

**DATA621**

**Homework #1**

*9/25/2022*

Critical Thinking Group 2

William Aiken

Donald Butler

Michael Ippolito

Bharani Nittala

Leticia Salazar

Santiago Torres

For this homework assignment, the goal is to explore, analyze, and model a data set of records representing statistics from baseball teams from the 1871 to 2006. Each record contains the performance statistics of one team for a given year, adjusted for a 162-game season. The data contains a training set and an evaluation set (. We built a multiple linear regression model on the training set to predict the number of wins for the team. Based on the resulting model, we predicted the number of wins for each observation in the evaluation set.

**1. DATA EXPLORATION**

The data set contains a training set (2,276 observations) and an evaluation set (259 observations). Table 1.1 lists the variables included in the data set, a definition for each variable, and the theoretical effect of the variable on the number of wins the team had that season.

Table

Description automatically generated

**Table 1.1 Variable names and definitions.**

Due to the number of fields in this data, we broke the dataset into intuitive sections and explored each section individually.

*1.1 Base Hits by Batter*

The following fields comprise the base hits by batter:

1. TARGET\_WINS - Number of wins
2. TEAM\_BATTING\_H - Base hits by batters (1B,2B,3B,HR)
3. TEAM\_BATTING\_2B - Doubles by batters (2B)
4. TEAM\_BATTING\_3B - Triples by batters (3B)
5. TEAM\_BATTING\_HR - Homeruns by batters (4B)

The means and medians are very similar for the base hit variables, implying little skew to the distributions (Table 1.2). R code producing this and all other tables and figures herein is included as Appendix A.

Table

Description automatically generated

**Table 1.2 Means, medians, and standard deviations of base hits by batters.**

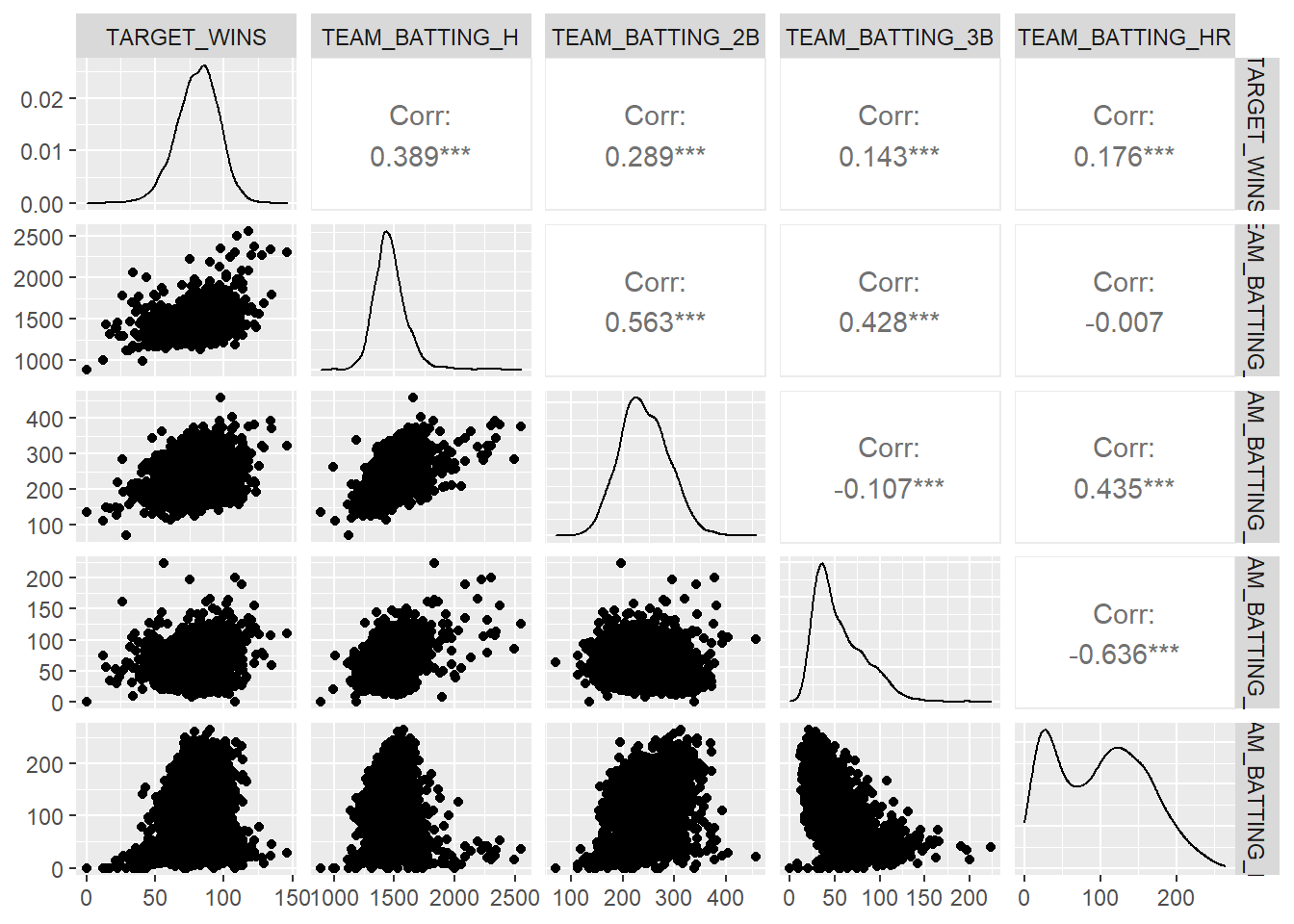
As shown, we see tight distributions except for all base hits by batters (TEAM\_BATTING\_H).

Unsurprisingly, all possible base hits (TEAM\_BATTING\_H) is correlated with winning. As the number of bases achieved by an at-bat increases, the correlation decreases (Figure 1.1).

Chart, box and whisker chart

Description automatically generated

**Figure 1.1 Box plots of base hit variables.**



**Figure 1.2 Correlation plot of base hit variables.**

Interestingly, doubles and triples are correlated with base hits while home runs are not, as shown in Figure 1.2.

*1.2 Batting*

Next, variables associated with batting were investigated:

1. TARGET\_WINS - Number of wins
2. TEAM\_BATTING\_BB - Walks by batters
3. TEAM\_BATTING\_HBP - Batters hit by pitch (get a free base)
4. TEAM\_BATTING\_SO - Strikeouts by batters
5. TEAM\_BASERUN\_SB - Stolen bases
6. TEAM\_BASERUN\_CS - Caught stealing

The measures of central tendency show us that most of these variables have slight skew to their distributions. The stolen bases variable has a large right skew to its distribution (Table 1.3 and Figure 1.3). We are missing values for strikeouts, stolen bases, and caught stealing.

Graphical user interface, table

Description automatically generated

**Table 1.3 Means, medians, and standard deviations of batting variables.**

Chart, box and whisker chart

Description automatically generated

**Figure 1.3 Box plots of batting variables.**

Of all the batting variables, only walks by batter has a correlation to wins (Figure 1.4).

Diagram

Description automatically generated with medium confidence

**Figure 1.4 Correlation plot of batting variables.**

*1.3 Fielding*

1. TARGET\_WINS - Number of wins
2. TEAM\_FIELDING\_E - Errors
3. TEAM\_FIELDING\_DP - Double Plays

The errors variable (TEAM\_FIELDING\_E) has an incredibly right-skewed distribution (Table 1.4 and Figure 1.5). It is also noted that some double plays values are missing.

Graphical user interface, text

Description automatically generated

**Table 1.4 Means, medians, and standard deviations of fielding variables.**

Chart, box and whisker chart

Description automatically generated

**Figure 1.5 Box plots of fielding variables.**

Both fielding variables are negatively correlated with wins (Figure 1.6)

Chart

Description automatically generated

**Figure 1.6 Correlation plot of batting variables.**

*1.4 Pitching*

1. TARGET\_WINS - Number of wins
2. TEAM\_PITCHING\_BB - Walks allowed
3. TEAM\_PITCHING\_H - Hits allowed
4. TEAM\_PITCHING\_HR - Homeruns allowed
5. TEAM\_PITCHING\_SO - Strikeouts by pitchers

As shown in Table 1.5 and Figure 1.7, hits allowed (TEAM\_PITCHING\_H) has a right skew, and some values are missing for strikeouts by pitcher (TEAM\_PITCHING\_SO).

Table

Description automatically generated

**Table 1.5 Means, medians, and standard deviations of pitching variables.**

Chart

Description automatically generated

**Figure 1.7 Box plots of pitching variables.**

The number of hits allowed by pitchers (TEAM\_PITCHING\_H) is negatively correlated with winning (Figure 1.8). Counter-intuitively, home runs allowed is positively correlated with Winning.

Diagram

Description automatically generated

**Figure 1.8 Correlation plot of pitching variables.**

*1.5 Overall Observations*

As shown in the correlation plot for all variables (Figure 1.9), there appear to be several strong correlations between explanatory variables and the target. From an initial inspection, it appears the team should focus on getting players on base through hits or walks. Teams can still win if the pitchers allow homeruns, hits, and walks to the other team.

Chart, scatter chart

Description automatically generated

**Figure 1.9 Correlation plot of all variables.**

The variables with positive correlation to TARGET\_WINS are shown in Table 1.6. Hits by batters tops the list at 0.389.

|  |  |
| --- | --- |
| **Variable** | **Correlation to TARGET\_WINS** |
| TEAM\_BATTING\_H | 0.389 |
| TEAM\_BATTING\_2B | 0.289 |
| TEAM\_BATTING\_BB | 0.233 |
| TEAM\_PITCHING\_HR | 0.189 |
| TEAM\_BATTING\_HR | 0.176 |
| TEAM\_BATTING\_3B | 0.143 |
| TEAM\_PITCHING\_BB | 0.124 |

**Table 1.6 Variables positively correlated with TARGET\_WINS.**

Table 1.7 shows variables that are negatively correlated with TARGET\_WINS. To win more games, it makes sense the team will need to make fewer errors and yield fewer hits to the opposing team.

|  |  |
| --- | --- |
| **Variable** | **Correlation to TARGET\_WINS** |
| TEAM\_FIELDING\_E | -0.176 |
| TEAM\_PITCHING\_H | -0.110 |

**Table 1.7 Variables negatively correlated with TARGET\_WINS.**

**2. DATA PREPARATION**

Table 2.1 shows the percentages of missing values for each variable. As shown, there are two variables missing many observations: TEAM\_BATTING\_HBP is missing over 90% of its values, while TEAM\_BASERUN\_CS is missing just around 30%.

Application

Description automatically generated with low confidence

**Table 2.1 Percentages of observations with missing values.**

Missing values were imputed using R’s multiple imputation by chained equations (mice) package. Imputation was accomplished using predictive mean matching with five multiple imputations over a maximum of five iterations. Figure 2.1 shows the variable summary after imputation, while Figure 2.2 illustrates this information with histograms.

A picture containing calendar

Description automatically generated

**Figure 2.1 Variable summary after imputation.**

Graphical user interface, application

Description automatically generated

**Figure 2.2 Histograms of variables after imputation.**

It was noted that following imputation, several variables exhibited impossibly high values, for example TEAM\_PITCHING\_H, which included values as high as 30,132. To correct for this, any imputed value exceeding three standard deviations was modified to be the median value for that variable. Figure 2.3 illustrates the new variable summary following this cleanup process.

A screenshot of a computer

Description automatically generated with low confidence

**Figure 2.3 Variable summary after cleanup.**

**3. MODEL BUILDING**

After imputing and cleaning the data, the process of model building was begun by fitting wins to the full training data set.

*3.1 Full Model*

By testing all variables in this first model, we are able to see how significant the variables are in our dataset. We will then be able to use this model to serve as a basis for our other models. Model results are given in Table 3.1, and residual plots are illustrated in Figure 3.1.

Table

Description automatically generated

**Table 3.1 Full model results.**

Chart

Description automatically generated with medium confidence

**Figure 3.1. Full model residual plots.**

*3.2 Log Transformation*

As shown in Table 3.2, many variables were skewed. To correct for this, we attempted a log transformation, which distributes skewness into a more “normally” distributed shape. We applied log transformations on highly skewed variables (less than -1 or greater than 1).

Background pattern, table

Description automatically generated

**Table 3.2. Variable skewness.**

It is noted that this model was not successful compared to the first model. There weren’t any significant changes between the two models, and we therefore discarded the second model. The results of the log-transformed model are given in Table 3.3. Residuals are plotted in Figure 3.2.

Table

Description automatically generated

**Table 3.3. Log-transformed model results.**

A picture containing schematic

Description automatically generated

**Figure 3.2. Log-transformed model residual plots.**

*3.3 Statistically Significant Variables*

For the third model, we focused on statistically significant variables, chosen primarily from their R output during the data exploration phase. Table 3.4

Table

Description automatically generated

**Table 3.4. Model results using statistically significant variables.**

*3.4 Backwards Elimination*

For this model, variables that are not statistically significant were removed to generate a best-fit model. Table 3.5 includes the results of this model, and residuals are plotted in Figure 3.3.

Table

Description automatically generated

**Table 3.5. Backwards elimination model results.**

Diagram

Description automatically generated

**Figure 3.3. Backwards-elimination model residual plots.**

*3.5 Power Transform*

Using a power model may be more effective considering each independent variable doesn’t appear to have a truly linear relationship with wins. Here we create a model using a cubit for each independent variable. Partial results are tabulated in Table 3.6; see Appendix A for full results. Residual plots are given in Figure 3.

Graphical user interface, application

Description automatically generated

**Table 3.6. Power transform results.**

Diagram, schematic

Description automatically generated

**Figure 3.4 Log transformation residual plots.**

*3.6 Power Transformation with Reverse Elimination*

Using reverse elimination on model 5, we removed variables with p-values higher than 0.05. Partial results are shown in Table 3.7, and residual plots are given in Figure 3.5.

Graphical user interface, application

Description automatically generated

**Table 3.7. Log transformation with reverse elimination results.**

Chart, scatter chart

Description automatically generated

**Figure 3.5. Log transformation residual plots.**

*3.7 Automated Reverse Elimination*

Based on residual analysis, all six models produced residuals that are heteroschedastic and not normally distributed. Therefore, we tried a different approach and ran the model without imputation and without transformation. We performed backwards elimination using an automated function to generate a new function call based on removing the parameter with the highest p-value (see Appendix A for R code and full model results). The final model output is given in Table 3.7, with residual plots shown in Figure 3.6

Table

Description automatically generated with medium confidence

**Table 3.7. Log transformation with reverse elimination results.**

Scatter chart

Description automatically generated

**Figure 3.6. Automated reverse elimination residual plots.**

**4. MODEL SELECTION**

Model selection was based on a number of factors, including verifying the validity of the model by checking for homoschedasticity of residuals, verifying that the residuals are normally distributed, and verifying linearity (parameters were assumed to be independent and not colinear).

To aid in residual analysis, we used the Breusch-Pagan test to check for homoschedasticity of residuals, along with the Shapiro test to verify they were normally distributed. We also generated plots to visually examine the residuals.

Based on residual analysis, none of the first six models exhibited residuals that were homoschedastic of residuals, nor were the residuals normally distributed. Therefore, the validity of the models couldn’t be ascertained. As a result, we ran a separate set of models programmatically to perform reverse elimination. This yielded a model that produced homoschedastic residuals that were normally distributed.

This model performed better than the previous models, with a higher adjusted R-squared value of 0.511 and lower mean squared error of 69.2. These values indicate a moderate linear relationship between the explanatory and response variables. However, it was noted that only 191 observations remained after specifying that the model remove NaN values. Therefore, despite the better performance, we selected model 6. While this model exhibited heteroschedastic, non-normal residuals, it performed the best in terms of R-squared and mean squared errors, while using the most robust dataset.

To evaluate whether the complexity of model 6 was warranted, we compared it to that of the first model, we an analysis of variance (ANOVA) test between the two models, the results of which are shown in Figure 4.1.

Text, letter

Description automatically generated

**Figure 4.1. ANOVA results.**

As indicated by the low P-value of 2.2e-16, there is enough evidence to reject the null hypothesis that the means are the same. Therefore, we must accept that the means are statistically different. This confirms that there is enough justification in adding complexity to the model. We’ll therefore select model 6. The coefficients from this model follow on Table 4.1

Text, table

Description automatically generated

**Table 4.1. Final model coefficients.**

*Predictions on Evaluation Dataset*

Like the training data, the evaluation dataset included a number of NaN values. Therefore, we imputed data using the same method as with the training set. Then, using the coefficients produced by model 6, we ran the predictions against the 259 observations in the evaluation dataset. Table 4.2 shows the first ten observations of the evaluation dataset including the number of predicted wins.

Table

Description automatically generated

**Table 4.2. First ten observations of evaluation set with predicted wins.**

A full listing of predictions is included in Appendix A.