David Dietrich

PREDICT 411, Section 58; Professor Don Wedding

Assignment 1 (OLS): Moneyball

January 25, 2015

Bingo Bonus points attempted:

- (10 Points) Hand in your SCORED FILE as a SAS DATA SET
- (20 Points) Once you select a champion model in Step 4, use PROC GLM and PROC GENMOD to do the OLS Regression.

INTRODUCTION

The purpose of this assignment is to analyze baseball team data from 1871 to 2006 in order to predict the number of wins that a baseball team will have in a regular season of 162 games. This will be accomplished by exploring the dataset, understanding relationships within the data, analyzing the dataset, and identifying the best predictors of baseball team wins. These variables will be selected using stepwise and manual selection methods. Once identified, linear regression models will be built and the best model will be selected. The resulting regression equation will be used to score new datasets and predict wins for other baseball teams outside of the original dataset.

DATA EXPLORATION

Exploring the data is the first step in the process of analyzing the baseball dataset. This step allows the analyst to understand the distribution of the data variables, get a sense of the mean values and outliers, and look for correlations within the data. As a starting point, a number of diagnostic statistics were run on the data to examine the distribution of the data. The output is shown in Figure 1 below (the numbers have been rounded to two decimal places).

Variable	Label	N	N Miss	Median	Mean	Min.	Max.	1st Pctl	99th Pctl
TEAM_BASERUN_CS	Caught Stealing	1504	772	49.00	52.80	0	201.00	16.00	143.00
TEAM_BASERUN_SB	Stolen bases	2145	131	101.00	124.76	0	697.00	23.00	439.00
TEAM_BATTING_2B	Doubles	2276	0	238.00	241.25	69.00	458.00	141.00	352.00
TEAM_BATTING_3B	Triples	2276	0	47.00	55.25	0	223.00	17.00	134.00
TEAM_BATTING_BB	Walks	2276	0	512.00	501.56	0	878.00	79.00	755.00
TEAM_BATTING_H	Base Hits	2276	0	1454.00	1469.27	891.00	2554.00	1188.00	1950.00
TEAM_BATTING_HR	Homeruns	2276	0	102.00	99.61	0	264.00	4.00	235.00
TEAM_BATTING_SO	Strikeouts	2174	102	750.00	735.61	0	1399.00	67.00	1193.00
TEAM_FIELDING_E	Errors	2276	0	159.00	246.48	65.00	1898.00	86.00	1237.00
TEAM_PITCHING_BB	Walks allowed	2276	0	536.500	553.01	0	3645.00	237.00	924.00
TEAM_PITCHING_H	Hits allowed	2276	0	1518.00	1779.21	1137.00	30132.00	1244.00	7093.00
TEAM_PITCHING_HR	Homeruns Allowed	2276	0	107.00	105.70	0	343.00	8.00	244.00
TEAM_PITCHING_SO	Strikeouts by pitchers	2174	102	813.500	817.73	0	19278.00	205.00	1474.00
TEAM_BATTING_HBP	Batters hit by pitch	191	2085	58.00	59.36	29.00	95.00	29.00	90.00

Figure 1, Descriptive Statistics for Baseball ("Moneyball) Dataset

A few observations can be seen from Figure 1. Several variables have missing values, shown in the column labeled "N Miss", such as the data variables related being *caught stealing, stolen bases, fielding errors, strikeouts by pitchers*, and *batters hit by pitches*. The missing data for these variables will need to be addressed or imputed when it comes time to prepare the data for analysis. Because the variable for batters hit by pitches is missing in nearly 90% of the records, this variable (TEAM_BATTING_HBP) will be omitted from the analysis. It is worth noting that the minimum value for several variables is zero, as highlighted in the table in the "Min." column. Having variables with values of zero can reflect a true, accurate value of zero for that variable, but it can also indicate that there were missing values and these missing values were set to zero by whoever created the dataset. This is something to consider when analyzing the data, and also worth comparing to the "P1" column, showing the first percentile of the data, and zero is less than this in the values in the table. Additionally, there are multiple instances where the maximum value for a given variable exceeds the value for the 99th percent of that variable. Exceeding the 99th percent of the data is possible, though should be rare, however there are several instances in Figure 1 where the maximum value exceeds this by a large amount, such as the case for hits allowed (max 30132 compared to a P99 of 7093) and pitching strikeouts (max 19278 compared to a P99 of 1474). These differences between min and P1 and between max and P99 will need to be addressed in the data preparation phase.

After examining some of the basic statistics about the dataset, a matrix was produced to examine correlations between the variable and the target number of wins per team and with respect to the other variables in the dataset. Figure 2 shows a correlation matrix run on the data.

	INDEX	TARGET_WINS	TEAM_BATTING_H	TEAM_BATTING_2B	TEAM_BATTING_3B	TEAM_BATTING_HR	TEAM_BATTING_BB	TEAM_BATTING_SO
INDEX	1.00000 2276	-0.02106 0.3153 2276	-0.01792 0.3928 2276	0.01118 0.5939 2276	-0.00581 0.7816 2276	0.05148 0.0140 2276	-0.02657 0.2052 2276	0.08145 0.0001 2174
TARGET_WINS	-0.02106 0.3153 2276	1.00000 2276	0.38877 <.0001 2276	0.28910 <.0001 2276	0.14261 <.0001 2276	0.17615 <.0001 2276	0.23256 <.0001 2276	-0.03175 0.1389 2174
TEAM_BATTING_H Base Hits by batters	-0.01792 0.3928 2276	0.38877 <.0001 2276	1.00000 2276	0.56285 <.0001 2276	0.42770 <.0001 2276	-0.00654 0.7550 2276	-0.07246 0.0005 2276	-0.46385 <.0001 2174
TEAM_BATTING_2B Doubles by batters	0.01118 0.5939 2276	0.28910 <.0001 2276	0.56285 <.0001 2276	1.00000 2276	-0.10731 <.0001 2276	0.43540 <.0001 2276	0.25573 <.0001 2276	0.16269 <.0001 2174
TEAM_BATTING_3B Triples by batters	-0.00581 0.7816 2276	0.14261 <.0001 2276	0.42770 <.0001 2276	-0.10731 <.0001 2276	1.00000 2276	-0.63557 <.0001 2276	-0.28724 <.0001 2276	-0.66978 <.0001 2174
TEAM_BATTING_HR Homeruns by batters	0.05148 0.0140 2276	0.17615 <.0001 2276	-0.00654 0.7550 2276	0.43540 <.0001 2276	-0.63557 <.0001 2276	1.00000 2276	0.51373 <.0001 2276	0.72707 <.0001 2174
TEAM_BATTING_BB Walks by batters	-0.02657 0.2052 2276	0.23256 <.0001 2276	-0.07246 0.0005 2276	0.25573 <.0001 2276	-0.28724 <.0001 2276	0.51373 <.0001 2276	1.00000 2276	0.37975 <.0001 2174
TEAM_BATTING_SO Strikeouts by batters	0.08145 0.0001 2174	-0.03175 0.1389 2174	-0.46385 <.0001 2174	0.16269 <.0001 2174	-0.66978 <.0001 2174	0.72707 <.0001 2174	0.37975 <.0001 2174	1.00000 2174
TEAM_BASERUN_SB Stolen bases	0.04027 0.0622 2145	0.13514 <.0001 2145	0.12357 <.0001 2145	-0.19976 <.0001 2145	0.53351 <.0001 2145	-0.45358 <.0001 2145	-0.10512 <.0001 2145	-0.25449 <.0001 2043
TEAM_BASERUN_CS Caught stealing	0.00057 0.9825 1504	0.02240 0.3853 1504	0.01671 0.5174 1504	-0.09981 0.0001 1504	0.34876 <.0001 1504	-0.43379 <.0001 1504	-0.13699 <.0001 1504	-0.21788 <.0001 1504
TEAM_BATTING_HBP Batters hit by pitch	0.07719 0.2885 191	0.07350 0.3122 191	-0.02911 0.6893 191	0.04608 0.5267 191	-0.17425 0.0159 191	0.10618 0.1438 191	0.04746 0.5144 191	0.22094 0.0021 191
TEAM_PITCHING_H Hits allowed	0.01710 0.4148 2276	-0.10994 <.0001 2276	0.30269 <.0001 2276	0.02369 0.2585 2276	0.19488 <.0001 2276	-0.25015 <.0001 2276	-0.44978 <.0001 2276	-0.37569 <.0001 2174
TEAM_PITCHING_HR Homeruns allowed	0.05099 0.0150 2276	0.18901 <.0001 2276	0.07285 0.0005 2276	0.45455 <.0001 2276	-0.56784 <.0001 2276	0.96937 <.0001 2276	0.45955 <.0001 2276	0.66718 <.0001 2174
TEAM_PITCHING_BB Walks allowed	-0.01529 0.4660 2276	0.12417 <.0001 2276	0.09419 <.0001 2276	0.17805 <.0001 2276	-0.00222 0.9155 2276	0.13693 <.0001 2276	0.48936 <.0001 2276	0.03701 0.0845 2174
TEAM_PITCHING_SO Strikeouts by pitchers	0.05589 0.0091 2174	-0.07844 0.0003 2174	-0.25266 <.0001 2174	0.06479 0.0025 2174	-0.25882 <.0001 2174	0.18471 <.0001 2174	-0.02076 0.3334 2174	0.41623 <.0001 2174
TEAM_FIELDING_E Errors	-0.00923 0.6596 2276	-0.17648 <.0001 2276	0.26490 <.0001 2276	-0.23515 <.0001 2276	0.50978 <.0001 2276	-0.58734 <.0001 2276	-0.65597 <.0001 2276	-0.58466 <.0001 2174
TEAM_FIELDING_DP Double Plays	0.02006 0.3710 1990	-0.03485 0.1201 1990	0.15536 <.0001 1990	0.29088 <.0001 1990	-0.32307 <.0001 1990	0.44899 <.0001 1990	0.43088 <.0001 1990	0.15489 <.0001 1888

Figure 2, Correlation Matrix for All Variables

Figure 2 shows a correlation matrix between the variables. Although useful for comparing variables, it is difficult to read with many variables. Figure 3 shows a subset of the variables, namely the correlations between the independent variables and the dependent variable that will be predicted by the model, "TARGET_WINS".

Figure 3 shows that most variables are only weakly correlated with the variable TARGET_WINS. Of these, the variables with the highest correlations are base hits by batters (0.38877), doubles by batters (0.28910) and walks by batters (0.23256). These correlations are not very high values, suggesting that these variables may not have very strong predictive power when a linear regression model is created, and together these variables may not account for most of the variation in the data for predicting baseball team wins.

In addition to the correlations with TARGET_WINS, several other relationships within the data were observed to have higher correlations. *Pitching homeruns allowed* was highly correlated (0.96937) with *team batting homeruns*. This makes sense as a team that gives up many homeruns in pitching requires another team to hit the homeruns. Team batting homeruns was also highly correlated with team strikeouts (0.72707). This also seems logical as homerun hitters typically strikeout more often than baseball players who are trying to have a high batting average or just get on base. Lastly, team pitching home runs allowed also was highly correlated with team batting doubles (0.45455) and negatively correlated with team batting triples (-0.56784). It makes sense that players who hit for power and can hit homeruns can also h

	INDEX	TARGET_WINS
INDEX	1.00000 2276	-0.02106 0.3153 2276
TARGET_WINS	-0.02106 0.3153 2276	1.00000 2276
TEAM_BATTING_H Base Hits by batters	-0.01792 0.3928 2276	0.38877 <.0001 2276
TEAM_BATTING_2B Doubles by batters	0.01118 0.5939 2276	0.28910 <.0001 2276
TEAM_BATTING_3B Triples by batters	-0.00581 0.7816 2276	0.14261 <.0001 2276
TEAM_BATTING_HR Homeruns by batters	0.05148 0.0140 2276	0.17615 <.0001 2276
TEAM_BATTING_BB Walks by batters	-0.02657 0.2052 2276	0.23256 <.0001 2276
TEAM_BATTING_SO Strikeouts by batters	0.06145 0.0001 2174	-0.03175 0.1389 2174
TEAM_BASERUN_SB Stolen bases	0.04027 0.0622 2145	0.13514 <.0001 2145
TEAM_BASERUN_CS Caught stealing	0.00057 0.9825 1504	0.02240 0.3853 1504
TEAM_BATTING_HBP Batters hit by pitch	0.07719 0.2885 191	0.07350 0.3122 191
TEAM_PITCHING_H Hits allowed	0.01710 0.4148 2276	-0.10994 <.0001 2276
TEAM_PITCHING_HR Homeruns allowed	0.05099 0.0150 2276	0.18901 <.0001 2276
TEAM_PITCHING_BB Walks allowed	-0.01529 0.4660 2276	0.12417 <.0001 2276
TEAM_PITCHING_SO Strikeouts by pitchers	0.05589 0.0091 2174	-0.07844 0.0003 2174
TEAM_FIELDING_E Errors	-0.00923 0.6598 2276	-0.17648 <.0001 2276
TEAM_FIELDING_DP Double Plays	0.02006 0.3710 1990	-0.03485 0.1201 1990

Figure 3, Correlation Matrix for TARGET_WINS

It seems counterintuitive that players who hit home runs may hit fewer triples, although this may actually make sense. Triples are generally considered the hardest to achieve as these kinds of hits require a player to have a balance of batting power coupled with fast running speed. It may be the case that those batters who hit for power are larger and slower than smaller players who can hit for triples.

The next portion of the data exploration phase involved looking at histograms of the data to explore the distributions of the variables. After creating histograms for the variables, several data variables appeared to have non-normal distributions, including stolen bases, batting home runs, and pitching hits allowed. These non-normal distributions can cause a problem for a linear regression model, since one of the modeling assumptions is that the data is normally distributed. Shown below is figure 4 with histograms for team stolen bases and batting home runs.

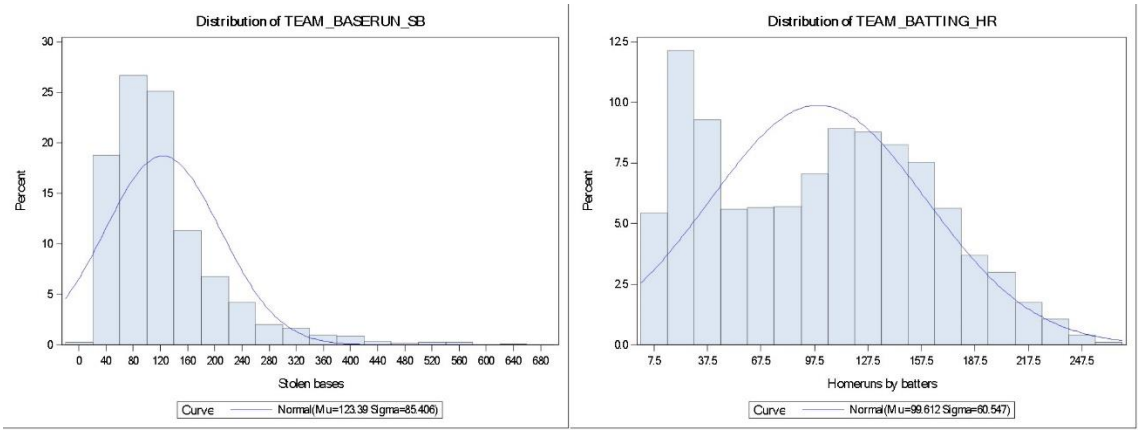

Figure 4, Histograms of Stolen Bases and Batting Home Runs

The histogram for stolen bases shows the distribution is skewed to the right, with most of the data occurring toward the left hand side of the graph. The histogram for team batting home runs reflects a more consistent distribution, but still one with a lot of homeruns at the low end (left hand side) of the distribution. These skewed distributions will need to be addressed during the data preparation phase, before the regression models are created and run.

DATA PREPARATION

After exploring the data, several problems with the data emerged and were addressed during the data preparation. First, because there were a significant number of missing values in the data, these needed to be filled in or removed before the linear regression models could be built. Because the variable for batters hit by pitches contained over 90% missing values, it did not make sense to impute values for this, since there would be more imputed values than original values. Therefore, TEAM_BATTING_HBP was removed from the analysis during the data preparation phase. In addition, a rule was applied such as that for the variables *caught stealing, stolen bases, errors,* and *strikeouts by pitchers*, missing values were replaced with the median value for that variable. During this step of imputing missing values, additional 'flag fields' were created with values of 0 and 1 to indicate which values of the original columns had missing data. In these new flag fields, a value of 1 indicates that the field originally had a missing value. Creating these flag fields can be useful during the analysis if one wants to see if missing values are predictive of target wins.

Although these steps addressed the missing values, there were still instances where data contained outliers that needed to be addressed. For instance, *hits allowed* (TEAM_PITCHING_HITS) had a maximum value of 30132, even though the value for that variable at the 99th percentile was significantly less, at 7093. Therefore, in this case and several others, the data was capped such that values that were less than the 1st percentile value were capped at the 1st percentile value, and values greater than the 99th percentile value were capped at the 99th percentile value. This step helped to ensure that the data was reasonable and reduced the number of outliers in the dataset in order to help with the model predictions during the analysis. The following variables were capped in this manner:

team pitching hits allowed, team pitching strikeouts, team pitching walks, team batting triples, team batting doubles, team batting total hits, team batting home runs, and *team fielding double plays*.

For batting statistics, although the data contained total hits, doubles, triples, and home runs, it did not contain a variable for singles. Since total hits contained singles, doubles, triples and home runs, singles could be derived by subtracting doubles, triples, and home runs from total hits (TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR). Once the value of singles were derived, the outliers were also capped at the 1st and 99th percentile values. (These capped values in the model contain the prefix "CAP_" followed by the original variable name.)

After doing research on the internet and reviewing the book "Moneyball", by Michael Lewis, there were two other derived variables that were suggested to include in the analysis, on-base-percentage (OBP) and slugging percentage. OBP is usually defined by the following formula:

OBP = (Hits + Walks + Hit by Pitch) / (At Bats + Walks + Hit by Pitch + Sacrifice Flies) .

Unfortunately, this dataset lacked important components of this formula, such as *At Bats* and *Sacrifice Flies,* and the data for *Hit by Pitch* contained mostly missing values. Therefore, the OBP variable that was derived for the model is simplified as the formula: OBP = (Hits + Walks). This formula was not nearly as comprehensive as the generally accepted OBP formula, but it was still worth including to see if it would be useful.

Now that hitting singles had been derived, the variable *slugging percentage* could be derived. Slugging percentage is usually defined as:

Slugging = ((Singles) + (2*Doubles) + (3*Triples) + (4*Home Runs))/At Bats.

Since this baseball dataset lacked *At Bats*, slugging was defined as the numerator only for this dataset:

Slugging = (singles) + (2*doubles) + (3*triples) + (4*home runs) .

In addition, another variable was created called *slugging ratio* as follows:

Slugging Ratio = ((singles) + (2*doubles) + (3*triples) + (4*home runs)) / Total Hits .

As noted in the Data Exploration section, there were several variables that had non-normal data distributions. This poses a potential problem for linear regression models (ordinary least squares in this case, OLS), as OLS models assume that the data is normally distributed. To address this issue, several of the variables with skewed data distributions were transformed by taking the base 10 logarithm of the original value. This has the benefit of changing the distribution of the data, and in many cases making it into a more normal distribution. Figure 5 shows a histogram showing the distribution of the data for stolen bases (TEAM_BASERUN_SB). As one can observe, the data is skewed to the right, with most of

the values on the left hand side of the graph and values trailing off as the number of stolen bases increases toward the right side of the x-axis.

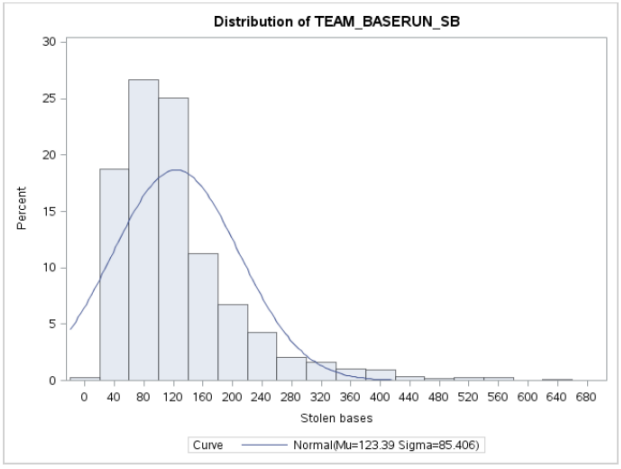

Figure 5, Histogram of Stolen Bases

Figure 6 shows a similar histogram on the stolen base variable once it has been transformed with base10 logarithms. As one can observe, the distribution in Figure 6 shows a much more normal distribution of the data for *stolen bases*.

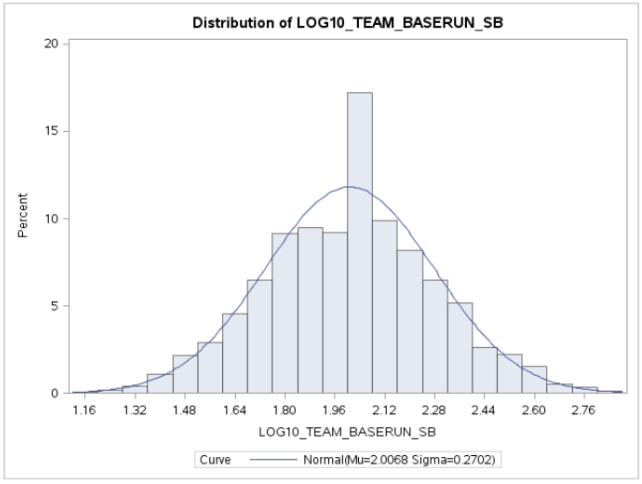

Figure 6, Histogram Showing Log10 Transformation of Stolen Bases Variable

In addition to *stolen bases*, the base10 logarithm was applied to transform the variables for fielding errors, batting home runs, and pitching hits allowed.

MODEL BUILDING

After performing the data preparation step, three regression models were built. The first one attempted contained a set of manually derived variables as shown below:

Model 1

WINS =		18.16749
+ Caught Stealing	*	-0.04068
+ Walks by batters	*	0.06146
+ Team Batting Strikeouts	*	-0.04044
+ Fielding Double Plays	*	-0.10987
+ Errors	*	-0.02120
+ Pitching walks	*	-0.03132
+ Pitching hits allowed	*	0.00344
+ Homeruns allowed	*	-0.07458
+ Pitching Strikeouts	*	0.02729
+ On Base Percentage	*	-0.01055
+ Slugging	*	0.04991
+ Slugging Ratio	*	-0.05295
+ Stolen Bases (Log10)	*	13.22081
+ Errors (Log10)	*	-7.52359
+ Batting Home Runs (Log10)	*	-2.32941
+ Pitching Hits Allowed (Log10)	*	2.54488

Usually, having more variables, up to a point, will give the model more predictive power and increase its ability to predict the outcome variable, which is Target Wins, in this case. However, despite having 16 variables, **Model 1** was not a very good model. To begin with, its adjusted R squared value was low (only 0.2937) indicating that this set of variables only explains about 29% of the variation in the data. The adjusted R square reflects a correlation of the variables with Target Wins, and reduces the adjusted R square for each additional variable in the model. In other words, this reflects how good a model is with the fewest variables possible, thus balancing correlation with parsimony.

Several observations can be made about the variables chosen for Model 1. Most of the variables make intuitive sense, although some do not. For example, one would expect that C*aught Stealing, Batting Strikeouts, Errors* and P*itching Walks* would be correlated with fewer wins, which they are. In addition,

one would expect that *Walks by Batters*, *Pitching Strikeouts*, and *Slugging* would be correlated to higher wins, and they are as well. However, some other variables raise more questions. One would expect that *On Base Percentage* and *Batting Home Runs* would predict wins well, although these are negatively correlated with wins in this model. Similarly, *Slugging Ratio* is negatively correlated with wins, yet one would expect Slugging Ratio to predict wins positively. Lastly, two of the variables in this model were shown not to be significant predictors of wins, Log10 of Team Batting Home Runs and Log10 of Team Pitching Hits. As one would expect these two variables to be predictive of wins, something seems amiss with the variables chosen. These things taken together would indicate that Model 1 is not a very good model and can be discarded, or at minimum needs refinement.

Model 2

As a next step, a second model ("Model 2") was created. The thinking behind Model 2 was to add more variables in the model and employ an automated variable selection technique, called Stepwise. Stepwise selection iteratively adds a new variable into the model, confirms it is significant and should remain, and tests to see if any of the variables should be dropped from the model in each successive iteration. At the end of the process, Stepwise will have chosen a set of variables that were deemed significant for the model. The output of this process for Model 2 is shown below.

WINS =		373.52100
+ Flag Batting Strikeouts	*	7.86250
+ Stolen Bases	*	0.06348
+ Flag Stolen Bases	*	38.39007
+ Caught Stealing	*	-0.03011
+ Flag Caught Stealing	*	3.28186
+ Pitching Strikeouts	*	0.00253
+ Flag Fielding Double Plays	*	5.44840
+ Pitching Hits Allowed	*	0.01379
+ Homeruns allowed	*	0.12447
+ Pitching Strikeouts Allowed	*	-0.01481
+ Triples	*	0.19252
+ On Base Percentage	*	0.07635
+ Walks by batters	*	-0.04778
+ Slugging	*	-0.02395
+ Fielding Double Plays	*	-0.09757
+ Log10 Fielding Errors	*	-49.38483

+ Log10 Pitching Hits Allowed * -91.35565

This model contains 17 variables and has a substantially higher adjusted R square value of 0.4143 . This suggests that it is a superior model to Model 1. One of the interesting things about this model is how influential the derived data fields are in this model. The flag fields for *Batting Strikeouts, Stolen Bases, Caught Stealing,* and *Fielding Double Plays* were all selected by the Stepwise model and have high coefficient values, indicating they have substantial influence in predicting wins. This suggests that it was worthwhile to create the flag fields in the data preparation phase of the analysis process. In addition, the variables that were transformed by logarithms (Log10) for *Fielding Errors* and *Pitching Hits Allowed* also have high coefficient values, signaling that these are predictive of wins as well.

Although Model 2 is better than Model 1, Model 2 still contains some problems. One of its issues is that it suffers from variance inflation, meaning that some of the variables are useful for predicting wins, but may be misleading because they may actually be more highly correlated with other variables than with predicting wins. Variables with a variance inflation factor ("VIF") over 10 are cause for concern, and those with VIF over 30 should be removed. Here are some of the variables with relatively high VIF's in Model 2: Team Pitching Hits Allowed (34), Homeruns allowed (20), On Base Percentage (42), Walks by batters (22), and Slugging (40). This issue will be addressed in Model 3.

Model 3

The approach for Model 3 was to use most of the variables identified through the Stepwise selection process in Model 2, while dealing with the problem of correlated variables mentioned above. To improve on Model 2, a correlation matrix was run with only the correlated variables in question. The result of this is shown below.

Pearson Correlation Coefficients, N = 2276 Prob > |r| under H0: Rho=0		
	TEAM_PITCHING_HR	**CAP_TEAM_PITCHING_H**
TEAM_PITCHING_HR Homeruns allowed	1.00000	-0.18667 <.0001
CAP_TEAM_PITCHING_H	-0.18667 <.0001	1.00000
OBP_TEAM_BATTING	0.37585 <.0001	-0.06266 0.0028
TEAM_BATTING_BB Walks by batters	0.45955 <.0001	-0.55993 <.0001
TEAM_BATTING_SLG	0.70344 <.0001	0.05403 0.0099

Figure 7: Correlation Matrix for Model 3 Candidate Variables

Pearson Correlation Coefficients, N = 2276 Prob > |r| under H0: Rho=0			
	OBP_TEAM_BATTING	**TEAM_BATTING_BB**	**TEAM_BATTING_SLG**
TEAM_PITCHING_HR Homeruns allowed	0.37585 <.0001	0.45955 <.0001	0.70344 <.0001
CAP_TEAM_PITCHING_H	-0.06266 0.0028	-0.55993 <.0001	0.05403 0.0099
OBP_TEAM_BATTING	1.00000	0.66142 <.0001	0.77693 <.0001
TEAM_BATTING_BB Walks by batters	0.66142 <.0001	1.00000	0.34031 <.0001
TEAM_BATTING_SLG	0.77693 <.0001	0.34031 <.0001	1.00000

Figure 8: Correlation Matrix for Model 3 Candidate Variables

Figures 7 and 8 show correlations among several of the variables that are desirable to be used in the model, along with correlation values among the variables. Correlation values closer to 1 indicate high correlations, while those close to 0 indicate low correlation. The intent of this matrix is to identify variables that are highly correlated to each other, and see if one can be safely removed so the other variables can be retained in Model 3. In this case, Walks (TEAM_BATTING_BB) was identified as this variable. Walks contributed little to the improvement in the adjusted R square value, and also was correlated with *Pitching Homeruns Allowed, Team Pitching Hits Allowed, On Base Percentage,* and *Slugging*. Once *Walks* was removed from the model, the variance inflation factors improved for the variables remaining in the model, and the removal of *Walks* did not adversely affect the adjusted R squared value in a substantial way. Following the correlation matrix, Model 3 was run, which produced the variables below.

WINS =		232.99000
+ Triples (Capped)	*	0.15827
+ Fielding double plays (Capped)	*	-0.09704
+ Pitching Hits Allowed (Capped)	*	0.00902
+ Pitching Strikeouts (Capped)	*	-0.01313
+ Stolen Bases	*	0.05885
+ Fielding errors (Log10)	*	-48.41700
+ Pitching hits allowed (Log10)	*	-40.20204
+ Caught stealing	*	3.56888

+ Flag for missing values for stolen bases	*	34.89288
+Flag for missing values for strike outs	*	9.00281
+ Flag for missing values for double plays	*	5.90136
+ On base percentage	*	0.03326
+ Homeruns allowed	*	0.06249

MODEL SELECTION

Model 3 was the best model of the three models that were tried. It had an adjusted R squared value of 0.4128 based on 16 variables, meaning it had similar predictive power as Model 2, but using fewer variables, which is preferable. For the most part, the variables make sense, although there are a few that raise concerns and seem counterintuitive. For instance, *Caught Stealing* showed a positive relationship to wins, which seems a bit strange. That is, being caught stealing should not result in more wins, it should result in more outs, which would contribute to more losses. This suggests either further, hidden correlations or the need for more domain knowledge of baseball to explain this dynamic. Similarly, it was unexpected that *Fielding Double Plays* and *Pitching Strikeouts* both are somewhat negatively correlated to wins. It was expected that both of these would positively influence wins. Lastly, the flag field for *Missing Values of Strikeouts* and *Homeruns Allowed* both are positively correlated with wins, which seems somewhat counterintuitive. This relationship is more nuanced, as previous models tried showed that strikeouts and homeruns are correlated with each other. That is, players who hit a lot of home runs also tend to strikeout more often than those who are hitting for high batting average, so high batting strikeouts could reveal a latent relationship for more homerun hitters in the baseball roster.

The diagnostic plots of the model seemed reasonable. Since the data had been prepared to handle outliers and missing values, this was not an issue for the models. Even though there were not serious outliers, there were several values that will have influence on the model, as highlighted in Figure 9 in red.

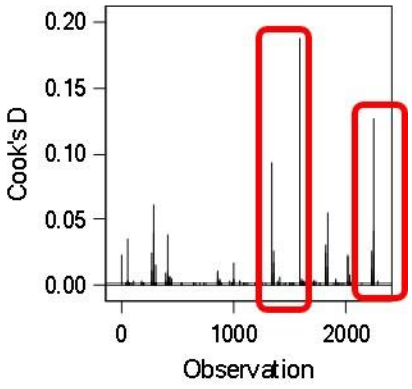

Figure 9: Cook's Distance Plot to Identify Outliers for Model 3

The points highlighted in Figure 9 are higher than the others and they are worth watching, but do not pose problems for the model. In general, Cook's D values greater than 1 pose issues, and these values are far below that, meaning that they influence the model, but may not be strictly considered outliers and thus they are not problematic in this situation.

CONCLUSION

Several models were developed to predict the number of wins for baseball teams, containing data from 1871 to 2006. Within that large timespan, the way baseball was played and officiated changed, thus causing changes in the underlying data over time in the way data was collected and also the duration of the baseball seasons. These changes caused the data to vary over time, meaning one had to account for these changes in the data and adjust for missing and derived values. The best linear model for predicting wins was derived using stepwise approach to variable selection. Once this was performed, variables were removed that were highly correlated or otherwise problematic, such that a final model was built with a subset of the variables selected during the stepwise selection process. Although the winning model performed reasonably well, there were some counterintuitive outcomes with regard to the signs of several variables, where several variables had unexpected positive or negative influence on predicting wins, where these variables were expected to have the opposite effect. This issue requires further investigation and research, which is beyond the scope of this document.

CODE

```
/*
 * David Dietrich
 * Created: January 25, 2015
 * Unit 01: Linear Regression. HW 01 Baseball
 */

libname mydata
'/home/daviddietrich2013/my_courses/donald.wedding/c_8888/PRED411/UNIT01/HW'
access=readonly;

*proc contents data=mydata.moneyball;
*run;

*proc print data=new_moneyball(obs=25);
```

```
data new_moneyball;
set mydata.moneyball;

/*Add Flag fields for missing values and impute missing values based on Median */
	IMP_TEAM_BATTING_SO = TEAM_BATTING_SO;
	m_TEAM_BATTING_SO = 0;
	if missing(IMP_TEAM_BATTING_SO) then do;
		IMP_TEAM_BATTING_SO = 750;
		m_TEAM_BATTING_SO = 1;
	end;

	IMP_TEAM_BASERUN_SB = TEAM_BASERUN_SB;
	m_TEAM_BASERUN_SB = 0;
	if missing(IMP_TEAM_BASERUN_SB) then do;
		IMP_TEAM_BASERUN_SB = 101;
		m_TEAM_BASERUN_SB = 1;
	end;

	IMP_TEAM_BASERUN_CS = TEAM_BASERUN_CS;
	m_TEAM_BASERUN_CS = 0;
	if missing(IMP_TEAM_BASERUN_CS) then do;
		IMP_TEAM_BASERUN_CS = 49;
		m_TEAM_BASERUN_CS = 1;
	end;

	IMP_TEAM_PITCHING_SO = TEAM_PITCHING_SO;
	m_TEAM_PITCHING_SO = 0;
```

```
	if missing(IMP_TEAM_PITCHING_SO) then do;
		IMP_TEAM_PITCHING_SO = 813;
		m_TEAM_PITCHING_SO = 1;
	end;

	IMP_TEAM_FIELDING_DP = TEAM_FIELDING_DP;
	m_TEAM_FIELDING_DP = 0;
	if missing(IMP_TEAM_FIELDING_DP) then do;
		IMP_TEAM_FIELDING_DP = 149;
		m_TEAM_FIELDING_DP = 1;
	end;

	IMP_TEAM_BATTING_HBP = TEAM_BATTING_HBP;
	m_TEAM_BATTING_HBP = 0;
	if missing(IMP_TEAM_BATTING_HBP) then do;
		m_TEAM_FIELDING_DP = 1;
	end;

/*Transform data to reduce problems from outliers*/

	CAP_TEAM_PITCHING_H = TEAM_PITCHING_H;
	if CAP_TEAM_PITCHING_H < 1244 then CAP_TEAM_PITCHING_H = 1244;
	if CAP_TEAM_PITCHING_H > 7000 then CAP_TEAM_PITCHING_H = 7000;

	CAP_TEAM_PITCHING_SO = IMP_TEAM_PITCHING_SO;
	if CAP_TEAM_PITCHING_SO < 208 then CAP_TEAM_PITCHING_SO = 208;
	if CAP_TEAM_PITCHING_SO > 1464 then CAP_TEAM_PITCHING_SO = 1464;
```

```
CAP_TEAM_PITCHING_BB = TEAM_PITCHING_BB;
if CAP_TEAM_PITCHING_BB < 237 then CAP_TEAM_PITCHING_BB = 237 ;
if CAP_TEAM_PITCHING_BB > 924 then CAP_TEAM_PITCHING_BB = 924;

CAP_TEAM_BATTING_3B = TEAM_BATTING_3B;
if CAP_TEAM_BATTING_3B < 17 then CAP_TEAM_BATTING_3B = 17;
if CAP_TEAM_BATTING_3B > 134 then CAP_TEAM_BATTING_3B = 134;

CAP_TEAM_BATTING_2B = TEAM_BATTING_2B;
if CAP_TEAM_BATTING_2B < 141 then CAP_TEAM_BATTING_2B = 141;
if CAP_TEAM_BATTING_2B > 352 then CAP_TEAM_BATTING_2B = 352;

CAP_TEAM_BATTING_H = TEAM_BATTING_H;
if CAP_TEAM_BATTING_H < 1188 then CAP_TEAM_BATTING_H = 1188;
if CAP_TEAM_BATTING_H > 1950 then CAP_TEAM_BATTING_H = 1950;

CAP_TEAM_BATTING_HR = TEAM_BATTING_HR;
if CAP_TEAM_BATTING_HR < 4 then CAP_TEAM_BATTING_HR = 4;
if CAP_TEAM_BATTING_HR > 235 then CAP_TEAM_BATTING_HR = 235;

OBP_TEAM_BATTING = TEAM_BATTING_BB + CAP_TEAM_BATTING_H ;
TEAM_BATTING_1B = CAP_TEAM_BATTING_H - CAP_TEAM_BATTING_2B - CAP_TEAM_BATTING_3B -
CAP_TEAM_BATTING_HR ;

CAP_TEAM_BATTING_1B = TEAM_BATTING_1B;
if CAP_TEAM_BATTING_1B < 882 then CAP_TEAM_BATTING_1B = 882;
if CAP_TEAM_BATTING_1B > 1486 then CAP_TEAM_BATTING_1B = 1486;
```

```
	TEAM_BATTING_SLG = CAP_TEAM_BATTING_1B + (2*CAP_TEAM_BATTING_2B) +
(3*CAP_TEAM_BATTING_3B) + (4*CAP_TEAM_BATTING_HR);

	TEAM_BATTING_SLG_RATIO =  (CAP_TEAM_BATTING_2B) + (CAP_TEAM_BATTING_3B) +
(CAP_TEAM_BATTING_HR)/CAP_TEAM_BATTING_H;


	CAP_TEAM_FIELDING_DP = IMP_TEAM_FIELDING_DP;

	if CAP_TEAM_FIELDING_DP < 80 then CAP_TEAM_FIELDING_DP = 80;

	if CAP_TEAM_FIELDING_DP  > 202 then CAP_TEAM_FIELDING_DP = 202;


	LOG10_TEAM_BASERUN_SB = LOG10(IMP_TEAM_BASERUN_SB);

	LOG10_TEAM_FIELDING_E = LOG10(TEAM_FIELDING_E);

	LOG10_CAP_TEAM_BATTING_HR = LOG10(CAP_TEAM_BATTING_HR);

	LOG10_TEAM_PITCHING_H = LOG10(TEAM_PITCHING_H);


drop TEAM_BATTING_HBP;

drop IMP_TEAM_BATTING_HBP;

drop INDEX;

run;


/*

PROC UNIVARIATE data=new_moneyball;

	HISTOGRAM / normal;

	Title;


*/


/*

proc means data=new_moneyball n nmiss median mean MIN MAX p1 p99;

   var TEAM_BASERUN_CS
```

TEAM_BASERUN_SB

TEAM_BATTING_1B

TEAM_BATTING_2B

TEAM_BATTING_3B

TEAM_BATTING_BB

TEAM_BATTING_H

TEAM_BATTING_HR

TEAM_BATTING_SO

CAP_TEAM_FIELDING_DP

TEAM_FIELDING_E

TEAM_PITCHING_BB

CAP_TEAM_PITCHING_BB

TEAM_PITCHING_H

CAP_TEAM_PITCHING_H

TEAM_PITCHING_HR

CAP_TEAM_PITCHING_SO

TEAM_PITCHING_SO

CAP_TEAM_BATTING_3B

CAP_TEAM_BATTING_2B

CAP_TEAM_BATTING_H

CAP_TEAM_BATTING_HR

OBP_TEAM_BATTING

TEAM_BATTING_1B

CAP_TEAM_BATTING_1B

TEAM_BATTING_SLG

TEAM_BATTING_SLG_RATIO

LOG10_TEAM_BASERUN_SB

LOG10_TEAM_FIELDING_E

```
                LOG10_CAP_TEAM_BATTING_HR
                LOG10_TEAM_PITCHING_H
;
RUN;
quit;

*/

/*
Title “Computing Pearson Correlation Coefficients on New_Moneyball Data”
 ods graphics on;
proc corr data=new_moneyball plot=matrix(nvar=10);
run;
ods graphics off;

quit;
*/

/* MODEL 1 - OLS Model with Derived vars
proc reg data = new_moneyball;
model TARGET_WINS =
                IMP_TEAM_BASERUN_CS
                TEAM_BATTING_BB
                IMP_TEAM_BATTING_SO
                CAP_TEAM_FIELDING_DP
                TEAM_FIELDING_E
                CAP_TEAM_PITCHING_BB
                CAP_TEAM_PITCHING_H
```

```
                TEAM_PITCHING_HR
                CAP_TEAM_PITCHING_SO
                OBP_TEAM_BATTING
                TEAM_BATTING_SLG
                TEAM_BATTING_SLG_RATIO
                LOG10_TEAM_BASERUN_SB
                LOG10_TEAM_FIELDING_E
                LOG10_CAP_TEAM_BATTING_HR
                LOG10_TEAM_PITCHING_H
                ;
RUN;

*/


/* MODEL 2 - OLS Model with Stepwise selection

proc reg data = new_moneyball;
        model TARGET_WINS =
                IMP_TEAM_BATTING_SO
                m_TEAM_BATTING_SO
                IMP_TEAM_BASERUN_SB
                m_TEAM_BASERUN_SB
                IMP_TEAM_BASERUN_CS
                m_TEAM_BASERUN_CS
                IMP_TEAM_PITCHING_SO
                m_TEAM_PITCHING_SO
                m_TEAM_FIELDING_DP
```

```
                m_TEAM_BATTING_HBP
                CAP_TEAM_PITCHING_H
                TEAM_PITCHING_HR
                CAP_TEAM_PITCHING_SO
                CAP_TEAM_PITCHING_BB
                CAP_TEAM_BATTING_3B
                CAP_TEAM_BATTING_2B
                CAP_TEAM_BATTING_H
                CAP_TEAM_BATTING_HR
                OBP_TEAM_BATTING
                CAP_TEAM_BATTING_1B
                TEAM_BATTING_BB
                TEAM_BATTING_SLG
                TEAM_BATTING_SLG_RATIO
                CAP_TEAM_FIELDING_DP
                LOG10_TEAM_BASERUN_SB
                LOG10_TEAM_FIELDING_E
                LOG10_CAP_TEAM_BATTING_HR
                LOG10_TEAM_PITCHING_H
                /SELECTION = STEPWISE VIF;
RUN;

*/

/*
Title “Computing Pearson Correlation Coefficients on New_Moneyball Data”
 ods graphics on;
proc corr data=new_moneyball plot=matrix(nvar=10);
```

```
        VAR TEAM_PITCHING_HR
        CAP_TEAM_PITCHING_H
        OBP_TEAM_BATTING
        TEAM_BATTING_BB
        TEAM_BATTING_SLG ;
run;
ods graphics off;


quit;


*/



/*

MODEL 2(b) - Running OLS with only significant vars with Stepwise to verify VIF
                        Removed Var TEAM_BATTING_BB, which had high VIF and low partial R Sq


proc reg data = new_moneyball;
model TARGET_WINS =
    CAP_TEAM_BATTING_3B
            CAP_TEAM_FIELDING_DP
            CAP_TEAM_PITCHING_H
            CAP_TEAM_PITCHING_SO
            IMP_TEAM_BASERUN_CS
            IMP_TEAM_BASERUN_SB
            IMP_TEAM_PITCHING_SO
            LOG10_TEAM_FIELDING_E
```

```
                LOG10_TEAM_PITCHING_H
                m_TEAM_BASERUN_CS
                m_TEAM_BASERUN_SB
                m_TEAM_BATTING_SO
                m_TEAM_FIELDING_DP
                OBP_TEAM_BATTING
                TEAM_BATTING_SLG
                TEAM_PITCHING_HR
                /SELECTION = STEPWISE VIF;
RUN;
*/


/*

MODEL 3 - Running OLS with only significant vars
proc reg data = new_moneyball;
model TARGET_WINS =
                CAP_TEAM_BATTING_3B
                CAP_TEAM_FIELDING_DP
                CAP_TEAM_PITCHING_H
                CAP_TEAM_PITCHING_SO
                IMP_TEAM_BASERUN_SB
                LOG10_TEAM_FIELDING_E
                LOG10_TEAM_PITCHING_H
                m_TEAM_BASERUN_CS
                m_TEAM_BASERUN_SB
                m_TEAM_BATTING_SO
```

```
                m_TEAM_FIELDING_DP
                OBP_TEAM_BATTING
                TEAM_PITCHING_HR
                ;
RUN;

*/

/* Running Model 3 OLS with PROC GLM

proc glm data=new_moneyball;
model TARGET_WINS =
                CAP_TEAM_BATTING_3B
                CAP_TEAM_FIELDING_DP
                CAP_TEAM_PITCHING_H
                CAP_TEAM_PITCHING_SO
                IMP_TEAM_BASERUN_SB
                LOG10_TEAM_FIELDING_E
                LOG10_TEAM_PITCHING_H
                m_TEAM_BASERUN_CS
                m_TEAM_BASERUN_SB
                m_TEAM_BATTING_SO
                m_TEAM_FIELDING_DP
                OBP_TEAM_BATTING
                TEAM_PITCHING_HR
                ;
RUN;
```

```
quit;
*/

/* Running Running Model 3 OLS with PROC GENMOD
proc genmod data=new_moneyball;
model TARGET_WINS =
                    CAP_TEAM_BATTING_3B
                    CAP_TEAM_FIELDING_DP
                    CAP_TEAM_PITCHING_H
                    CAP_TEAM_PITCHING_SO
                    IMP_TEAM_BASERUN_SB
                    LOG10_TEAM_FIELDING_E
                    LOG10_TEAM_PITCHING_H
                    m_TEAM_BASERUN_CS
                    m_TEAM_BASERUN_SB
                    m_TEAM_BATTING_SO
                    m_TEAM_FIELDING_DP
                    OBP_TEAM_BATTING
                    TEAM_PITCHING_HR
                    / link=identity dist=normal;
run;

*/

/*DATA SCORING STEP*/


data SCOREFILE;
```

```
set mydata.moneyball_test;

        IMP_TEAM_BATTING_SO = TEAM_BATTING_SO;
        m_TEAM_BATTING_SO = 0;
        if missing(IMP_TEAM_BATTING_SO) then do;
                IMP_TEAM_BATTING_SO = 750;
                m_TEAM_BATTING_SO = 1;
        end;

        IMP_TEAM_BASERUN_SB = TEAM_BASERUN_SB;
        m_TEAM_BASERUN_SB = 0;
        if missing(IMP_TEAM_BASERUN_SB) then do;
                IMP_TEAM_BASERUN_SB = 101;
                m_TEAM_BASERUN_SB = 1;
        end;

        IMP_TEAM_BASERUN_CS = TEAM_BASERUN_CS;
        m_TEAM_BASERUN_CS = 0;
        if missing(IMP_TEAM_BASERUN_CS) then do;
                IMP_TEAM_BASERUN_CS = 49;
                m_TEAM_BASERUN_CS = 1;
        end;

        IMP_TEAM_PITCHING_SO = TEAM_PITCHING_SO;
        m_TEAM_PITCHING_SO = 0;
        if missing(IMP_TEAM_PITCHING_SO) then do;
                IMP_TEAM_PITCHING_SO = 813;
                m_TEAM_PITCHING_SO = 1;
```

```
	end;

	IMP_TEAM_FIELDING_DP = TEAM_FIELDING_DP;
	m_TEAM_FIELDING_DP = 0;
	if missing(IMP_TEAM_FIELDING_DP) then do;
		IMP_TEAM_FIELDING_DP = 149;
		m_TEAM_FIELDING_DP = 1;
	end;

	IMP_TEAM_BATTING_HBP = TEAM_BATTING_HBP;
	m_TEAM_BATTING_HBP = 0;
	if missing(IMP_TEAM_BATTING_HBP) then do;
		m_TEAM_FIELDING_DP = 1;
	end;

/*Transform data to reduce problems from outliers */

	CAP_TEAM_PITCHING_H = TEAM_PITCHING_H;
	if CAP_TEAM_PITCHING_H < 1244 then CAP_TEAM_PITCHING_H = 1244;
	if CAP_TEAM_PITCHING_H > 7000 then CAP_TEAM_PITCHING_H = 7000;

	CAP_TEAM_PITCHING_SO = IMP_TEAM_PITCHING_SO;
	if CAP_TEAM_PITCHING_SO < 208 then CAP_TEAM_PITCHING_SO = 208;
	if CAP_TEAM_PITCHING_SO > 1464 then CAP_TEAM_PITCHING_SO = 1464;

	CAP_TEAM_PITCHING_BB = TEAM_PITCHING_BB;
	if CAP_TEAM_PITCHING_BB < 237 then CAP_TEAM_PITCHING_BB = 237 ;
	if CAP_TEAM_PITCHING_BB > 924 then CAP_TEAM_PITCHING_BB = 924;
```

```
CAP_TEAM_BATTING_3B = TEAM_BATTING_3B;
if CAP_TEAM_BATTING_3B < 17 then CAP_TEAM_BATTING_3B = 17;
if CAP_TEAM_BATTING_3B > 134 then CAP_TEAM_BATTING_3B = 134;

CAP_TEAM_BATTING_2B = TEAM_BATTING_2B;
if CAP_TEAM_BATTING_2B < 141 then CAP_TEAM_BATTING_2B = 141;
if CAP_TEAM_BATTING_2B > 352 then CAP_TEAM_BATTING_2B = 352;

CAP_TEAM_BATTING_H = TEAM_BATTING_H;
if CAP_TEAM_BATTING_H < 1188 then CAP_TEAM_BATTING_H = 1188;
if CAP_TEAM_BATTING_H > 1950 then CAP_TEAM_BATTING_H = 1950;

CAP_TEAM_BATTING_HR = TEAM_BATTING_HR;
if CAP_TEAM_BATTING_HR < 4 then CAP_TEAM_BATTING_HR = 4;
if CAP_TEAM_BATTING_HR > 235 then CAP_TEAM_BATTING_HR = 235;

OBP_TEAM_BATTING = TEAM_BATTING_BB + CAP_TEAM_BATTING_H ;
TEAM_BATTING_1B = CAP_TEAM_BATTING_H - CAP_TEAM_BATTING_2B - CAP_TEAM_BATTING_3B -
CAP_TEAM_BATTING_HR ;

CAP_TEAM_BATTING_1B = TEAM_BATTING_1B;
if CAP_TEAM_BATTING_1B < 882 then CAP_TEAM_BATTING_1B = 882;
if CAP_TEAM_BATTING_1B > 1486 then CAP_TEAM_BATTING_1B = 1486;

TEAM_BATTING_SLG = CAP_TEAM_BATTING_1B + (2*CAP_TEAM_BATTING_2B) +
(3*CAP_TEAM_BATTING_3B) + (4*CAP_TEAM_BATTING_HR);
TEAM_BATTING_SLG_RATIO =  (CAP_TEAM_BATTING_2B) + (CAP_TEAM_BATTING_3B) +
(CAP_TEAM_BATTING_HR)/CAP_TEAM_BATTING_H;
```

```
        CAP_TEAM_FIELDING_DP = IMP_TEAM_FIELDING_DP;
        if CAP_TEAM_FIELDING_DP < 80 then CAP_TEAM_FIELDING_DP = 80;
        if CAP_TEAM_FIELDING_DP  > 202 then CAP_TEAM_FIELDING_DP = 202;

        LOG10_TEAM_BASERUN_SB = LOG10(IMP_TEAM_BASERUN_SB);
        LOG10_TEAM_FIELDING_E = LOG10(TEAM_FIELDING_E);
        LOG10_CAP_TEAM_BATTING_HR = LOG10(CAP_TEAM_BATTING_HR);
        LOG10_TEAM_PITCHING_H = LOG10(TEAM_PITCHING_H);

drop TEAM_BATTING_HBP;
drop IMP_TEAM_BATTING_HBP;

P_TARGET_WINS =
        232.99000
+          0.15327   * CAP_TEAM_BATTING_3B
+               -0.09704  * CAP_TEAM_FIELDING_DP
+               0.00902   * CAP_TEAM_PITCHING_H
+               -0.01313  * CAP_TEAM_PITCHING_SO
+               0.05885   * IMP_TEAM_BASERUN_SB
+               -48.41700 * LOG10_TEAM_FIELDING_E
+               -40.20204 * LOG10_TEAM_PITCHING_H
+               3.56888   * m_TEAM_BASERUN_CS
+               34.89288  * m_TEAM_BASERUN_SB
+               9.00281   * m_TEAM_BATTING_SO
+               5.90136   * m_TEAM_FIELDING_DP
+               0.03326   * OBP_TEAM_BATTING
+               0.06249   * TEAM_PITCHING_HR
```

```
                    ;
if P_TARGET_WINS  = 0 then P_TARGET_WINS = 81;
if P_TARGET_WINS  = . then P_TARGET_WINS = 81;
if P_TARGET_WINS < 38 then P_TARGET_WINS = 38;
if P_TARGET_WINS  > 114 then P_TARGET_WINS = 114;

keep INDEX;
keep P_TARGET_WINS;

run;

*proc print data = SCOREFILE(obs=5);
*run;

/*  */
libname ddlib '/home/daviddietrich2013/' ;

data ddlib.DAVID_DIETRICH_FILE;
set SCOREFILE;
run;

proc print data = ddlib.DAVID_DIETRICH_FILE;
run;
```

BINGO BONUS

- (20 points) Try PROC GLM and PROC GENMOD for the OLS portion and rerun

PROC GLM was tried, using the same variables in Model 3. The outputs were different than the regular PROC REG used in most of the paper. With PROC GLM, some of the outputs were slightly different, though generally consistent. For instance, the output for PROC GLM provided the same R-Square value (0.4161), although by default it did not provide an adjusted R square value. Also, the output tables provided additional information on mean square values for each variable, as well as Type 1 SS and Type III SS. From the SAS user manual, "Type I sums of squares (SS), also called *sequential sums of squares*, are the incremental improvement in error sums of squares as each effect is added to the model." This seems somewhat analogous to the Partial R scores in the regular PROC REG statement, which show the contribution of each variable to the model's overall R-Square and Adj R Square value. Although the Type III SS are shown as the "partial sums of squares" and would more properly correspond to the Partials from PROC REG.

Similarly, PROC GENMOD was run with the same variables from Model 3 (and from PROC GLM). It appears that this uses a link function, which can be specified to behave as an OLS as it was here with an identify link function. Similar to the PROC GLM, executing PROC GENMOD created a model with the same coefficients and significance levels, although the output was a bit different. In this case, one of the key diagnostic outputs was the Wald Chi-Square. After doing some research into the Wald Chi-Square, it shows it is a likelihood estimate for each of the variables in the model having influence on the dependent variable (Target Wins in this case). In this case, the "estimate" value for each variable are the same as the parameter scores from a regular OLS run with PROC REG, suggesting that the PROC GENMOD has other options that it can be run with, but when run with an identify link function, it behaves similar to a regular OLS with PROC REG.

- *(10 pts) Hand in your scored data as a SAS Data Set (see attached)*