FROM: John Hokkanen, 411-Section 60

TO: Dr. Joseph Wilck

DATE: October 1, 2016

PROJECT: Create linear regression prediction algorithm to estimate baseball team wins, produce a list of estimates, and document the process of model selection.

## SUMMARY

The following document explains the methodology used to evaluate the moneyball dataset and prepare a prediction method. The supplied data set has a handful of variables that have missing data values. One variable was deleted because the vast majority of values were missing, but the others could be managed by assigning median values. Extreme outliers (e.g., above 1%) were trimmed back on a few variables, and a few variables were transformed because they were skewed A few new combination variables were created for modeling.

Because the dataset contains data from multiple eras of baseball, it is noisy and difficult to model. After developing an initial linear regression, additional effort was taken to classify the data and segment it into various buckets. Out of an abundance of caution about overfitting, the final method uses only four buckets, with a linear regression equation having been separately tuned to each bucket of data. This method was validated with a randomly selected portion of the data that had been set aside, and the combined regression approach appears to be both valid and stable. To help ensure stability on future data, parsimony was an important consideration, and, consequently, variables with higher VIFs and variables that ran contrary to understood logical relationships were avoided. Equations and performance data are set forth at the end of the document.

## PREPARATION

Lacking in knowledge of baseball, I first obtained some background knowledge. This was accomplished by watching the movie Moneyball, reading articles on Wikipedia about baseball (e.g., the "deadball era"), downloading and examining current performance data from Major League Baseball ("MLB"), and reading some web articles about various statistics (e.g., the worst ten baseball seasons). With some rudimentary business knowledge about the numbers and modern thinking about variables matter (e.g., on-base percentage), I began the exploration and preparation of the data set.

## DATA EXPLORATION

Though separated for this write-up, the process of data exploration and preparation was a combined, iterative process. I would examine the data, make changes to how it was handled, and examine it some more. For example, I might use WEKA to review data, see what categories it used to classify the data, and prepare additional data elements.

The data process began with the partitioning of the training data into a training data set (80%) and a validation data set (20%). The validation data was set aside for model validation and

comparison purposes; however, it was used to confirm the final output range of the target variable and it was also inspected to ensure that it did not have missing values for a variable that had none in the training set.

At the most basic level, the SAS proc means output revealed a number of important facts about the data. (Note: Actual outputs may have other irrelevant items like Random_Num that have been removed for discussion.)

| Variable | Label | Mean | Median | Mode | N | N Miss | Minimum | Maximum |
|---|---|---|---|---|---|---|---|---|
| TARGET_WINS | | 80.58 | 82 | 83 | 1828 | 0 | 0 | 146 |
| TEAM_BATTING_H | Base Hits by batters | 1468.14 | 1453 | 1458 | 1828 | 0 | 891 | 2554 |
| TEAM_BATTING_2B | Doubles by batters | 241.18 | 239 | 220 | 1828 | 0 | 69 | 458 |
| TEAM_BATTING_3B | Triples by batters | 54.75 | 47 | 35 | 1828 | 0 | 0 | 197 |
| TEAM_BATTING_HR | Homeruns by batters | 99.37 | 102 | 108 | 1828 | 0 | 0 | 264 |
| TEAM_BATTING_BB | Walks by batters | 501.32 | 512 | 492 | 1828 | 0 | 0 | 878 |
| TEAM_BATTING_SO | Strikeouts by batters | 737.40 | 752 | 0 | 1742 | 86 | 0 | 1399 |
| TEAM_BASERUN_SB | Stolen bases | 124.73 | 101 | 69 | 1724 | 104 | 0 | 697 |
| TEAM_BASERUN_CS | Caught stealing | 52.67 | 49 | 37 | 1208 | 620 | 0 | 201 |
| TEAM_BATTING_HBP | Batters hit by pitch | 59.47 | 58 | 56 | 159 | 1669 | 29 | 90 |
| TEAM_PITCHING_H | Hits allowed | 1783.25 | 1516 | 1400 | 1828 | 0 | 1137 | 30132 |
| TEAM_PITCHING_HR | Homeruns allowed | 105.44 | 107 | 114 | 1828 | 0 | 0 | 320 |
| TEAM_PITCHING_BB | Walks allowed | 553.47 | 537 | 495 | 1828 | 0 | 0 | 3645 |
| TEAM_PITCHING_SO | Strikeouts by pitchers | 824.29 | 817 | 0 | 1742 | 86 | 0 | 19278 |
| TEAM_FIELDING_E | Errors | 246.77 | 158 | 122 | 1828 | 0 | 65 | 1898 |
| TEAM_FIELDING_DP | Double Plays | 146.63 | 149 | 149 | 1597 | 231 | 64 | 228 |

Central findings arising from the proc means are as follows:

- One variable, TEAM_BATTING_HBP has values in less than 10% of the records (shown in red). Though it may have value, the time involved to investigate that question would be better spent on other modeling questions. Therefore, it would be removed from the list of variables to be considered as inputs.
- Five other variables, highlighted in blue, have missing values which will need to be assigned values. Examination of the mean and median of three of the variables (shown in orange) reveals a substantial discrepancy between the mean and median. This discrepancy is presumably caused by outliers due to variety in the data from various baseball eras or unknown reason. To keep it simple, all variables were treated the same and had missing values replaced by their median values. Imputation of values by other method was set aside for later consideration.
- Three variables show suspect maximum values (shown in yellow). MLB data was examined for last year maximums, which showed a greater-than-fivefold value for the following items:

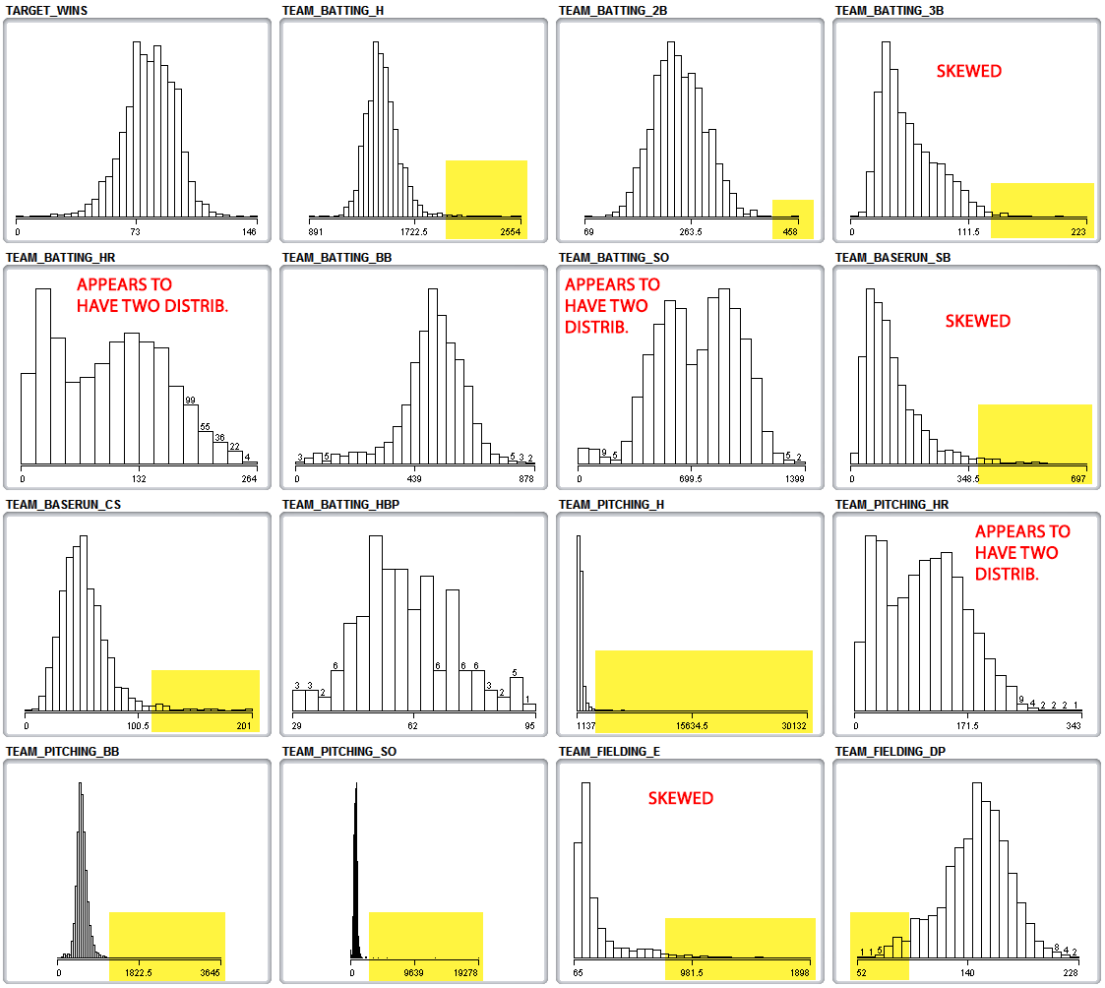
| Variable | Ratio of Max to MLB Max |
|---|---|
| TEAM_PITCHING_H | 19.86288728 |
| TEAM_PITCHING_BB | 6.157094595 |

| | |
|---|---|
| TEAM_PITCHING_SO | 13.73076923 |
| TEAM_FIELDING_E | 15.30645161 |

Bar charts would be examined for these outlier values specifically for but also for other variables.

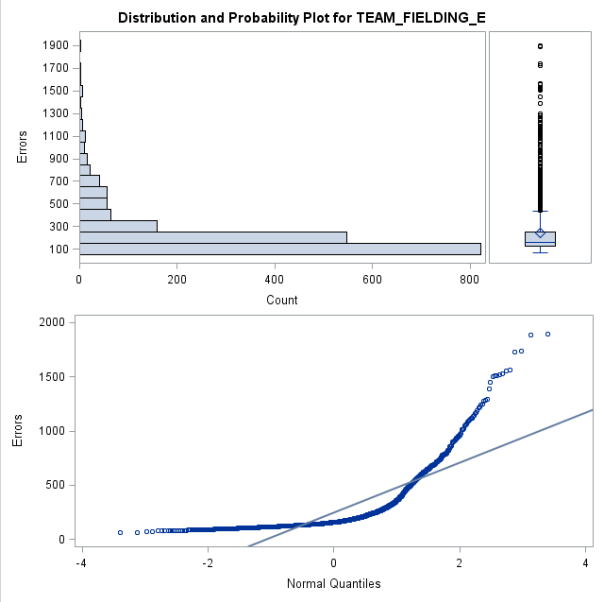
- The minimum value for Target_Wins, shown in green, is 0. While theoretically possible, review of MLB data shows that no team has ever had a zero-win season, and therefore the data must be missing. Inspection of the data revealed that this problem was limited to a single training record. Rather than guess or impute an output value for this one training record, it was deleted from model development.

Further data inspection was performed with both SAS and WEKA's interactive environment. Though SAS has comprehensive reports, WEKA's interactive environment for basic data exploration was very useful since I am on assignment without a printer. Here are visualizations from WEKA with my markups:

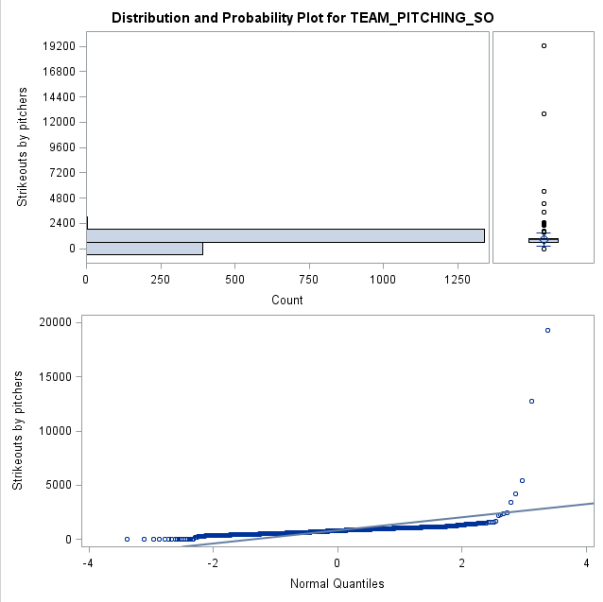



I have highlighted areas where it appears that there are outliers, and I have noted where distributions are skewed or appear to be the combination of two distributions (possibly arising

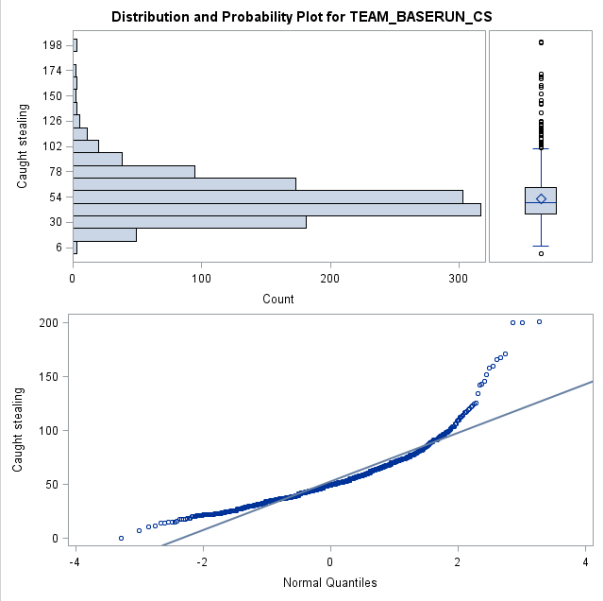
from differences in early baseball and then modern baseball). Further inspection with SAS's tools like proc univariate revealed potential issues with some of the variables. For space purposes, I will illustrate with four examples some issues:



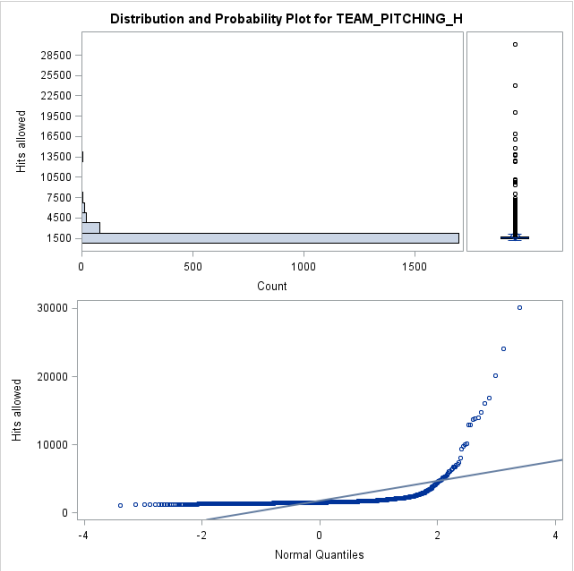

As the SAS output shows, Team_Fielding_E is both highly skewed and has a long tail.




Very substantial outliers at the top end that need to be addressed in Team_Pitching_SO.




Team_Baserun_CS has a long tail, and the outliers at the top end will have a significant effect unless modified.



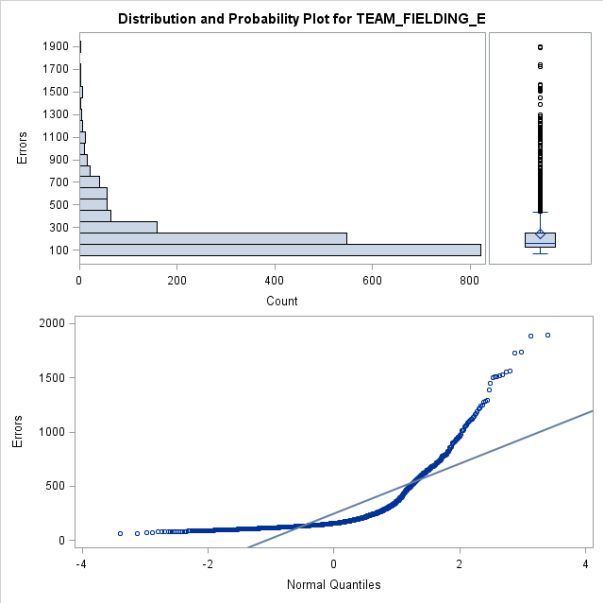
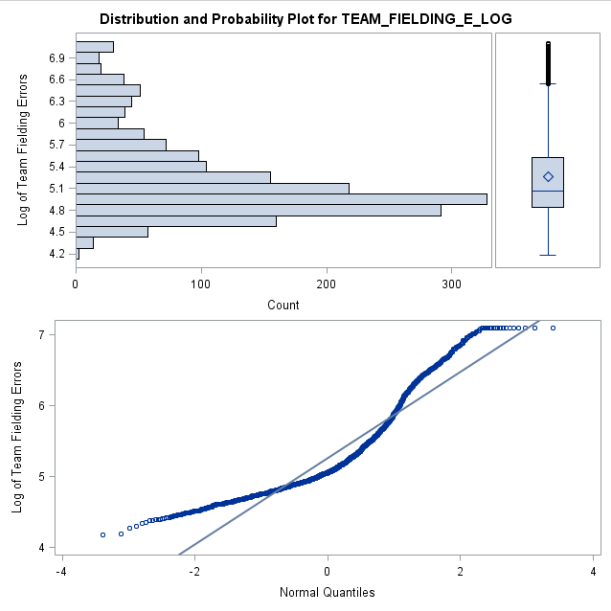
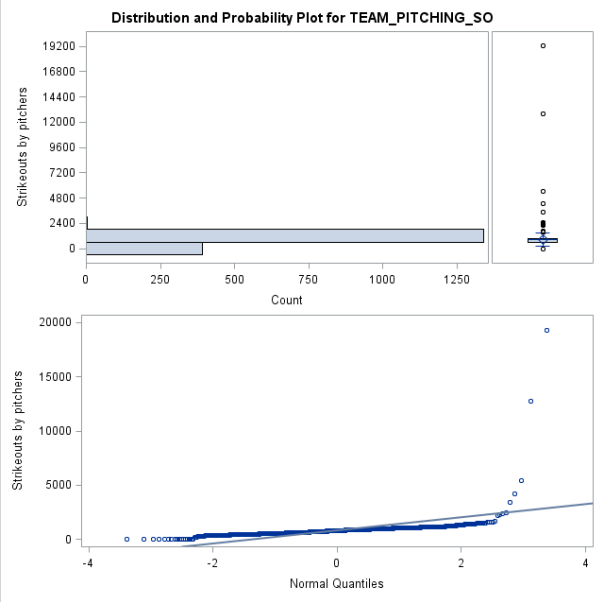
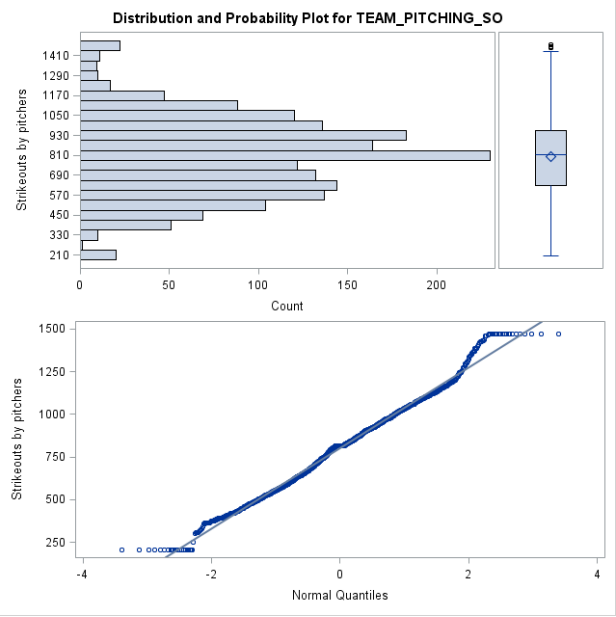

Team_Pitching_H has values that go completely beyond reason as previously identified.

## DATA PREPARATION

Based on the prior analysis, the following table shows the modifications to the raw data performed:

| Variable(s) | Action taken |
|---|---|
| M_TEAM_FIELDING_DP<br>M_TEAM_BASERUN_SB<br>M_TEAM_BATTING_SO<br>M_TEAM_PITCHING_SO<br>M_TEAM_BASERUN_CS | Missing values set to median |
| TEAM_PITCHING_H<br>TEAM_PITCHING_BB | Top 1% of values set to most adjacent value. The bottom end of these did not appear to have significant outliers and was left unmodified. |
| TEAM_BASERUN_CS<br>TEAM_PITCHING_SO<br>TEAM_FIELDING_E | Top and Bottom 1% set to most adjacent values. |
| TEAM_BATTING_H<br>TEAM_BATTING_2B<br>TEAM_BATTING_3B | Despite the top end having very high values, the first three variables were initially left unmodified because a consolidation variable, TEAM_ONBASE_COUNT (see below) was created using the first three. After the models were evaluated, the onbase count was disaggregated and original variables included.<br><br>When the disaggregated variables were then used for regressions, the variables were then trimmed to have the top 1% reduced to the adjacent values. However, given the bucketing that was being done, this action actually reduced the information signal within the data and RSquared value went down. In short, it was better to leave the raw, higher values, and so they were left intact in their raw form for final model generation. |
| TEAM_BASERUN_SB | Similar to the previous section, it was discovered that there was information in the top end of the values, and so the trimming of the top was eliminated and the variable was left intact. |
| TEAM_BATTING_1B<br>TEAM_ONBASE_COUNT<br>TEAM_BASES_TOTAL | These were newly calculated counts based on existing variables. OnBase_Count was hits plus walks; it was created on the basis of current thinking about what makes a winning baseball team. Bases_Total weighted the counts for the base value they represented, e.g., 2x for doubles, 3x for triples. As will be shown later, the new consolidated variable that performed best was TEAM_ONBASE_COUNT and the others were eventually discarded. |
| TEAM_ONBASE_TO_SO | Other variables like onbase to strikeout ratio were calculated. They were largely found to not have significant predictive value. |
| TEAM_BATTING_3B_LOG<br>TEAM_BASERUN_SB_LOG<br>TEAM_FIELDING_E_LOG<br>TEAM_PITCHING_H_LOG | Variables that had skewed distributions had separate log variables computed to provide a more even weighting for the variables in the regression equations. |
| TEAM_BATTING_HBP | This variable was dropped because of too many missing values. |
| TARGET_WINS=**0** | 1 record deleted |

The effects of the missing data and transformations can be seen as follows in the following two examples:

| ORIGINAL VARIABLE | MODIFIED VARIABLE |
|---|---|
|  |  As you can see, the variable has been substantially transformed trimming the end and taking the log funciton. |
|  |  Like Team_Baserun_CS (below), Team_Pitching_SO was made more usable and tracked the quantile plot much better after the missing values and outliers were adjusted. |

## Collinearity

As an initial matter, my examination of collinear variables focused on variables which had logical relationships, namely the collinearity among offensive variables and the same with

defensive variables. I discovered correlations across the two categories, but those correlations probably have the baseball era as their origin rather than any inherent logical relationship. For example, TEAM_BATTING_HR and TEAM_PITCHING_HR are strongly correlated (.969), but that is almost surely due to the general discrepancies between the magnitudes of the numbers across eras. Other variables like TEAM_BATTING_H and TEAM_FIELDING_E are also correlated to a lesser degree (.23), but probably for the same reason. I realize that it is *possible* that teams hiring for offense might do so at the expense of defense, but teasing out such relationships would require further exploration.

Within an offense/defense category, variables had a variety of collinearity. As expected, TEAM_ONBASE_COUNT, as an aggregation of the offensive hitting variables, correlated highly. Among the primary offensive hitting variables, many showed collinearity, but again, this came as no surprise since different eras (e.g., deadball era) had more hitting across the board. Interestingly, TEAM_BATTING_2B correlated substantially only with TEAM_BATTING_3B (.435) and TEAM_BATTING_BB (.262). (As it would turn out in later model development, these two variables were of interest as significant inputs because of their added value.) Other relationships (e.g., stolen bases to caught stealing, which had a .235 correlation) would have been a surprise only if the correlation did not exist. Among the defensive variables it came as no surprise that TEAM_PITCHING_H correlated with most other defensive variables.

The collinearity analysis immediately suggested that it might be wise to pursue common factor analysis to determine whether common factors might be developed within each category. This effort was discarded after Professor Wilck commented that such efforts had not been fruitful in the past. Consequently, I concluded that I would seek instead to use a minimum of unrelated variables to avoid the brittleness that might arise from regressions with numerous collinear variables.

## MODEL DEVELOPMENT

Having fully resolved missing values, outlier values, and skewed distributions, models were developed in an iterative way first by generating general models and then looking at models with trimmed down variable choices made to remove problems. The first models were generated by allowing SAS to do variable selection on all variables based on the forward, backward, and stepside methods. The forward selection fit is shown as follows:

| Forward Variable | Label | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| | Tolerance | Variance Inflation |
|---|---|---|---|---|---|---|---|---|
| **Intercept** | Intercept | **1** | **-189.0361** | 60.8323 | -3.11 | 0.0019 | . | 0 |
| **TEAM_ONBASE_COUNT** | On Base Count | **1** | -0.04457 | 0.02243 | -1.99 | 0.0471 | 0.00523 | 191.315 |
| **TEAM_BASES_TOTAL** | Total Bases | **1** | 0.09028 | 0.02779 | 3.25 | 0.0012 | 0.00168 | **593.71** |
| **TEAM_ONBASE_PERCENT** | Hits/Onbase Percent | **1** | -3.78377 | 0.66364 | -5.7 | <.0001 | 0.00724 | 138.028 |
| **TEAM_ONBASE_TO_SO** | Onbase to Strikeout Ratio | **1** | -0.56376 | 0.23446 | -2.4 | 0.0163 | 0.2574 | 3.88499 |
| **TEAM_BATTING_2B** | Doubles by batters | **1** | -0.11746 | 0.02965 | -3.96 | <.0001 | 0.04465 | 22.3956 |
| **TEAM_BATTING_3B** | Triples by batters | **1** | -0.0373 | 0.06091 | -0.61 | **0.5403** | 0.03095 | 32.3113 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **TEAM_BATTING_HR** | Homeruns by batters | **1** | -0.22867 | 0.08269 | -2.77 | 0.0057 | 0.0035 | **285.651** |
| **TEAM_BASERUN_SB** | Stolen bases | **1** | 0.03976 | 0.00667 | 5.96 | <.0001 | 0.26932 | 3.71302 |
| **TEAM_BASERUN_SB_LOG** | Log of Stolen Bases | **1** | -0.41198 | 0.67996 | -0.61 | **0.5447** | 0.32998 | 3.0305 |
| **TEAM_BASERUN_CS** | Caught stealing | **1** | -0.03893 | 0.01986 | -1.96 | 0.0501 | 0.81779 | 1.2228 |
| **TEAM_PITCHING_H** | Hits allowed | **1** | -0.00714 | 0.00733 | -0.97 | 0.3301 | 0.00281 | **356.278** |
| **TEAM_PITCHING_H_MEDIAN** | | **1** | -0.0067 | 0.00742 | -0.9 | **0.3662** | 0.00494 | 202.396 |
| **TEAM_PITCHING_H_LOG** | Log of Pitching Hits | **1** | 70.97683 | 26.7504 | 2.65 | 0.008 | 0.00172 | 580.011 |
| **TEAM_PITCHING_H_MEDIAN_LOG** | Log of Pitching_Median Hits | **1** | 19.34481 | 25.1389 | 0.77 | **0.4417** | 0.00277 | 360.798 |
| **TEAM_PITCHING_BB** | Walks allowed | **1** | -0.07064 | 0.01175 | -6.01 | <.0001 | 0.04721 | 21.1799 |
| **TEAM_PITCHING_SO** | Strikeouts by pitchers | **1** | -0.00801 | 0.00216 | -3.7 | 0.0002 | 0.33323 | 3.00089 |
| **TEAM_FIELDING_E_LOG** | Log of Team Fielding Errors | **1** | -14.98116 | 1.26083 | -11.88 | <.0001 | 0.14748 | 6.78071 |
| **TEAM_FIELDING_DP** | Double Plays | **1** | -0.1197 | 0.01417 | -8.45 | <.0001 | 0.72443 | 1.38039 |

The variables highlighted in yellow were those selected by the stepwise method. The backward method selected the same stepwise variables, plus one (TEAM_BATTING_1B). I note the following observations about the forward selection:

- The four variables that were excluded by the stepwise selection were avoided due to the low t values, thus rendering the variables not significant. See red scores. It is possible that one or more of these t values could change depending on the mix, but they are suspect as variables.
- A few variables were simply transformed versions of themselves (e.g., TEAM_PITCHING_H_LOG); if at all, only one of the versions should be used.
- Other than the transformed versions, some variables also had high variance inflation, suggesting significant collinearity with other variables. This came as no surprise given that TEAM_BASES_TOTAL was simply a weighted version of TEAM_ONBASE_COUNT. Because TEAM_ONBASE_COUNT is the most significant, this variable will be used for modeling while avoiding the others.
- Finally, the intercept value looks quite strange to me as a high negative number. In general teams win more than 40 games, and I theorize that a good working model would have a positive number. This needs to be inspected in the proposed models.

Using the information above, additional iterations of running forward, backward, and stepwise model selections yielded the following reduced set of variables in a stepwise selection:

| Variable | Label | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| | Tolerance | Variance Inflation |
|---|---|---|---|---|---|---|---|---|
| **Intercept** | Intercept | **1** | 128.9115 | 10.11335 | 12.75 | <.0001 | . | 0 |
| **TEAM_ONBASE_COUNT** | On Base Count | **1** | 0.04615 | 0.00224 | 20.62 | <.0001 | 0.49212 | 2.03202 |
| **TEAM_BATTING_SO** | Strikeouts by batters | **1** | -0.0186 | 0.002 | -9.31 | <.0001 | 0.38564 | 2.59312 |
| **TEAM_BASERUN_SB** | Stolen bases | **1** | 0.06143 | 0.00521 | 11.79 | <.0001 | 0.41351 | 2.41832 |
| **TEAM_BASERUN_CS** | Caught stealing | **1** | -0.05835 | 0.01908 | -3.06 | 0.0023 | 0.83049 | 1.2041 |
| **TEAM_PITCHING_BB** | Walks allowed | **1** | -0.0144 | 0.00322 | -4.48 | <.0001 | 0.59062 | 1.69314 |
| **TEAM_FIELDING_E_LOG** | Log of Team | **1** | -20.82077 | 1.28243 | -16.24 | <.0001 | 0.13357 | 7.4866 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Fielding Errors | | | | | | | |
| **TEAM_FIELDING_DP** | Double Plays | **1** | -0.10545 | 0.01524 | -6.92 | <.0001 | 0.58667 | 1.70453 |
| **M_TEAM_PITCHING_SO** | Team Pitching Strike Out Missing Flag | **1** | 9.7952 | 1.62581 | 6.02 | <.0001 | 0.68698 | 1.45564 |
| **M_TEAM_BATTING_SOZero** | | **1** | 5.83216 | 3.39123 | 1.72 | 0.0856 | 0.76831 | 1.30155 |
| **M_TEAM_BASERUN_SB** | Team Baserun Stolen Bases Missing Flag | **1** | 33.42745 | 1.92938 | 17.33 | <.0001 | 0.40759 | 2.45342 |
| **M_TEAM_FIELDING_DP** | Team Fielding DP Missing Flag | **1** | 4.9093 | 1.66132 | 2.96 | 0.0032 | 0.26819 | 3.7287 |

This model looked better in a number of ways. First, the intercept is now positive, though possibly too high. All the excessively collinear variables are gone and the VIFs were fine. The t test values are acceptable, with the exception of M_TEAM_BATTING_SOZero, a flag which represented whether there was a zero (i.e., missing value). Consequently, this variable would be dropped. Nevertheless, I thought that there were still too many variables, especially among the flag variables.

To further analyze the data, I pulled the training data into Excel so that I could browse around the data with filters and sorts. In so doing, I concluded that M_TEAM_BASERUN_SB was quite an interesting indicator. It had substantially more top win records than random chance would suggest. The other missing item records were interesting, but less stark, and so I decided to focus on this one, and after some additional running of models and removing variables, I settled on the following model as a single regression baseline model:

| Variable | Label | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| | Tolerance | Variance Inflation |
|---|---|---|---|---|---|---|---|---|
| **Intercept** | Intercept | **1** | 84.70323 | 6.03322 | 14.04 | <.0001 | . | 0 |
| **TEAM_ONBASE_COUNT** | On Base Count | **1** | 0.04056 | 0.00192 | 21.13 | <.0001 | 0.69089 | 1.4474 |
| **TEAM_BATTING_3B** | Triples by batters | **1** | 0.14548 | 0.01612 | 9.03 | <.0001 | 0.42761 | 2.33858 |
| **TEAM_BASERUN_SB** | Stolen bases | **1** | 0.04349 | 0.0044 | 9.89 | <.0001 | 0.59857 | 1.67064 |
| **TEAM_FIELDING_E_LOG** | Log of Team Fielding Errors | **1** | -15.76686 | 0.90099 | -17.5 | <.0001 | 0.27944 | 3.57858 |
| **TEAM_FIELDING_DP** | Double Plays | **1** | -0.10829 | 0.01344 | -8.06 | <.0001 | 0.77948 | 1.28291 |
| **M_TEAM_BASERUN_SB** | Team Baserun Stolen Bases Missing Flag | **1** | 27.95732 | 1.76416 | 15.85 | <.0001 | 0.50343 | 1.98637 |

This model had an adjusted rsquared value of about .38, and so that was the baseline rsquared value that any subsequent model had to beat to be considered.

## MODEL SELECTION

The foregoing work was simply the starting point for more enhanced model selection. I thought that I might improve on the single regression model performance by segmenting the data into various component pieces and developing partial models tuned to these segments of the data; the net result would be an average total performance that would exceed the performance of the single regression equation.

I ran many model experiments using various categories. To facilitate that, I turned to WEKA to examine how various classifiers would select models. As part of that process, I used the continuous Target_Wins variable to run the classifiers with the continuous target variable. However, I also broke Target_Wins into various categories (from 2 to 10 categories) to use other classifiers that would use the categories. Some (e.g., DecisionStump) generated very simple categories like TEAM_ONBASE_CNT <= 2036.5 : 76.95, TEAM_ONBASE_CNT > 2036.5 : 88.43, TEAM_ONBASE_CNT missing : 80.82). Even this information was useful because it confirmed my previous analysis that TEAM_ONBASE_COUNT was a top variable.

Another experiment involved running a regression on only the data items where TEAM_BASERUN_SB was missing. I discovered that this was a wonderful break in the data and a regression against this missing data could exceed a .6 adjusted rsquared value. Having then removed the missing items, I identified my next data slice by using WEKA, and iteratively so, developing each subsequent classification on the further reduced dataset. Because I wanted to keep the data categories broad to avoid overfitting (and resulting in a model unreliable for prediction), I finally settled on four buckets of data. I speculate as to the reason for the classification, but, in the end, I do not know and am simply letting the data drive the outcome:

| Segment | Thoughts as to segmentation |
|---|---|
| IF M_TEAM_BASERUN_SB NE **0** | Capture high performance resulting from missing values not falling uniformly across all win categories. |
| THEN,<br>IF TEAM_ONBASE_COUNT<=**1916** | This segment kept reappearing across various WEKA categorizers. It probably represents the category which includes a large portion of modern baseball where the sum of all hits+walks is less than 1916. |
| THEN<br>IF TEAM_BATTING_3B<**54.5** | The exact underlying reason for this segment was not clear to me, but I speculate that it is seeking to capture another portion of modern baseball teams which was not captured by the previous segment. |
| THEN<br>Everything else | This is just a remainder bucket which contains a mishmash of remaining data. As it would turn out, it would be difficult to get good performance out of this data. |

Having identified this data segmentation, I then began the process of trying to obtain the best models in each category. With more knowledge about the data and a baseline model in hand, I redeveloped models for each data segment. In this process, I began with the base model, with a goal of using fewer rather than more variables.

In the process, I also reexamined my initial assumption that I should use TEAM_ONBASE_COUNT. Though I was very satisfied with this variable in the single regression, I wondered whether it would perform as well as the underlying counts for the individual data segments. By removing it and developing regressions with the individual counts, I developed regressions with better adjusted rsquared performance within each category but one. Doing

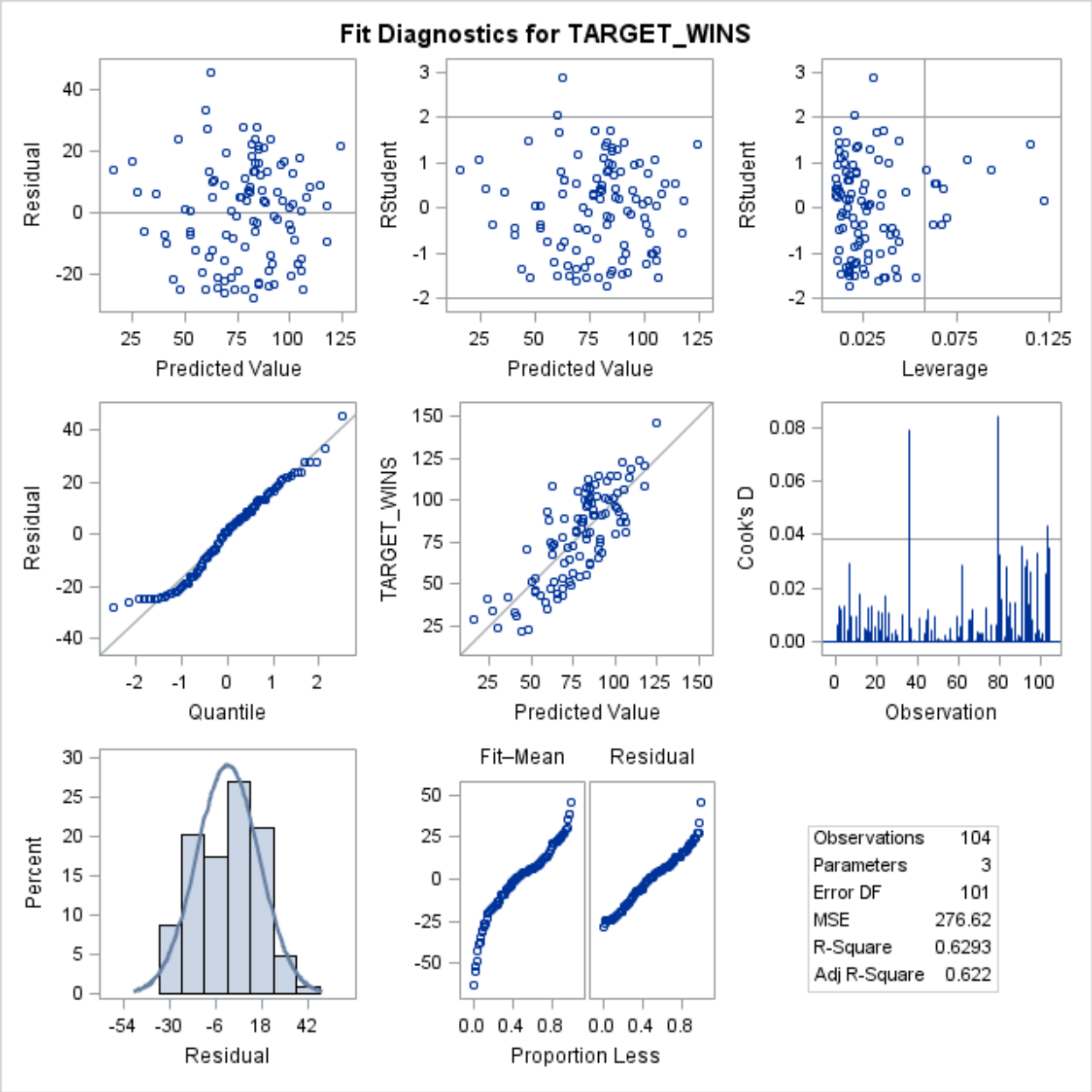
this complicated matters because I discovered that some variables were no longer significant for the data segment on which I was working, and so it lengthened the development work. The final five models are as follows:

| Model | Equation | Training Adj Rsquared |
|---|---|---|
| General-purpose champion | 91.43086 + TEAM_BATTING_1B*0.05851 + TEAM_BATTING_3B*0.19044 + TEAM_BATTING_HR*0.06798 + TEAM_BATTING_BB*0.03153 + TEAM_BASERUN_SB*0.04801 + TEAM_FIELDING_E_LOG*-18.63946 + TEAM_FIELDING_DP*-0.10908 + M_TEAM_BASERUN_SB*28.56166 | 0.3896 |
| Category 1<br>M_TEAM_BASERUN_SB NE 0 | 222.79316 + TEAM_ONBASE_COUNT*0.08085 + TEAM_FIELDING_E_LOG*-42.32208 | 0.6220 |
| Category 2<br>TEAM_ONBASE_COUNT<=1916 | 134.53606 + TEAM_BATTING_1B*0.03722 + TEAM_BATTING_3B*0.22457 + TEAM_BATTING_HR*0.05541 + TEAM_BATTING_BB*0.0214 + TEAM_BASERUN_SB*0.07427 + TEAM_FIELDING_E_LOG*-21.73163 + TEAM_FIELDING_DP*-0.14339 | 0.3659 |
| Category 3<br>IF TEAM_BATTING_3B<54.5 | 88.95051 + TEAM_BATTING_1B*0.04008 + TEAM_BATTING_2B*-0.03403 + TEAM_BATTING_3B*0.14123 + TEAM_BATTING_HR*0.06973 + TEAM_BATTING_BB*0.04436 + TEAM_BASERUN_SB*0.02833 + TEAM_FIELDING_E_LOG*-14.35022 + TEAM_FIELDING_DP*-0.07433 | 0.2345 |
| Category 4<br>Anything remaining | 105.76296 + TEAM_BATTING_1B*0.06091 + TEAM_BATTING_HR*0.12878 + TEAM_BASERUN_SB*0.0621 + TEAM_FIELDING_E_LOG*-16.94452 + TEAM_FIELDING_DP*-0.10968 | 0.2958 |

It is interesting to see how the intercept values and variables changed for the different data segments. I do not have enough baseball knowledge to explain why these variables were the most relevant for the respective data categories, but I do note that all of them had both an offensive and defensive component. Now, with a relatively high degree of confidence in my models, I proceeded to to run some additional checks on the models by reviewing diagnostic and residual plots and running the General Linear Model procedure to see coefficients were returned with that procedure.

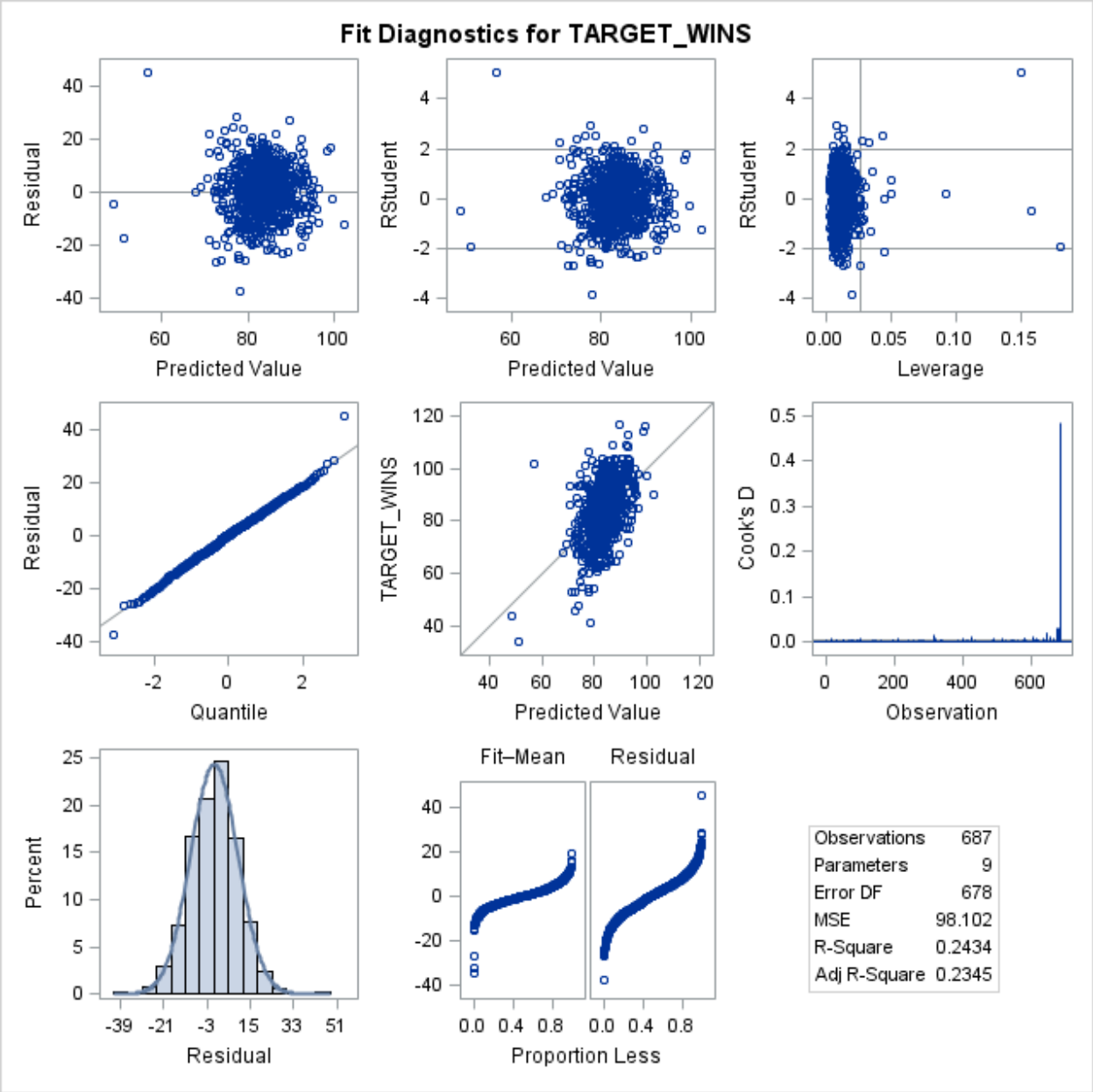
DIAGNOSTIC/RESIDUAL PLOTS

Space does not permit examination of all five models, but two provide good exemplars. The Category 1 model had excellent performance, and the plots are as follows:




As the preceding diagnostics show, no major concerns exist. There are a couple of outliers and a few high leverage points, but, in general, the results are promising.

The poorest performer, the Category 3 model, also had good looking residual plots as follows:




One very significant outlier exists, but there are fewer leverage data points, and the residual quantile graph looks great.

As an additional check on the models, I ran the five models with PROC GLM, and it produced identical coefficients. Because GLM does not make assumptions about the data and it returned identical regression equations, my confidence in the models was further enhanced.

As a final check, I used the twenty percent of the training data that had been set aside to run validation tests; a positive result on these tests would provide confidence of the model's performance with unknown test data. Squared errors were calculated for single regression, combined, and average models for both the training and validation data as follows:

**Squared Error Validation Results**

| Variable | Min | Max | Mean | Median | N | Miss | SumofSq | Err per N | % of AvgMdl |
|---|---|---|---|---|---|---|---|---|---|
| Single Regression – Training Data | 0.0000 | 4057.0 | 148.5 | 63.5 | 1827 | 0 | 271388 | 148.5 | 61.0% |
| Single Regression – Validation Data | 0.0000 | 2629.8 | 158.5 | 68.9 | 448 | 0 | 70988 | 158.5 | 62.7% |
| Four Bucket Model – Training Data | 0.0003 | 2073.8 | 135.4 | 59.7 | 1827 | 0 | 247348 | 135.4 | 55.6% |
| Four Bucket Model – Validation Data | 0.0023 | 2133.0 | 144.2 | 60.6 | 448 | 0 | 64578 | 144.1 | 57.0% |
| Average Model – Training Data | 0.0000 | 4489.0 | 243.5 | 100.0 | 1827 | 0 | 444836 | 243.5 | 100.0% |
| Average Model – Validation Data | 0.0000 | 4761.0 | 252.7 | 121.0 | 448 | 0 | 113199 | 252.7 | 100.0% |

Note: This is consolidated output from two different runs and the output variable names have been changed to make it more readable.

Calculation of the average model error shows that the validation data differed slightly in distribution from the training data, and the average model did worse than in training. Thus, one would expect the models to perform slightly worse, especially given that they were tuned to the training data. The performance impact was slight (1.4% increased error), and, as expected, the four category suite of models maintained its 5% improvement over the single regression.

CONCLUSION

The validation results suggest that if future data is similar to the development data, the algorithm should be stable and provide similar performance. Many issues were addressed in the creation of this model, but opportunities remain for performance increases. If further performance gains are desired, then I would suggest the following steps be pursued:

1) Use the classification tool to generate a finely honed classification that may be overfit for general use but would establish an upper limit as to how much performance one might expect out of the data.
2) Investigate if there is any method to tag the data from different baseball eras so that algorithms may be tuned to the different eras.
3) Assess whether data imputation for the missing values will give lift to the regressions, and, if it does, impute the missing values with a more sophisticated approach than using the mean.
4) Pursue classification development further. For purposes of this analysis, only four buckets of data were chosen; however, increasing the categories to ten or twenty, provided that the data segmentation makes sense, would tease out additional performance opportunity.

# Appendix

## Extra Credit Remarks

To ensure that extra credit items are fully considered, I thought it might be useful to identify the following areas that may be a consideration for extra credit:

- SAS programming methodologies, including the use of macro variables, macro procedures, and structured programming techniques like the use of include files
  In this regard, I found that leveraging macros across PROC invocation was quite easy. However, I spent many hours attempting to understand how to leverage macros properly with data steps, but found it to be an advanced topic that did not yield easily to my programming background.

- Exploration and analysis of missing data items and use for data segmentation

- Use of WEKA for data exploration and data segmentation

- Confirmation of the models using GLM proc

- Use of segregated validation data and performance checking to test against non-trained data