more detailed evaluation will be performed later in this paper, it indicates that model v4 is less predictive than models v2 and v3. This shows that removing the counter-intuitive coefficients decreased the model's accuracy. However, even though it is less predictive, the fact that it has fewer variables and the coefficients are in line with expectations, may make this model easier to understand and implement.

**Trimming**

After building the models I trimmed the predicted values that each one produced. The upper bound for all of the models is 113. The lower threshold varied between models, but is always in the range of 15-30. By doing this, all predicted win totals higher than 113 got set to 113. Similarly, all

predicted win totals that were below the lower threshold got set to the value of the lower threshold. As a result, the adjusted $r^2$ value for each model increased. The results are in the following table.

| Model | Adjusted $r^2$ Before | Adjusted $r^2$ After |
|---|---|---|
| v1 | 0.3213 | 0.3257 |
| v2 | 0.4382 | 0.4445 |
| v3 | 0.4378 | 0.4440 |
| v4 | 0.3838 | 0.3881 |

**Model Selection**

Each of the first three models is comprised of several coefficients that are counter-intuitive. Although it may be tempting to disbelieve these models, further analysis may provide some enlightenment. For example, each of the first three models imply that double plays lead to fewer wins. This may seem counter-intuitive. However, in order to have a double play, there must be base runners in the field. Perhaps this is an indication that teams who have a lot of double plays also frequently give up more hits, which in turn leads to fewer wins.

The same could be said for Pitching Strikeouts. According to models v2 and v3, pitching strikeouts lead to fewer wins. Most would tend believe the opposite. However, it could be true that pitchers who throw a lot of strikes and record a lot of strikeouts, also give up a lot of hits because they are throwing so many hittable pitches. Further investigation will be needed to prove these hypotheses, however, that is beyond the scope of this document.

In order to select the best model I used criteria including adjusted $r^2$, AIC, BIC, Mallow's Cp, and a bit of judgment. Each of these measures the accuracy of the model while penalizing it for complexity. The adjusted $r^2$measure is the ratio of the regression mean squares to the total mean squares (Hosmer, Lemeshow, & Sturdivant, 2013). It differs from $r^2$ in that it provides a correction for the number of variables in the model. As the table below show, model v2 has the highest value for adjusted $r^2$ out of all four models. This indicates that it has the strongest correlation between predicted wins and actual wins.

| Model | Adjusted-$r^2$ | AIC | BIC | Cp |
|---|---|---|---|---|
| v1 | 0.3301 | 11631.4509 | 11632.6922 | 15 |
| v2 | 0.4445 | 11190.5507 | 11192.5591 | 14.6482 |
| v3 | 0.444 | 11192.57 | 11194.5478 | 16.4992 |
| v4 | 0.3881 | 11410.6078 | 11412.2303 | 7.9973 |

The Akaike Information Criterion (AIC) is a measure that includes the log-likelihood of the fitted model and the number of regression coefficients. Lower AIC values are preferred over larger ones (Hosmer, et al, 2013). Model v2 has the lowest AIC score which indicates that it is preferred over the other models.

The Bayesian Information Criterion (BIC) is a measure that is similar to AIC. BIC adjusts for the number of fitted parameters with a penalty that increases with the sample size (Gelman, Hwang, & Vehtari, 2013). With BIC, lower values are better. Once again, model v2 has the best score for this measure.

Mallows' Cp helps identify if the correct number of variables are in the model. Mallows suggested that you should choose the first model in which Cp is less than or equal to the total number of parameters (Cody, 2011). Model v2 fits this criteria.

All four models have their strong points. Model v1 uses only the original variables and does not base its predictions off of imputed-value flags. This makes the model more understandable. Model v2 scored the highest on all four evaluation measures, however it also had the highest number of counter-intuitive coefficients. Model v3 scored slightly lower on all of the adequacy measures compared to model v2, but has fewer counter-intuitive coefficients. Model v4 contains only intuitive coefficients, but was not as predictive as models v2 and v3.

Ultimately, I recommend model v3. This model scored only marginally lower than the highest scoring model in all of the evaluation measures and contains fewer counter-intuitive coefficients. Out of all the models, this I believe model v3 represents the best balance between understandability and predictability.

## Conclusion

A number of different models were built to predict the number of games that a professional baseball team won in a season between the years 1871 to 2006. The recommended model, model v3, was derived using the backward selection technique. Of all the models built, this one offers the strongest balance of understandability and predictability. The downside to this model that it contains several coefficients that indicate that certain plays, which are generally regarded as positive, will lead to fewer wins. Further analysis will need to be performed to confirm these notions. Perhaps on the contrary, this investigation may identify underlying causes that lead to the counter-intuitive coefficients, such as teams with more double plays allow more base runners. Allowing more base runners may lead to more runs for the opposing team, and in turn fewer wins. That investigation, however, is beyond the scope of this document.

**References**

Cody, Ron. 2011. SAS Statistics by Example. Cary, NC: SAS Institute Inc.

Gelman, A., Hwang, J., & Vehtari, A. 2013. Understanding Predictive Information Criteria for Bayesian Models. Columbia University

Hosmer, Jr D. Lemeshow S, Sturdivant, R. 2013. Applied Logistic Regression. New Jersey: John Wiley & Sons Inc.

**Appendix A: Bingo Bonus – Part 1**
**PROC REG vs PROC GLM and PROC GENMOD**

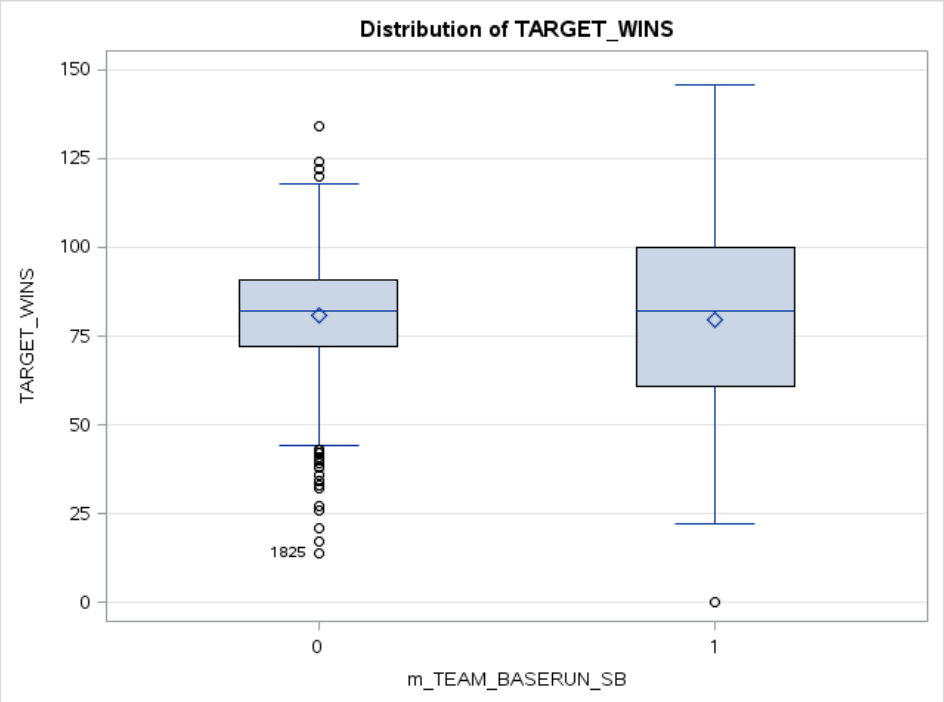
******I believe I should receive 20 Bingo Bonus Points for this. *******

I ran each of my four models through all 3 procedures (PROC REG, PROC GLM, and PROC GENMOD). The resulting coefficients (shown on the proceeding pages) were virtually identical. The only differences in the coefficients are probably attributed to rounding. The output from PROC REG included a great deal of fit diagnostics that the others did not. It has charts for the residuals for all of the variables as well as measures like the QQ-Plot and Cook's D. I found those things very helpful for this project.

I did a little more reading on PROC GLM and found that it can be very beneficial when looking at how a target variable is distributed across one or more predictor variables. For example, at one point while I was building the models for this paper I was curious about what TARGET_WINS looked like for the variables with missing values. After reading about PROC GLM I ran the following code:

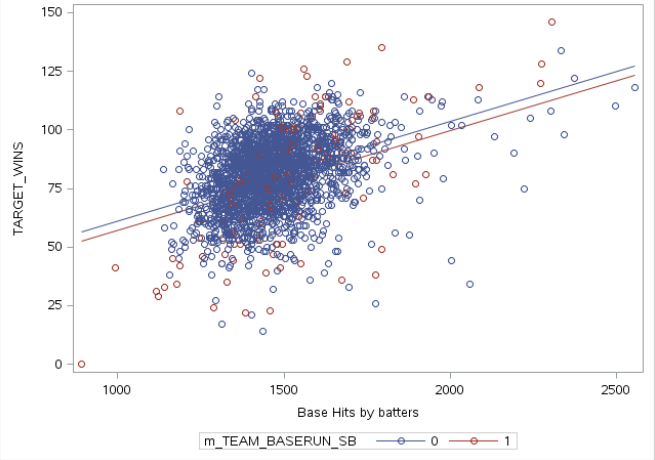
```
PROC GLM data=moneyball_temp_v100;
        class m_TEAM_BASERUN_SB;
        model target_wins = TEAM_BATTING_H m_TEAM_BASERUN_SB / ss3;
        means m_TEAM_BASERUN_SB / hovtest;
run;
```

In the output I could see what the mean, standard deviation, and distribution of Target Wins was for each value of my flag variable. I could see this being useful in future projects.




| Level of m_TEAM_BASERUN_SB | N | TARGET_WINS | | TEAM_BATTING_H | |
|---|---|---|---|---|---|
| | | Mean | Std Dev | Mean | Std Dev |
| 0 | 2143 | 80.9034064 | 14.6364440 | 1466.14979 | 136.880101 |
| 1 | 132 | 79.4848485 | 27.7173534 | 1523.40909 | 228.002911 |

I also really like the scatterplot that was included in the output. The scatterplot below showed me how TARGET_WINS relates to TEAM_BATTING_H and color-coded the data points based on the flag variable. This seems like a good way to visually analyze the data.




**The coefficients from each of the three procedures for all my models are below:**

**Model v1**

**PROC REG**

| Variable | Parameter Estimate |
|---|---|
| Intercept | -104.46878 |
| TEAM_BATTING_H | 0.03893 |
| TEAM_BATTING_2B | -0.03056 |
| TEAM_BATTING_3B | 0.09502 |
| TEAM_BATTING_HR | 0.15094 |
| TEAM_BATTING_BB | 0.05111 |
| IMP_TEAM_BATTING_SO | -0.01447 |
| IMP_TEAM_BASERUN_SB | 0.02926 |
| IMP_TEAM_BASERUN_CS | -0.01085 |
| TEAM_PITCHING_H_LOG | 20.00236 |
| IMP_TEAM_PITCHING_HR | -0.09569 |
| TEAM_PITCHING_BB_TRIM | -0.03639 |
| TEAM_PITCHING_SO_TRIM | 0.01200 |
| TEAM_FIELDING_E | -0.04439 |
| IMP3_TEAM_FIELDING_DP | -0.13995 |

**PROC GLM**

| Parameter | Estimate |
|---|---|
| Intercept | -104.4687835 |
| TEAM_BATTING_H | 0.0389264 |
| TEAM_BATTING_2B | -0.0305645 |
| TEAM_BATTING_3B | 0.0950164 |
| TEAM_BATTING_HR | 0.1509440 |
| TEAM_BATTING_BB | 0.0511062 |
| IMP_TEAM_BATTING_SO | -0.0144726 |
| IMP_TEAM_BASERUN_SB | 0.0292597 |
| IMP_TEAM_BASERUN_CS | -0.0108541 |
| TEAM_PITCHING_H_LOG | 20.0023553 |
| IMP_TEAM_PITCHING_HR | -0.0956871 |
| TEAM_PITCHING_BB_TRI | -0.0363857 |
| TEAM_PITCHING_SO_TRI | 0.0119964 |
| TEAM_FIELDING_E | -0.0443938 |
| IMP3_TEAM_FIELDING_D | -0.1399490 |

**PROC GENMOD**

| Parameter | DF | Estimate |
|---|---|---|
| Intercept | 1 | -104.469 |
| TEAM_BATTING_H | 1 | 0.0389 |
| TEAM_BATTING_2B | 1 | -0.0306 |
| TEAM_BATTING_3B | 1 | 0.0950 |
| TEAM_BATTING_HR | 1 | 0.1509 |
| TEAM_BATTING_BB | 1 | 0.0511 |
| IMP_TEAM_BATTING_SO | 1 | -0.0145 |
| IMP_TEAM_BASERUN_SB | 1 | 0.0293 |
| IMP_TEAM_BASERUN_CS | 1 | -0.0109 |
| TEAM_PITCHING_H_LOG | 1 | 20.0024 |
| IMP_TEAM_PITCHING_HR | 1 | -0.0957 |
| TEAM_PITCHING_BB_TRI | 1 | -0.0364 |
| TEAM_PITCHING_SO_TRI | 1 | 0.0120 |
| TEAM_FIELDING_E | 1 | -0.0444 |
| IMP3_TEAM_FIELDING_D | 1 | -0.1399 |
| Scale | 1 | 12.8522 |

## Model v2

**PROC REG**

| Variable | Parameter Estimate |
|---|---|
| Intercept | 93.67577 |
| TEAM_BATTING_2B | -0.08857 |
| TEAM_BATTING_HR | -0.10669 |
| TEAM_BATTING_BB | 0.02175 |
| IMP_TEAM_BASERUN_SB | 0.02794 |
| IMP_TEAM_BASERUN_CS | -0.02446 |
| TEAM_PITCHING_H_LOG | -8.62949 |
| TEAM_PITCHING_BB_TRIM | -0.03581 |
| TEAM_PITCHING_SO_TRIM | -0.01342 |
| TEAM_FIELDING_E | -0.07625 |
| IMP3_TEAM_FIELDING_DP | -0.11826 |
| TOTAL_BASES_TOUCHED | 0.03150 |
| TOTAL_BASES_GIVEN | 0.01240 |
| m_TEAM_FIELDING_DP | 2.47435 |
| m_TEAM_BASERUN_SB | 39.83820 |
| m_TEAM_PITCHING_SO | 7.59126 |
| m_TEAM_BASERUN_CS | 1.20322 |

**PROC GLM**

| Parameter | Estimate |
|---|---|
| Intercept | 93.67576950 |
| TEAM_BATTING_2B | -0.08857334 |
| TEAM_BATTING_HR | -0.10668668 |
| TEAM_BATTING_BB | 0.02175414 |
| IMP_TEAM_BASERUN_SB | 0.02794226 |
| IMP_TEAM_BASERUN_CS | -0.02445914 |
| TEAM_PITCHING_H_LOG | -8.62948804 |
| TEAM_PITCHING_BB_TRI | -0.03580818 |
| TEAM_PITCHING_SO_TRI | -0.01342306 |
| TEAM_FIELDING_E | -0.07624607 |
| IMP3_TEAM_FIELDING_D | -0.11825998 |
| TOTAL_BASES_TOUCHED | 0.03149933 |
| TOTAL_BASES_GIVEN | 0.01239833 |
| m_TEAM_FIELDING_DP | 2.47434594 |
| m_TEAM_BASERUN_SB | 39.83819971 |
| m_TEAM_PITCHING_SO | 7.59126223 |
| m_TEAM_BASERUN_CS | 1.20321838 |

**PROC GENMOD**

| Parameter | DF | Estimate |
|---|---|---|
| Intercept | 1 | 93.6758 |
| TEAM_BATTING_2B | 1 | -0.0886 |
| TEAM_BATTING_HR | 1 | -0.1067 |
| TEAM_BATTING_BB | 1 | 0.0218 |
| IMP_TEAM_BASERUN_SB | 1 | 0.0279 |
| IMP_TEAM_BASERUN_CS | 1 | -0.0245 |
| TEAM_PITCHING_H_LOG | 1 | -8.6295 |
| TEAM_PITCHING_BB_TRI | 1 | -0.0358 |
| TEAM_PITCHING_SO_TRI | 1 | -0.0134 |
| TEAM_FIELDING_E | 1 | -0.0762 |
| IMP3_TEAM_FIELDING_D | 1 | -0.1183 |
| TOTAL_BASES_TOUCHED | 1 | 0.0315 |
| TOTAL_BASES_GIVEN | 1 | 0.0124 |
| m_TEAM_FIELDING_DP | 1 | 2.4743 |
| m_TEAM_BASERUN_SB | 1 | 39.8382 |
| m_TEAM_PITCHING_SO | 1 | 7.5913 |
| m_TEAM_BASERUN_CS | 1 | 1.2032 |
| Scale | 1 | 11.7153 |

## Model v3

**PROC REG**

| Variable | Parameter Estimate |
|---|---|
| Intercept | 91.84478 |
| TEAM_BATTING_H | 0.03265 |
| TEAM_BATTING_2B | -0.02673 |
| TEAM_BATTING_3B | 0.09201 |
| TEAM_BATTING_HR | 0.01751 |
| TEAM_BATTING_BB | 0.05098 |
| IMP_TEAM_BASERUN_SB | 0.06068 |
| IMP_TEAM_BASERUN_CS | -0.03090 |
| TEAM_PITCHING_H_LOG | -8.23266 |
| TEAM_PITCHING_BB_TRIM | -0.03332 |
| TEAM_PITCHING_SO_TRIM | -0.01368 |
| TEAM_FIELDING_E | -0.07595 |
| IMP3_TEAM_FIELDING_DP | -0.12137 |
| TOTAL_BASES_GIVEN | 0.01192 |
| m_TEAM_BATTING_SO | 7.88521 |
| m_TEAM_FIELDING_DP | 2.48453 |
| m_TEAM_BASERUN_SB | 40.15057 |

**PROC GLM**

| Parameter | Estimate |
|---|---|
| Intercept | 91.84478474 |
| TEAM_BATTING_H | 0.03265374 |
| TEAM_BATTING_2B | -0.02672735 |
| TEAM_BATTING_3B | 0.09201175 |
| TEAM_BATTING_HR | 0.01751337 |
| TEAM_BATTING_BB | 0.05098023 |
| IMP_TEAM_BASERUN_SB | 0.06068489 |
| IMP_TEAM_BASERUN_CS | -0.03089936 |
| TEAM_PITCHING_H_LOG | -8.23265593 |
| TEAM_PITCHING_BB_TRI | -0.03331920 |
| TEAM_PITCHING_SO_TRI | -0.01367816 |
| TEAM_FIELDING_E | -0.07594767 |
| IMP3_TEAM_FIELDING_D | -0.12137179 |
| TOTAL_BASES_GIVEN | 0.01191522 |
| m_TEAM_BATTING_SO | 7.88520988 |
| m_TEAM_FIELDING_DP | 2.48452813 |
| m_TEAM_BASERUN_SB | 40.15056668 |

**PROC GENMOD**

| Parameter | DF | Estimate |
|---|---|---|
| Intercept | 1 | 91.8448 |
| TEAM_BATTING_H | 1 | 0.0327 |
| TEAM_BATTING_2B | 1 | -0.0267 |
| TEAM_BATTING_3B | 1 | 0.0920 |
| TEAM_BATTING_HR | 1 | 0.0175 |
| TEAM_BATTING_BB | 1 | 0.0510 |
| IMP_TEAM_BASERUN_SB | 1 | 0.0607 |
| IMP_TEAM_BASERUN_CS | 1 | -0.0309 |
| TEAM_PITCHING_H_LOG | 1 | -8.2327 |
| TEAM_PITCHING_BB_TRI | 1 | -0.0333 |
| TEAM_PITCHING_SO_TRI | 1 | -0.0137 |
| TEAM_FIELDING_E | 1 | -0.0759 |
| IMP3_TEAM_FIELDING_D | 1 | -0.1214 |
| TOTAL_BASES_GIVEN | 1 | 0.0119 |
| m_TEAM_BATTING_SO | 1 | 7.8852 |
| m_TEAM_FIELDING_DP | 1 | 2.4845 |
| m_TEAM_BASERUN_SB | 1 | 40.1506 |
| Scale | 1 | 11.7201 |

## Model v4

**PROC REG**

| Variable | Parameter Estimate |
|---|---|
| Intercept | 10.13693 |
| TEAM_BATTING_H | 0.04141 |
| TEAM_BATTING_3B | 0.06809 |
| TEAM_BATTING_HR | 0.06852 |
| TEAM_BATTING_BB | 0.01873 |
| IMP_TEAM_BATTING_SO | -0.01196 |
| IMP_TEAM_BASERUN_SB | 0.05641 |
| TEAM_FIELDING_E | -0.04370 |
| m_TEAM_BATTING_SO | 11.95759 |
| m_TEAM_BASERUN_SB | 31.10365 |

**PROC GLM**

| Parameter | Estimate |
|---|---|
| Intercept | 10.13693132 |
| TEAM_BATTING_H | 0.04140767 |
| TEAM_BATTING_3B | 0.06809246 |
| TEAM_BATTING_HR | 0.06851817 |
| TEAM_BATTING_BB | 0.01872658 |
| IMP_TEAM_BATTING_SO | -0.01196300 |
| IMP_TEAM_BASERUN_SB | 0.05641125 |
| TEAM_FIELDING_E | -0.04369770 |
| m_TEAM_BATTING_SO | 11.95758630 |
| m_TEAM_BASERUN_SB | 31.10364885 |

**PROC GENMOD**

| Parameter | DF | Estimate |
|---|---|---|
| Intercept | 1 | 10.1369 |
| TEAM_BATTING_H | 1 | 0.0414 |
| TEAM_BATTING_3B | 1 | 0.0681 |
| TEAM_BATTING_HR | 1 | 0.0685 |
| TEAM_BATTING_BB | 1 | 0.0187 |
| IMP_TEAM_BATTING_SO | 1 | -0.0120 |
| IMP_TEAM_BASERUN_SB | 1 | 0.0564 |
| TEAM_FIELDING_E | 1 | -0.0437 |
| m_TEAM_BATTING_SO | 1 | 11.9576 |
| m_TEAM_BASERUN_SB | 1 | 31.1036 |
| Scale | 1 | 12.2884 |

**Appendix B: Bingo Bonus – Part 2**
**Recreate Assignment 1 in R**

******I believe I should receive 20 Bingo Bonus Points for this. *******

## Format the data

```
attach(Moneyball)

IMP_TEAM_BATTING_SO <- ifelse(TEAM_BATTING_SO=='.',125,TEAM_BATTING_SO)
M_TEAM_BATTING_SO <- ifelse(TEAM_BATTING_SO=='.',1,0)
IMP_TEAM_FIELDING_DP <- ifelse(TEAM_FIELDING_DP=='.',125,TEAM_FIELDING_DP)
M_TEAM_FIELDING_DP <- ifelse(TEAM_FIELDING_DP=='.',1,0)
IMP_TEAM_BASERUN_SB <- ifelse(TEAM_BASERUN_SB=='.',125,TEAM_BASERUN_SB)
M_TEAM_BASERUN_SB <- ifelse(TEAM_BASERUN_SB=='.',1,0)
IMP_TEAM_PITCHING_SO <- ifelse(TEAM_PITCHING_SO=='.',125,TEAM_PITCHING_SO)
M_TEAM_PITCHING_SO <- ifelse(TEAM_PITCHING_SO=='.',1,0)
IMP_TEAM_PITCHING_HR <- ifelse(TEAM_PITCHING_HR=='.',125,TEAM_PITCHING_HR)
M_TEAM_PITCHING_HR <- ifelse(TEAM_PITCHING_HR=='.',1,0)
IMP_TEAM_BASERUN_CS <- ifelse(TEAM_BASERUN_CS=='.',125,TEAM_BASERUN_CS)
M_TEAM_BASERUN_CS <- ifelse(TEAM_BASERUN_CS=='.',1,0)
TEAM_PITCHING_H_LOG <- log(TEAM_PITCHING_H)
TEAM_PITCHING_H_TRIM <- ifelse(TEAM_PITCHING_H >7000,7000,TEAM_PITCHING_H)
TEAM_PITCHING_SO_TRIM <- ifelse(IMP_TEAM_PITCHING_SO >2310,2310,IMP_TEAM_PITCHING_SO)
TEAM_PITCHING_BB_TRIM <- ifelse(TEAM_PITCHING_BB >797,797,TEAM_PITCHING_BB)
TOTAL_BASES_TOUCHED <- (TEAM_BATTING_HR *4) + (TEAM_BATTING_3B * 3) + (TEAM_BATTING_2B * 2) +
TEAM_BATTING_H + TEAM_BATTING_BB + IMP_TEAM_BASERUN_SB
TOTAL_BASES_GIVEN <- (TEAM_PITCHING_HR * 4) + TEAM_PITCHING_H_TRIM + TEAM_PITCHING_BB_TRIM
IMP3_TEAM_FIELDING_DP <- TEAM_FIELDING_DP
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E <
200,152,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >= 200 & TEAM_FIELDING_E <
300,140,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >= 300 & TEAM_FIELDING_E <
400,129,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >= 400 & TEAM_FIELDING_E <
500,118,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >= 500 & TEAM_FIELDING_E <
600,108,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >= 600 & TEAM_FIELDING_E <
700,196,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >= 700 & TEAM_FIELDING_E <
800,86,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >= 800 & TEAM_FIELDING_E <
900,76,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >= 900 & TEAM_FIELDING_E <
1000,68,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELIDNG_DP <- ifelse(M_TEAM_FIELDING_DP==1 & TEAM_FIELDING_E >=
1000,60,IMP3_TEAM_FIELDING_DP)
IMP3_TEAM_FIELDING_DP <- ifelse(IMP3_TEAM_FIELDING_DP=='.',60,IMP3_TEAM_FIELDING_DP)
```

## Create a new data set with the original and new fields

```
Moneyball_Temp<-data.frame(Moneyball,IMP_TEAM_BATTING_SO, M_TEAM_BATTING_SO,
IMP_TEAM_FIELDING_DP, M_TEAM_FIELDING_DP, IMP_TEAM_BASERUN_SB, M_TEAM_BASERUN_SB,
```

IMP_TEAM_PITCHING_SO, M_TEAM_PITCHING_SO, IMP_TEAM_PITCHING_HR, M_TEAM_PITCHING_HR, IMP_TEAM_BASERUN_CS, M_TEAM_BASERUN_CS, TEAM_PITCHING_H_LOG, TEAM_PITCHING_H_TRIM, TEAM_PITCHING_SO_TRIM, TEAM_PITCHING_BB_TRIM, IMP3_TEAM_FIELDING_DP, TOTAL_BASES_TOUCHED, TOTAL_BASES_GIVEN)

## Build Model v1

mod_v1<-lm(TARGET_WINS~TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO + IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS + TEAM_PITCHING_H_LOG + IMP_TEAM_PITCHING_HR + TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM + TEAM_FIELDING_E + IMP3_TEAM_FIELDING_DP, data=Moneyball_Temp)

step(mod_v1, direction="forward")

```
> step(mod_v1, direction="forward")
Start:  AIC=11823.62
TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
    TEAM_BATTING_BB + IMP_TEAM_BATTING_SO + IMP_TEAM_BASERUN_SB +
    IMP_TEAM_BASERUN_CS + TEAM_PITCHING_H_LOG + IMP_TEAM_PITCHING_HR +
    TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM + TEAM_FIELDING_E +
    IMP3_TEAM_FIELDING_DP


Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO +
    IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS + TEAM_PITCHING_H_LOG +
    IMP_TEAM_PITCHING_HR + TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM +
    TEAM_FIELDING_E + IMP3_TEAM_FIELDING_DP, data = Moneyball_Temp)

Coefficients:
          (Intercept)         TEAM_BATTING_H         TEAM_BATTING_2B         TEAM_BATTING_3B         TEAM_BATTING_BB     IMP_TEAM_BATTING_SO
            -4.366e+01              4.142e-02              -2.181e-02              1.115e-01              3.337e-02              2.479e-04
  IMP_TEAM_BASERUN_SB     IMP_TEAM_BASERUN_CS     TEAM_PITCHING_H_LOG    IMP_TEAM_PITCHING_HR  TEAM_PITCHING_BB_TRIM  TEAM_PITCHING_SO_TRIM
            -9.382e-03              5.217e-02              7.099e+00              3.871e-02             -1.740e-02              4.143e-03
     TEAM_FIELDING_E  IMP3_TEAM_FIELDING_DP
            -2.220e-02             -1.491e-02
```

summary(mod_v1)

```
> summary(mod_v1)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO +
    IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS + TEAM_PITCHING_H_LOG +
    IMP_TEAM_PITCHING_HR + TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM +
    TEAM_FIELDING_E + IMP3_TEAM_FIELDING_DP, data = Moneyball_Temp)

Residuals:
    Min      1Q  Median      3Q     Max
-53.864  -8.760   0.087   8.799  58.009

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -4.366e+01  1.858e+01  -2.349 0.018897 *
TEAM_BATTING_H       4.142e-02  3.632e-03  11.405  < 2e-16 ***
TEAM_BATTING_2B     -2.181e-02  9.089e-03  -2.399 0.016509 *
TEAM_BATTING_3B      1.115e-01  1.634e-02   6.824 1.13e-11 ***
TEAM_BATTING_BB      3.337e-02  8.812e-03   3.787 0.000156 ***
IMP_TEAM_BATTING_SO  2.479e-04  2.305e-03   0.108 0.914355
IMP_TEAM_BASERUN_SB -9.382e-03  2.391e-03  -3.923 8.99e-05 ***
IMP_TEAM_BASERUN_CS  5.217e-02  1.278e-02   4.081 4.64e-05 ***
TEAM_PITCHING_H_LOG  7.099e+00  2.814e+00   2.523 0.011713 *
IMP_TEAM_PITCHING_HR 3.871e-02  7.287e-03   5.311 1.19e-07 ***
TEAM_PITCHING_BB_TRIM -1.740e-02  7.610e-03  -2.287 0.022302 *
TEAM_PITCHING_SO_TRIM  4.143e-03  2.214e-03   1.871 0.061419 .
TEAM_FIELDING_E     -2.220e-02  2.907e-03  -7.637 3.26e-14 ***
IMP3_TEAM_FIELDING_DP -1.491e-02  1.050e-02  -1.419 0.155950
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.39 on 2262 degrees of freedom
Multiple R-squared:  0.2817, Adjusted R-squared:  0.2776
F-statistic: 68.25 on 13 and 2262 DF,  p-value: < 2.2e-16
```

**Build Model v2**

Moneyball_Temp_v2<-data.frame(TARGET_WINS,IMP_TEAM_BATTING_SO, M_TEAM_BATTING_SO, IMP_TEAM_FIELDING_DP, M_TEAM_FIELDING_DP, IMP_TEAM_BASERUN_SB, M_TEAM_BASERUN_SB, IMP_TEAM_PITCHING_SO, IMP_TEAM_PITCHING_HR, M_TEAM_BASERUN_CS, TEAM_PITCHING_H_LOG, TEAM_PITCHING_H_TRIM, TEAM_PITCHING_BB_TRIM, TOTAL_BASES_TOUCHED)

leaps( x=Moneyball_Temp_v2[,2:14], y=Moneyball_Temp_v2[,1], names=names(Moneyball_Temp_v2)[2:14], method="adjr2", nbest=1)

```
> leaps( x=Moneyball_Temp[,2:14], y=Moneyball_Temp[,1], names=names(Moneyball_Temp)[2:14], method="adjr2", nbest=1)
$which
  IMP_TEAM_BATTING_SO M_TEAM_BATTING_SO IMP_TEAM_FIELDING_DP M_TEAM_FIELDING_DP IMP_TEAM_BASERUN_SB M_TEAM_BASERUN_SB
1               FALSE              FALSE                FALSE              FALSE               FALSE              FALSE
2               FALSE              FALSE                FALSE              FALSE                TRUE              FALSE
3               FALSE              FALSE                FALSE              FALSE                TRUE              FALSE
4               FALSE               TRUE                FALSE              FALSE                TRUE              FALSE
5               FALSE               TRUE                FALSE              FALSE                TRUE               TRUE
```

summary(mod_v2)

```
> summary(mod_v2)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_HR +
    TEAM_BATTING_BB + IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS +
    TEAM_PITCHING_H_LOG + TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM +
    TEAM_FIELDING_E + IMP3_TEAM_FIELDING_DP + TOTAL_BASES_TOUCHED +
    TOTAL_BASES_GIVEN + M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB +
    M_TEAM_PITCHING_SO + M_TEAM_BASERUN_CS, data = Moneyball_Temp)

Residuals:
    Min      1Q  Median      3Q     Max
-66.456  -8.075   0.357   8.434  45.810

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           2.174998  32.220920   0.068   0.9462
TEAM_BATTING_2B      -0.107838   0.010948  -9.850  < 2e-16 ***
TEAM_BATTING_HR      -0.154803   0.011989 -12.912  < 2e-16 ***
TEAM_BATTING_BB       0.018985   0.008954   2.120   0.0341 *
IMP_TEAM_BASERUN_SB  -0.049929   0.003118 -16.015  < 2e-16 ***
IMP_TEAM_BASERUN_CS  -0.022842   0.016223  -1.408   0.1593
TEAM_PITCHING_H_LOG  -0.648114   4.848185  -0.134   0.8937
TEAM_PITCHING_BB_TRIM -0.034963   0.007144  -4.894 1.06e-06 ***
TEAM_PITCHING_SO_TRIM  0.007410   0.001165   6.360 2.44e-10 ***
TEAM_FIELDING_E      -0.051749   0.003564 -14.520  < 2e-16 ***
IMP3_TEAM_FIELDING_DP -0.041658   0.010018  -4.158 3.33e-05 ***
TOTAL_BASES_TOUCHED   0.042015   0.002090  20.106  < 2e-16 ***
TOTAL_BASES_GIVEN     0.006723   0.001581   4.253 2.20e-05 ***
M_TEAM_FIELDING_DP    7.457364   1.342897   5.553 3.13e-08 ***
M_TEAM_BASERUN_SB    27.157930   1.693774  16.034  < 2e-16 ***
M_TEAM_PITCHING_SO   15.082374   1.555922   9.694  < 2e-16 ***
M_TEAM_BASERUN_CS     5.206109   1.145482   4.545 5.79e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.44 on 2259 degrees of freedom
Multiple R-squared:  0.3807, Adjusted R-squared:  0.3764
F-statistic: 86.81 on 16 and 2259 DF,  p-value: < 2.2e-16
```

## Build Model v3

mod_v3 <- lm(TARGET_WINS~TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO + IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS + TEAM_PITCHING_H_LOG + IMP_TEAM_PITCHING_HR + TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM + TEAM_FIELDING_E + IMP3_TEAM_FIELDING_DP + M_TEAM_BATTING_SO + M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB + M_TEAM_PITCHING_SO + M_TEAM_BASERUN_CS + TOTAL_BASES_TOUCHED + TOTAL_BASES_GIVEN, data=Moneyball_Temp)

step(mod_v3, direction="backward")

```
Step:  AIC=11489.81
TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
    TEAM_BATTING_BB + IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS +
    TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM + TEAM_FIELDING_E +
    IMP3_TEAM_FIELDING_DP + M_TEAM_BATTING_SO + M_TEAM_FIELDING_DP +
    M_TEAM_BASERUN_SB + M_TEAM_BASERUN_CS + TOTAL_BASES_GIVEN

                      Df Sum of Sq    RSS   AIC
<none>                             349536 11490
- IMP_TEAM_BASERUN_CS    1       397 349933 11490
- TEAM_BATTING_2B        1      1469 351004 11497
- IMP_TEAM_BASERUN_SB    1      1850 351386 11500
- IMP3_TEAM_FIELDING_DP  1      2713 352249 11505
- M_TEAM_BASERUN_CS      1      3271 352807 11509
- TEAM_PITCHING_BB_TRIM  1      4438 353974 11516
- M_TEAM_FIELDING_DP     1      5842 355378 11526
- TEAM_PITCHING_SO_TRIM  1      6226 355762 11528
- TEAM_BATTING_3B        1      6333 355869 11529
- TOTAL_BASES_GIVEN      1     11179 360715 11560
- TEAM_BATTING_BB        1     11579 361114 11562
- M_TEAM_BATTING_SO      1     14776 364312 11582
- TEAM_BATTING_H         1     28896 378432 11669
- TEAM_FIELDING_E        1     42780 392316 11751
- M_TEAM_BASERUN_SB      1     43569 393104 11755

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_BB + IMP_TEAM_BASERUN_SB +
    IMP_TEAM_BASERUN_CS + TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM +
    TEAM_FIELDING_E + IMP3_TEAM_FIELDING_DP + M_TEAM_BATTING_SO +
    M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB + M_TEAM_BASERUN_CS +
    TOTAL_BASES_GIVEN, data = Moneyball_Temp)

Coefficients:
          (Intercept)         TEAM_BATTING_H         TEAM_BATTING_2B         TEAM_BATTING_3B         TEAM_BATTING_BB     IMP_TEAM_BASERUN_SB
            -4.044949               0.045406              -0.026432               0.098345               0.063676              -0.007775
  IMP_TEAM_BASERUN_CS  TEAM_PITCHING_BB_TRIM  TEAM_PITCHING_SO_TRIM         TEAM_FIELDING_E  IMP3_TEAM_FIELDING_DP       M_TEAM_BATTING_SO
            -0.025913              -0.036223               0.007385              -0.053289              -0.041920              15.111187
   M_TEAM_FIELDING_DP      M_TEAM_BASERUN_SB      M_TEAM_BASERUN_CS      TOTAL_BASES_GIVEN
             7.980990              27.609668               5.262646               0.006657
```

summary(mod_v3)

```
> summary(mod_v3)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO +
    IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS + TEAM_PITCHING_H_LOG +
    IMP_TEAM_PITCHING_HR + TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM +
    TEAM_FIELDING_E + IMP3_TEAM_FIELDING_DP + M_TEAM_BATTING_SO +
    M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB + M_TEAM_PITCHING_SO +
    M_TEAM_BASERUN_CS + TOTAL_BASES_TOUCHED + TOTAL_BASES_GIVEN,
    data = Moneyball_Temp)

Residuals:
    Min      1Q  Median      3Q     Max
-63.192  -8.166   0.295   8.537  43.996

Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         11.769115  33.221836   0.354 0.723178
TEAM_BATTING_H       0.038280   0.007838   4.884 1.11e-06 ***
TEAM_BATTING_2B     -0.043982   0.017028  -2.583 0.009858 **
TEAM_BATTING_3B      0.084298   0.022685   3.716 0.000207 ***
TEAM_BATTING_BB      0.048529   0.013552   3.581 0.000350 ***
IMP_TEAM_BATTING_SO  0.002358   0.002183   1.080 0.280040
IMP_TEAM_BASERUN_SB -0.015919   0.007385  -2.156 0.031215 *
IMP_TEAM_BASERUN_CS -0.024202   0.016263  -1.488 0.136864
TEAM_PITCHING_H_LOG -2.627733   5.037188  -0.522 0.601954
IMP_TEAM_PITCHING_HR -0.022914   0.026898  -0.852 0.394359
TEAM_PITCHING_BB_TRIM -0.030582   0.007914  -3.864 0.000114 ***
TEAM_PITCHING_SO_TRIM  0.005417   0.002111   2.566 0.010347 *
TEAM_FIELDING_E      -0.051320   0.003692 -13.902  < 2e-16 ***
IMP3_TEAM_FIELDING_DP -0.042036   0.010026  -4.193 2.86e-05 ***
M_TEAM_BATTING_SO    15.139177   1.556087   9.729  < 2e-16 ***
M_TEAM_FIELDING_DP    7.441642   1.351894   5.505 4.12e-08 ***
M_TEAM_BASERUN_SB    27.236565   1.694158  16.077  < 2e-16 ***
M_TEAM_PITCHING_SO         NA         NA      NA       NA
M_TEAM_BASERUN_CS     5.531519   1.159346   4.771 1.95e-06 ***
TOTAL_BASES_TOUCHED   0.008069   0.006992   1.154 0.248625
TOTAL_BASES_GIVEN     0.007010   0.001711   4.097 4.33e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.44 on 2256 degrees of freedom
Multiple R-squared:  0.3818, Adjusted R-squared:  0.3766
F-statistic: 73.33 on 19 and 2256 DF,  p-value: < 2.2e-16
```

## Build Model v4

mod_v4 <- lm(TARGET_WINS~TEAM_BATTING_H + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO + IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS + TEAM_PITCHING_BB_TRIM + TEAM_PITCHING_SO_TRIM + TEAM_FIELDING_E + M_TEAM_BATTING_SO + M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB + M_TEAM_PITCHING_SO + M_TEAM_BASERUN_CS, data=Moneyball_Temp)

step(mod_v4, direction="backward")

```
Step:  AIC=11557.82
TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B + TEAM_BATTING_HR +
    TEAM_BATTING_BB + IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS +
    TEAM_PITCHING_SO_TRIM + TEAM_FIELDING_E + M_TEAM_BATTING_SO +
    M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB + M_TEAM_BASERUN_CS

                      Df Sum of Sq    RSS   AIC
<none>                             361089 11558
- IMP_TEAM_BASERUN_CS    1       612 361701 11560
- M_TEAM_BASERUN_CS      1      2583 363673 11572
- M_TEAM_FIELDING_DP     1      2599 363688 11572
- IMP_TEAM_BASERUN_SB    1      2629 363718 11572
- TEAM_BATTING_HR        1      3286 364376 11576
- TEAM_BATTING_3B        1      4834 365923 11586
- TEAM_PITCHING_SO_TRIM  1      6850 367939 11599
- TEAM_BATTING_BB        1     10124 371214 11619
- M_TEAM_BATTING_SO      1     13243 374332 11638
- TEAM_FIELDING_E        1     29813 390902 11736
- M_TEAM_BASERUN_SB      1     32776 393865 11754
- TEAM_BATTING_H         1     71328 432417 11966

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B +
    TEAM_BATTING_HR + TEAM_BATTING_BB + IMP_TEAM_BASERUN_SB +
    IMP_TEAM_BASERUN_CS + TEAM_PITCHING_SO_TRIM + TEAM_FIELDING_E +
    M_TEAM_BATTING_SO + M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB +
    M_TEAM_BASERUN_CS, data = Moneyball_Temp)

Coefficients:
          (Intercept)         TEAM_BATTING_H         TEAM_BATTING_3B         TEAM_BATTING_HR         TEAM_BATTING_BB     IMP_TEAM_BASERUN_SB
            -7.011466               0.049256               0.087080               0.035116               0.026018              -0.009245
  IMP_TEAM_BASERUN_CS  TEAM_PITCHING_SO_TRIM         TEAM_FIELDING_E       M_TEAM_BATTING_SO      M_TEAM_FIELDING_DP      M_TEAM_BASERUN_SB
            -0.032165               0.007542              -0.035787              13.840622               5.281291              23.586424
    M_TEAM_BASERUN_CS
             4.612665
```

summary(mod_v4)

```
> summary(mod_v4)

Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B +
    TEAM_BATTING_HR + TEAM_BATTING_BB + IMP_TEAM_BATTING_SO +
    IMP_TEAM_BASERUN_SB + IMP_TEAM_BASERUN_CS + TEAM_PITCHING_BB_TRIM +
    TEAM_PITCHING_SO_TRIM + TEAM_FIELDING_E + M_TEAM_BATTING_SO +
    M_TEAM_FIELDING_DP + M_TEAM_BASERUN_SB + M_TEAM_PITCHING_SO +
    M_TEAM_BASERUN_CS, data = Moneyball_Temp)

Residuals:
    Min      1Q  Median      3Q     Max
-46.474  -8.217   0.078   8.856  43.840

Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -7.630715   3.835490  -1.990 0.046766 *
TEAM_BATTING_H        0.049429   0.002357  20.974  < 2e-16 ***
TEAM_BATTING_3B       0.088590   0.015871   5.582 2.66e-08 ***
TEAM_BATTING_HR       0.035076   0.007745   4.529 6.23e-06 ***
TEAM_BATTING_BB       0.024848   0.006382   3.893 0.000102 ***
IMP_TEAM_BATTING_SO   0.002556   0.002192   1.166 0.243792
IMP_TEAM_BASERUN_SB  -0.009055   0.002283  -3.967 7.50e-05 ***
IMP_TEAM_BASERUN_CS  -0.033115   0.016450  -2.013 0.044232 *
TEAM_PITCHING_BB_TRIM  0.001295   0.005209   0.249 0.803660
TEAM_PITCHING_SO_TRIM  0.005551   0.002097   2.647 0.008185 **
TEAM_FIELDING_E      -0.035965   0.002909 -12.364  < 2e-16 ***
M_TEAM_BATTING_SO     13.920274   1.526265   9.120  < 2e-16 ***
M_TEAM_FIELDING_DP     5.164184   1.313998   3.930 8.74e-05 ***
M_TEAM_BASERUN_SB     23.543252   1.648116  14.285  < 2e-16 ***
M_TEAM_PITCHING_SO          NA         NA      NA       NA
M_TEAM_BASERUN_CS      4.796447   1.156303   4.148 3.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.63 on 2261 degrees of freedom
Multiple R-squared:  0.3608, Adjusted R-squared:  0.3568
F-statistic: 91.15 on 14 and 2261 DF,  p-value: < 2.2e-16
```

**Compare AIC Values**

```
extractAIC(mod_v1)
extractAIC(mod_v2)
extractAIC(mod_v3)
extractAIC(mod_v4)
```

```
> extractAIC(mod_v1)
[1]    14.00 11823.62
> extractAIC(mod_v2)
[1]    17.00 11492.01
> extractAIC(mod_v3)
[1]    20.0 11494.2
> extractAIC(mod_v4)
[1]    15.00 11560.22
```

**Comparison of SAS and R**

My opinion of SAS vs. R is probably skewed by the fact that I have very little experience with R, but at this point I find SAS much more user-friendly and easy to use. I like how you can run PROC REG in SAS and get all of the diagnostics presented to you on one page. It takes a bunch of different commands to do the same thing in R. I also had a very hard time getting R to do things like select the 3 best model for each different number of variables like you can do in SAS with the /Selection BEST=3 command. I eventually found the LEAPS package that allowed you do to that, but it was very picky about the data and the format it was in. I'm not ready to give up on R quite yet though. I think I will like it better as I become more familiar with it.

## Appendix C: SAS Code

**Correlation**

```
ods graphics on;
proc corr data= mydata.moneyball plot=matrix(histogram nvar=all);
run;
ods graphics off;
```

**Missing Values**

```
title "Variables With Missing Values";
proc means data=mydata.moneyball n nmiss;
var team_batting_so team_baserun_sb team_baserun_cs team_batting_hbp team_pitching_so team_fielding_dp;
run;
```

**Scatterplot**

```
title "Pitching Strikeouts";
proc sgplot data = moneyball_temp_v10;
        scatter x=TEAM_PITCHING_SO y=TARGET_WINS;
run;
quit
```

**Histogram**

```
Title "Before Trim";
proc sgplot data=moneyball_temp_v10;
histogram TEAM_PITCHING_BB;
run;

Title "After Trim";
proc sgplot data=moneyball_temp_v10;
histogram TEAM_PITCHING_BB_TRIM;
run;
```

**Data Prep for Model**

```
libname mydata '/home/johnboggio2014/my_courses/donald.wedding/c_8888/PRED411/UNIT01/HW/'
access=readonly;

data moneyball_temp_v100;
set mydata.moneyball;

TEAM_BASERUN_CSP = TEAM_BASERUN_CS / TEAM_BASERUN_SB;

/************************** Remove outlier record **************************/
if target_wins = 12 then delete;
```

```
/************************** Impute Missing Values  **************************/

IMP_TEAM_BATTING_SO = TEAM_BATTING_SO;
m_TEAM_BATTING_SO = 0;
If missing(IMP_TEAM_BATTING_SO) or IMP_TEAM_BATTING_SO = 0 then do;
        IMP_TEAM_BATTING_SO = 736;
        m_TEAM_BATTING_SO = 1;
END;

IMP_TEAM_FIELDING_DP = TEAM_FIELDING_DP;
m_TEAM_FIELDING_DP = 0;
IF missing(IMP_TEAM_FIELDING_DP) or IMP_TEAM_FIELDING_DP = 0 then do;
        IMP_TEAM_FIELDING_DP = 149;
        m_TEAM_FIELDING_DP = 1;
END;

IMP_TEAM_BASERUN_SB = TEAM_BASERUN_SB;
m_TEAM_BASERUN_SB = 0;
IF missing(IMP_TEAM_BASERUN_SB) or IMP_TEAM_BASERUN_SB = 0 then do;
        IMP_TEAM_BASERUN_SB = 125;
        m_TEAM_BASERUN_SB = 1;
END;

IMP_TEAM_PITCHING_SO = TEAM_PITCHING_SO;
m_TEAM_PITCHING_SO = 0;
If missing(IMP_TEAM_PITCHING_SO) or IMP_TEAM_PITCHING_SO = 0 then do;
        IMP_TEAM_PITCHING_SO = 818;
        m_TEAM_PITCHING_SO = 1;
END;

IMP_TEAM_PITCHING_HR = TEAM_PITCHING_HR;
m_TEAM_PITCHING_HR = 0;
If missing(IMP_TEAM_PITCHING_HR) or IMP_TEAM_PITCHING_HR = 0 then do;
        IMP_TEAM_PITCHING_HR = 105;
        m_TEAM_PITCHING_HR = 1;
END;

IMP_TEAM_BASERUN_CS = TEAM_BASERUN_CS;
IMP_TEAM_BASERUN_CS2 = TEAM_BASERUN_CS;
m_TEAM_BASERUN_CS = 0;
If missing(IMP_TEAM_BASERUN_CS) or IMP_TEAM_BASERUN_CS = 0 then do;
        IMP_TEAM_BASERUN_CS = 53;
        IMP_TEAM_BASERUN_CS2 = IMP_TEAM_BASERUN_SB * 0.6;
        m_TEAM_BASERUN_CS = 1;
END;



/************************************ 0 - Values ************************************/
If (TEAM_BATTING_3B = 0 AND TEAM_BATTING_H > 1000) then TEAM_BATTING_3B = 55;
```

```
/***************************** Transformations ******************************/
TEAM_PITCHING_H_LOG = log(TEAM_PITCHING_H);

TEAM_PITCHING_H_TRIM = TEAM_PITCHING_H;
If TEAM_PITCHING_H_TRIM > 7000 then TEAM_PITCHING_H_TRIM = 7000;

TEAM_PITCHING_SO_TRIM = IMP_TEAM_PITCHING_SO;
If TEAM_PITCHING_SO_TRIM > 2310 then TEAM_PITCHING_SO_TRIM = 2310;

TEAM_PITCHING_BB_TRIM = TEAM_PITCHING_BB;
If TEAM_PITCHING_BB_TRIM > 797 then TEAM_PITCHING_BB_TRIM = 797;


/********************************* New Fields *******************************/
TOTAL_BASES_TOUCHED = (TEAM_BATTING_HR * 4)
                                          + (TEAM_BATTING_3B * 3)
                                          + (TEAM_BATTING_2B * 2)
                                          + TEAM_BATTING_H
                                          + TEAM_BATTING_BB
                                          + IMP_TEAM_BASERUN_SB;

/** Extra Base Hits **/
EXTRA_BASE_HITS = TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR;


TOTAL_BASES_GIVEN = (TEAM_PITCHING_HR * 4)
                                           + TEAM_PITCHING_H_TRIM
                                           + TEAM_PITCHING_BB_TRIM;


IMP3_TEAM_FIELDING_DP = TEAM_FIELDING_DP;
If M_TEAM_FIELDING_DP = 1 then do;
        If TEAM_FIELDING_E < 200 then IMP3_TEAM_FIELDING_DP = 152;
        else if TEAM_FIELDING_E >= 200 and TEAM_FIELDING_E < 300 then IMP3_TEAM_FIELDING_DP = 140;
        else if TEAM_FIELDING_E >= 300 and TEAM_FIELDING_E < 400 then IMP3_TEAM_FIELDING_DP = 129;
        else if TEAM_FIELDING_E >= 400 and TEAM_FIELDING_E < 500 then IMP3_TEAM_FIELDING_DP = 118;
        else if TEAM_FIELDING_E >= 500 and TEAM_FIELDING_E < 600 then IMP3_TEAM_FIELDING_DP = 108;
        else if TEAM_FIELDING_E >= 600 and TEAM_FIELDING_E < 700 then IMP3_TEAM_FIELDING_DP = 96;
        else if TEAM_FIELDING_E >= 700 and TEAM_FIELDING_E < 800 then IMP3_TEAM_FIELDING_DP = 86;
        else if TEAM_FIELDING_E >= 800 and TEAM_FIELDING_E < 900 then IMP3_TEAM_FIELDING_DP = 76;
        else if TEAM_FIELDING_E >= 900 and TEAM_FIELDING_E < 1000 then IMP3_TEAM_FIELDING_DP = 68;
        else IMP3_TEAM_FIELDING_DP = 60;

END;
```

**Model v1**

```
/********************** Model v1 ************************/
PROC REG data=moneyball_temp_v100;
model target_wins =

TEAM_BATTING_H
TEAM_BATTING_2B
TEAM_BATTING_3B
TEAM_BATTING_HR
TEAM_BATTING_BB
IMP_TEAM_BATTING_SO
IMP_TEAM_BASERUN_SB
IMP_TEAM_BASERUN_CS
TEAM_PITCHING_H_LOG
IMP_TEAM_PITCHING_HR
TEAM_PITCHING_BB_TRIM
TEAM_PITCHING_SO_TRIM
TEAM_FIELDING_E
IMP3_TEAM_FIELDING_DP

/selection = forward;
;
run;


/*************************** Calculate P_TARGET_WINS ***************************/
data moneyball_deploy_v100;
set moneyball_temp_v100;

P_TARGET_WINS = 26.60243
                                    + (0.04208 * TEAM_BATTING_H)
                                    + (-0.02332 * TEAM_BATTING_2B)
                                    + (0.09433 * TEAM_BATTING_3B)
                                    + (0.13978 * TEAM_BATTING_HR)
                                    + (0.04175 * TEAM_BATTING_BB)
                                    + (-0.01841 * IMP_TEAM_BATTING_SO)
                                    + (0.02881 * IMP_TEAM_BASERUN_SB)
                                    + (0.00512 * TEAM_PITCHING_H_TRIM)
                                    + (-0.08087 * IMP_TEAM_PITCHING_HR)
                                    + (-0.02761 * TEAM_PITCHING_BB_TRIM)
                                    + (0.01464 * TEAM_PITCHING_SO_TRIM)
                                    + (-0.04042 * TEAM_FIELDING_E)
                                    + (-0.13564 * IMP3_TEAM_FIELDING_DP);

If P_TARGET_WINS > 113 then P_TARGET_WINS = 113;
IF P_TARGET_WINS < 30 then P_TARGET_WINS = 30;

run;

data MONEYBALL_TEST;
set moneyball_deploy_v100 (keep=INDEX target_wins P_TARGET_WINS);
```

```
proc reg data=moneyball_deploy_v100;
model target_wins = p_target_wins;
run;
```

**Model v2**

```
/********************* Model v2 ************************/
PROC REG data=moneyball_temp_v100;
model target_wins =

TEAM_BATTING_H
TEAM_BATTING_2B
TEAM_BATTING_3B
TEAM_BATTING_HR
TEAM_BATTING_BB
IMP_TEAM_BATTING_SO
IMP_TEAM_BASERUN_SB
IMP_TEAM_BASERUN_CS
TEAM_PITCHING_H_LOG
IMP_TEAM_PITCHING_HR
TEAM_PITCHING_BB_TRIM
TEAM_PITCHING_SO_TRIM
TEAM_FIELDING_E
IMP3_TEAM_FIELDING_DP
TOTAL_BASES_TOUCHED
EXTRA_BASE_HITS
TOTAL_BASES_GIVEN

m_TEAM_BATTING_SO
m_TEAM_FIELDING_DP
m_TEAM_BASERUN_SB
m_TEAM_PITCHING_SO
m_TEAM_BASERUN_CS
/selection = rsquare cp adjrsq best=1;
;
run;


/********************* Run PROC REG for best model ************************/
PROC REG data=moneyball_temp_v100;
 model target_wins =
TEAM_BATTING_2B TEAM_BATTING_HR TEAM_BATTING_BB IMP_TEAM_BASERUN_SB IMP_TEAM_BASERUN_CS
TEAM_PITCHING_H_LOG TEAM_PITCHING_BB_TRIM TEAM_PITCHING_SO_TRIM TEAM_FIELDING_E
IMP3_TEAM_FIELDING_DP TOTAL_BASES_TOUCHED TOTAL_BASES_GIVEN m_TEAM_FIELDING_DP
m_TEAM_BASERUN_SB m_TEAM_PITCHING_SO m_TEAM_BASERUN_CS
;
run;
/************************** Calculate P_TARGET_WINS ************************/
data moneyball_deploy_v100;
```

```
set moneyball_temp_v100;

P_TARGET_WINS = 93.67577
                                    + (-0.08857 * TEAM_BATTING_2B)
                                    + (-0.10669 * TEAM_BATTING_HR)
                                    + (0.02175 * TEAM_BATTING_BB)
                                    + (0.02794 * IMP_TEAM_BASERUN_SB)
                                    + (-0.02446 * IMP_TEAM_BASERUN_CS)
                                    + (-8.62949 * TEAM_PITCHING_H_LOG)
                                    + (-0.03581 * TEAM_PITCHING_BB_TRIM)
                                    + (-0.01342 * TEAM_PITCHING_SO_TRIM)
                                    + (-0.07625 * TEAM_FIELDING_E)
                                    + (-0.11826 * IMP3_TEAM_FIELDING_DP)
                                    + (0.0315 * TOTAL_BASES_TOUCHED)
                                    + (0.0124 * TOTAL_BASES_GIVEN)
                                    + (2.47435 * m_TEAM_FIELDING_DP)
                                    + (39.8382 * m_TEAM_BASERUN_SB)
                                    + (7.59126 * m_TEAM_PITCHING_SO)
                                    + (1.20322 * m_TEAM_BASERUN_CS);

If P_TARGET_WINS > 113 then P_TARGET_WINS = 113;
IF P_TARGET_WINS < 20 then P_TARGET_WINS = 20;


run;

data MONEYBALL_TEST;
set moneyball_deploy_v100 (keep=INDEX target_wins P_TARGET_WINS);

proc reg data=moneyball_deploy_v100;
model target_wins = p_target_wins;
run;
```

**Model v3**

```
/************************** Model v3 *************************/
PROC REG data=moneyball_temp_v100;
model target_wins =

TEAM_BATTING_H
TEAM_BATTING_2B
TEAM_BATTING_3B
TEAM_BATTING_HR
TEAM_BATTING_BB
IMP_TEAM_BATTING_SO
IMP_TEAM_BASERUN_SB
IMP_TEAM_BASERUN_CS
TEAM_PITCHING_H_LOG
IMP_TEAM_PITCHING_HR
TEAM_PITCHING_BB_TRIM
TEAM_PITCHING_SO_TRIM
```

```
TEAM_FIELDING_E
IMP3_TEAM_FIELDING_DP
TOTAL_BASES_TOUCHED
EXTRA_BASE_HITS
TOTAL_BASES_GIVEN
m_TEAM_BATTING_SO
m_TEAM_FIELDING_DP
m_TEAM_BASERUN_SB
m_TEAM_PITCHING_SO
m_TEAM_BASERUN_CS

/selection = backward;
;
run;


/************************** Calculate P_TARGET_WINS **************************/
data moneyball_deploy_v100;
set moneyball_temp_v100;

P_TARGET_WINS = 91.84478
                                    + (0.03265 * TEAM_BATTING_H)
                                    + (-0.02673 * TEAM_BATTING_2B)
                                    + (0.09201 * TEAM_BATTING_3B)
                                    + (0.01751 * TEAM_BATTING_HR)
                                    + (0.05098 * TEAM_BATTING_BB)
                                    + (0.06068 * IMP_TEAM_BASERUN_SB)
                                    + (-0.0309 * IMP_TEAM_BASERUN_CS)
                                    + (-8.23266 * TEAM_PITCHING_H_LOG)
                                    + (-0.03332 * TEAM_PITCHING_BB_TRIM)
                                    + (-0.01368 * TEAM_PITCHING_SO_TRIM)
                                    + (-0.07595 * TEAM_FIELDING_E)
                                    + (-0.12137 * IMP3_TEAM_FIELDING_DP)
                                    + (0.01192 * TOTAL_BASES_GIVEN)
                                    + (7.88521 * m_TEAM_BATTING_SO)
                                    + (2.48453 * m_TEAM_FIELDING_DP)
                                    + (40.15057 * m_TEAM_BASERUN_SB);

If P_TARGET_WINS > 113 then P_TARGET_WINS = 113;
IF P_TARGET_WINS < 20 then P_TARGET_WINS = 20;

run;

data MONEYBALL_TEST;
set moneyball_deploy_v100 (keep=INDEX target_wins P_TARGET_WINS);

proc reg data=moneyball_deploy_v100;
model target_wins = p_target_wins;
run;
```

**Model v4**

```
/************************** Model v4 ***********************************/
```

```
PROC REG data=moneyball_temp_v100;
model target_wins =

TEAM_BATTING_H
TEAM_BATTING_3B
TEAM_BATTING_HR
TEAM_BATTING_BB
IMP_TEAM_BATTING_SO
IMP_TEAM_BASERUN_SB
IMP_TEAM_BASERUN_CS
TEAM_PITCHING_BB_TRIM
TEAM_FIELDING_E
m_TEAM_BATTING_SO
m_TEAM_FIELDING_DP
m_TEAM_BASERUN_SB
m_TEAM_PITCHING_SO
m_TEAM_BASERUN_CS


/selection = backward;
;
run;


/************************** Calculate P_TARGET_WINS **************************/
data moneyball_deploy_v100;
set moneyball_temp_v100;

P_TARGET_WINS_mv = 10.13693
                                        + (0.04141 * TEAM_BATTING_H)
                                        + (0.06809 * TEAM_BATTING_3B)
                                        + (0.06852 * TEAM_BATTING_HR)
                                        + (0.01873 * TEAM_BATTING_BB)
                                        + (-0.01196 * IMP_TEAM_BATTING_SO)
                                        + (0.05641 * IMP_TEAM_BASERUN_SB)
                                        + (-0.0437 * TEAM_FIELDING_E)
                                        + (11.95759 * m_TEAM_BATTING_SO)
                                        + (31.10365 * m_TEAM_BASERUN_SB)
                                        ;

If P_TARGET_WINS_mv > 113 then P_TARGET_WINS_mv = 113;
IF P_TARGET_WINS_mv < 15 then P_TARGET_WINS_mv = 15;

run;

data MONEYBALL_TEST;
set moneyball_deploy_v100 (keep=INDEX target_wins P_TARGET_WINS);

proc reg data=moneyball_deploy_v100;
model target_wins = p_target_wins;
run;
```

**Appendix D: SAS Stand Alone Scoring Program**

```
%macro SCORE( INFILE, OUTFILE );

data &OUTFILE;
set &INFILE;

/************************** Impute Missing Values  **************************/
IMP_TEAM_BATTING_SO = TEAM_BATTING_SO;
m_TEAM_BATTING_SO = 0;
If missing(IMP_TEAM_BATTING_SO) or IMP_TEAM_BATTING_SO = 0 then do;
        IMP_TEAM_BATTING_SO = 736;
        m_TEAM_BATTING_SO = 1;
END;

IMP_TEAM_FIELDING_DP = TEAM_FIELDING_DP;
m_TEAM_FIELDING_DP = 0;
IF missing(IMP_TEAM_FIELDING_DP) or IMP_TEAM_FIELDING_DP = 0 then do;
        IMP_TEAM_FIELDING_DP = 149;
        m_TEAM_FIELDING_DP = 1;
END;

IMP_TEAM_BASERUN_SB = TEAM_BASERUN_SB;
m_TEAM_BASERUN_SB = 0;
IF missing(IMP_TEAM_BASERUN_SB) or IMP_TEAM_BASERUN_SB = 0 then do;
        IMP_TEAM_BASERUN_SB = 125;
        m_TEAM_BASERUN_SB = 1;
END;

IMP_TEAM_PITCHING_SO = TEAM_PITCHING_SO;
m_TEAM_PITCHING_SO = 0;
If missing(IMP_TEAM_PITCHING_SO) or IMP_TEAM_PITCHING_SO = 0 then do;
        IMP_TEAM_PITCHING_SO = 818;
        m_TEAM_PITCHING_SO = 1;
END;

IMP_TEAM_PITCHING_HR = TEAM_PITCHING_HR;
m_TEAM_PITCHING_HR = 0;
If missing(IMP_TEAM_PITCHING_HR) or IMP_TEAM_PITCHING_HR = 0 then do;
        IMP_TEAM_PITCHING_HR = 105;
        m_TEAM_PITCHING_HR = 1;
END;

IMP_TEAM_BASERUN_CS = TEAM_BASERUN_CS;
m_TEAM_BASERUN_CS = 0;
If missing(IMP_TEAM_BASERUN_CS) or IMP_TEAM_BASERUN_CS = 0 then do;
        IMP_TEAM_BASERUN_CS = 53;
        m_TEAM_BASERUN_CS = 1;
END;
```

```
/********************************** 0 - Values **********************************/
If (TEAM_BATTING_3B = 0 AND TEAM_BATTING_H > 1000) then TEAM_BATTING_3B = 55;


/******************************** Transformations ********************************/
TEAM_PITCHING_H_LOG = log(TEAM_PITCHING_H);

TEAM_PITCHING_H_TRIM = TEAM_PITCHING_H;
If TEAM_PITCHING_H_TRIM > 7000 then TEAM_PITCHING_H_TRIM = 7000;

TEAM_PITCHING_SO_TRIM = IMP_TEAM_PITCHING_SO;
If TEAM_PITCHING_SO_TRIM > 2310 then TEAM_PITCHING_SO_TRIM = 2310;

TEAM_PITCHING_BB_TRIM = TEAM_PITCHING_BB;
If TEAM_PITCHING_BB_TRIM > 797 then TEAM_PITCHING_BB_TRIM = 797;


/******************************** New Fields ********************************/
TOTAL_BASES_TOUCHED = (TEAM_BATTING_HR * 4)
                                        + (TEAM_BATTING_3B * 3)
                                        + (TEAM_BATTING_2B * 2)
                                        + TEAM_BATTING_H
                                        + TEAM_BATTING_BB
                                        + IMP_TEAM_BASERUN_SB;

/** Extra Base Hits **/
EXTRA_BASE_HITS = TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR;


TOTAL_BASES_GIVEN = (TEAM_PITCHING_HR * 4)
                                         + TEAM_PITCHING_H_TRIM
                                         + TEAM_PITCHING_BB_TRIM;


IMP3_TEAM_FIELDING_DP = TEAM_FIELDING_DP;
/**
If m_TEAM_FIELDING_DP = 1 AND IMP_TEAM_BASERUN_SB < 50 then IMP2_TEAM_FIELDING_DP = 170;
elseif m_TEAM_FIELDING_DP = 1 AND IMP2_TEAM_BASERUN_SB >= 50 and IMP2_TEAM_BASERUN_SB < 100 then
IMP2_TEAM_FIELDING_DP = 153;
**/
If M_TEAM_FIELDING_DP = 1 then do;
        If TEAM_FIELDING_E < 200 then IMP3_TEAM_FIELDING_DP = 152;
        else if TEAM_FIELDING_E >= 200 and TEAM_FIELDING_E < 300 then IMP3_TEAM_FIELDING_DP = 140;
        else if TEAM_FIELDING_E >= 300 and TEAM_FIELDING_E < 400 then IMP3_TEAM_FIELDING_DP = 129;
        else if TEAM_FIELDING_E >= 400 and TEAM_FIELDING_E < 500 then IMP3_TEAM_FIELDING_DP = 118;
        else if TEAM_FIELDING_E >= 500 and TEAM_FIELDING_E < 600 then IMP3_TEAM_FIELDING_DP = 108;
        else if TEAM_FIELDING_E >= 600 and TEAM_FIELDING_E < 700 then IMP3_TEAM_FIELDING_DP = 96;
        else if TEAM_FIELDING_E >= 700 and TEAM_FIELDING_E < 800 then IMP3_TEAM_FIELDING_DP = 86;
        else if TEAM_FIELDING_E >= 800 and TEAM_FIELDING_E < 900 then IMP3_TEAM_FIELDING_DP = 76;
        else if TEAM_FIELDING_E >= 900 and TEAM_FIELDING_E < 1000 then IMP3_TEAM_FIELDING_DP = 68;
        else IMP3_TEAM_FIELDING_DP = 60;
```

```
END;

/******************************** P_TARGET_WINS  ********************************/
P_TARGET_WINS = 91.84478
                                    + (0.03265 * TEAM_BATTING_H)
                                    + (-0.02673 * TEAM_BATTING_2B)
                                    + (0.09201 * TEAM_BATTING_3B)
                                    + (0.01751 * TEAM_BATTING_HR)
                                    + (0.05098 * TEAM_BATTING_BB)
                                    + (0.06068 * IMP_TEAM_BASERUN_SB)
                                    + (-0.0309 * IMP_TEAM_BASERUN_CS)
                                    + (-8.23266 * TEAM_PITCHING_H_LOG)
                                    + (-0.03332 * TEAM_PITCHING_BB_TRIM)
                                    + (-0.01368 * TEAM_PITCHING_SO_TRIM)
                                    + (-0.07595 * TEAM_FIELDING_E)
                                    + (-0.12137 * IMP3_TEAM_FIELDING_DP)
                                    + (0.01192 * TOTAL_BASES_GIVEN)
                                    + (7.88521 * m_TEAM_BATTING_SO)
                                    + (2.48453 * m_TEAM_FIELDING_DP)
                                    + (40.15057 * m_TEAM_BASERUN_SB);


If P_TARGET_WINS > 113 then P_TARGET_WINS = 113;
IF P_TARGET_WINS < 20 then P_TARGET_WINS = 20;
if missing(P_TARGET_WINS) then P_TARGET_WINS = 81;

run;

/******************************** Keep 2 Fields ********************************/
data &OUTFILE;
set &OUTFILE (keep=INDEX P_TARGET_WINS);
run;

%mend;



/******************************** Call the Macro ********************************/
%SCORE(mydata.moneyball_test, moneyball_score);
```