DATA621 Homework 2 - Predicting Wins in Professional Baseball

Erik Nylander

February 21, 2016

1 Data Exploration

In this analysis we will be using a data set of baseball records from 1871 to 2006 . Each of the 2276 rows of data represents a single teams statistics for the season and the statistics have been adjusted to match the performance for a 162 game season. The data also consists of an evaluation data set that contains 259 teams data with the number of wins removed. The data set consists of 17 different variables with different theoretical effects on the teams overall wins.

1.1 Explanation of the Variables

- Target and Identification Variables
  - INDEX - Identification Variable
  - TARGET_WINS - Number of wins, this is our target variable.
- Positive Impact on Wins
  - TEAM_BATTING_H - Base Hits by batters (1B, 2B, 3B, HR)
  - TEAM_BATTING_2B - Doubles by batters (2B)
  - TEAM_BATTING_3B - Triples by batters (3B)
  - TEAM_BATTING_HR - Home runs by batters (HR)
  - TEAM_BATTING_BB - Walks by batters
  - TEAM_BATTING_HBP - Batters hit by pitch, data is missing for 2085 of the teams and we will be dropping this variable.
  - TEAM_BASERUN_SB - Stolen Bases, there are 131 missing values that we will fix.
  - TEAM_FIELDING_DB - Double Plays, there are 286 missing values that we will fix.
  - TEAM_PITCHING_SO - Strikeouts by the teams pitchers, there are 102 missing values that we will fix.
- Negative Impact on Wins
  - TEAM_BATTING_SO - Strikeouts by the teams batters, there are 102 missing values that we will fix.
  - TEAM_BASERUN_CS - Number of times the teams base runners were caught stealing a base, There are 772 missing values in this data and we will be dropping this variable.
  - TEAM_FIELDING_E - Fielding errors committed by the team.
  - TEAM_PITCHING_BB - Walks allowed by the teams pitchers.
  - TEAM_PITCHING_H - Hits allowed by the teams pitchers.
  - TEAM_PITCHING_HR - Home runs allowed by the teams pitchers.

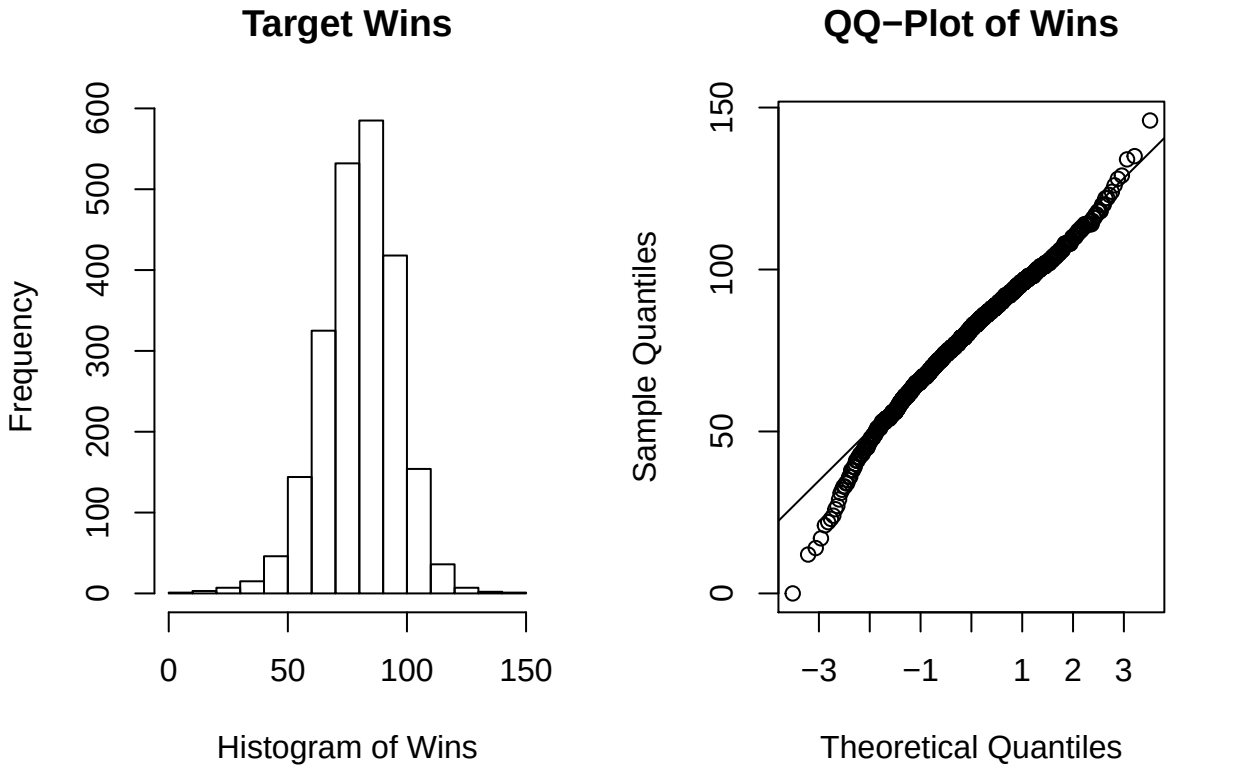
1.2 Observations of Missing Values

Looking at our variables we note that there are a number of missing observations in the data set. Most notably the number of batters hit by a pitch (TEAM_BATTING_HPB) is missing more than 91% of its observations and the number of base runners that were caught stealing (TEAM_BASERUN_CS) are

missing more than 33% of its observations. Given the high percentage of missing values we will be electing to drop these two variables. We also note that the following variables are missing no more than 6% of their observations: Strike Outs (TEAM_BATTING_SO), Stolen Bases (TEAM_BASERUN_SB), Strike Outs by Pitchers (TEAM_PITCHING_SO), and finally the number of Double Plays Turned by the team (TEAM_FIELDING_DP) is missing less than 13% of it's observations.

1.3 Shape of the Data

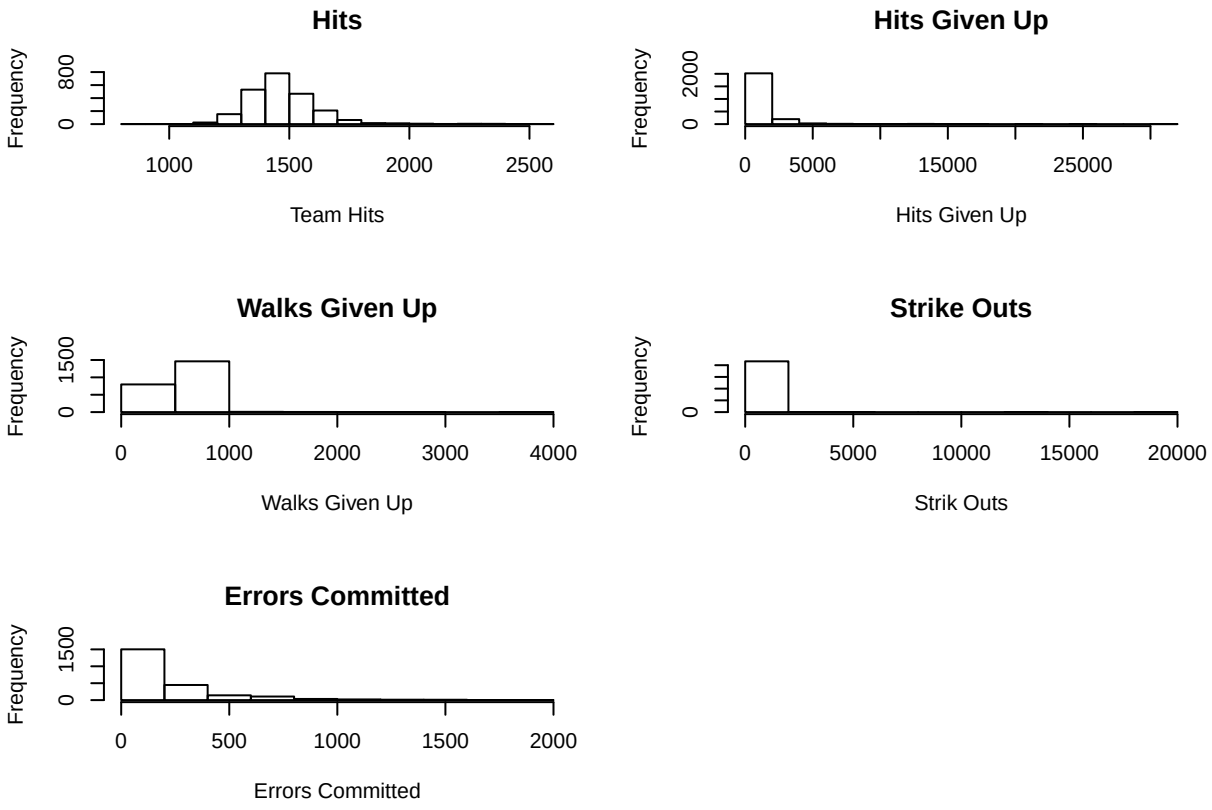
First lets take look at our target variable, the number of wins in a given season (TARGET_WINS). In the figure below we look at the histogram of our wins and the QQ-Plot. We note that there is some skew to the left side of the data and there is one observation of a season with no wins along with many observations of seasons with more than 120 wins. After doing some research, https://en.wikipedia.org/wiki/List_of_worst_Major_League_Baseball_season_records, we note that there were no professional baseball teams with no wins so we will be eliminating this data point. Even the teams with less than 20 and more than 120 wins appear to based on the projection of teams pre 1900 to a 162 game season.



1.3 Odd Features in the Data

There are also a number of the explanatory variables that have some odd features. In the figure below we look at the Number of hits that a team has, the number of hits that they have given up, the number of walks given up, the number of strike outs by the team's pitchers, and the number of errors committed. We notice that in all five of these graphs we have data points that fail to make sense. We can see that on average a team gets 5500 at bats a season (https://www.teamrankings.com/mlb/stat/at-bats-per-game), this makes it impossible for us to get some of the values seen in the data. Given that these values do not make sense in game of baseball will simply remove these observations as they

are likely due to miss-entered data or the result of the extrapolation to a 162 game season.



1.4 Correlations in the Data

Finally we will be looking at the correlations in our variables. In Table 1 we see the following correlations of the variables to the target wins. We do notice that some of these correlations are in an unusual direction and are counter-intuitive. We would not expect that the number of hits allowed would have a positive correlation with wins and that the stolen bases would have a positive correlation with wins. These finding are counter to what we were given as the theory for the problem and lead us to believe that it may be difficult to construct a good model for the number of wins that a team achieves in a season. We also note that all of the correlations have a relatively small value, which will further lead to issues in building the model. There are also a number of other variables that have high correlation values and we will be monitoring this throughout the analysis.

Table 1: Correlations between TARGET_WINS and the Predictor Variables

Variable	Correlation
TEAM_BATTING_H	0.352
TEAM_BATTING_2B	0.213
TEAM_BATTING_3B	0.123
TEAM_BATTING_HR	0.22
TEAM_BATTING_BB	0.302
TEAM_BATTING_SO	-0.059
TEAM_BASERUN_SB	-0.121
TEAM_PITCHING_H	0.221
TEAM_PITCHING_HR	0.219

Variable	Correlation
TEAM_PITCHING_BB	0.271
TEAM_PITCHING_SO	-0.066
TEAM_FIEDLING_E	-0.185
TEAM_FIELDING_DP	-0.036

2 Data Preperation

Our primary goal in the preparation of the data for analysis is to fix some of the errors that we discovered in the exploration of the data. We will need to remove observations where the value makes no sense in the history of professional baseball. We were able to use Many of our figures were also gathered from: http://www.baseball-almanac.com to find the historical information that we will be comparing our data to. We will also attempt to re-create the Runs Created formula that was first utilized by Bill James. While we are missing some of the values, the construction of this variable may allow us to simplify our model. Finally we will be replacing our missing values with the medians of their respective data sets. This will allow us to use all of the remaining data for the analysis and gives us the opportunity to work with a missing data replacement technique.

2.1 Fixing Variables

After some considerable research we will take the following steps to clean up our data and remove observations that do not make sense for the game of baseball. Lets look at each of the variables and the work that we have done to them.

Target Wins (TARGET_WINS) - Our target variable while in great shape as can seen in the first figure does have an issue. There is one observation that has 0 wins. This has not happened in the history of professional sports so we will remove this observation.

Team Hits (TEAM_BATTING_H) - This variable has a number of teams that have hit totals that do jive with the reality of baseball. One of the teams with the best hit totals ever was the 1894 Philadelphia Phillies that had 1,734 hits in a 128 game season. This would work out to a 2,192 hits in a 162 game season. We will be dropping any observations above this point. Why they may be projected values they do make sense for our analysis.

Team Stolen Bases - (TEAM_BASERUN_SB) - The main issue that we have here is the observations of teams that have no stolen bases for the year. The lowest recorded was 13 by Washington in 1957 so we will remove all observations below this value.

Hits Given Up - (TEAM_PITCHING_H) - Where to start, we have a team that gave up more than 20,000 hits in a season and even after the filtering of teams for other issues we still have 17 teams that have given up more than 4,000 hits. Even at 4,000 hits that a team that would average giving up 25 hits a game. since the maximum ever in a single game was 33, we can safely remove these observations from the data set.

Home Runs Given Up - (TEAM_PITCHING_HR) - The only issue in this data set is that there are a few teams that gave up zero home runs. The best achieved in the major league was 4 home runs given up by the 1902 Pittsburgh Pirates in a 140 game season. Projecting this to 162 games gives us a best ever figure of 5 home runs in a season. We will use this as our cutoff for the data.

After performing the above steps to prune the data to only contain values that have been seen, or make sense from projections, in professional baseball we have managed to also clear up some of the issues with absurd values in our other variables. It is difficult to know if the issues with the data were mis-entered or the result of projecting the values in a linear fashion when the underlying structure was not linear. We now have a data set that consists of 2114 observations and 15 predictor variables.

2.2 Replacing Missing Values

Now that we have cleaned up our data set we see that there are three variables that still have some missing values. The TEAM_BATTING_SO, TEAM_PITCHING_SO, and TEAM_FIELDING_DP have 100, 100, and 180 missing values respectively. In this report we will be using the median value of each variable respectively to fill in these missing values.

2.3 Creating the Runs Created Statistic

Finally we will be attempting to re-create the runs created variable that was first developed and implemented by Bill James. The runs creates value is calculated using the following formula:

$$\text{RC} = \frac{(\text{Hits} + \text{Walks})(\text{Total Bases})}{(\text{At Bats}) + (\text{Walks})}$$

Where Total Bases is calculated by:

$$\text{Total Bases} = 1\text{B} + 2(2\text{B}) + 3(3\text{B}) + 4(4\text{B})$$

We will calculate At Bats as:

$$\text{At Bats} = \text{Hits} + \text{Walks} + \text{Strike Outs}$$

While this at bats calculation is missing sacrifice flies and other plate appearances that do not fall into hits, walks, or strike outs it will hopefully give us a fairly constant way to measure offensive production across our data set. The Figure below shows the histogram and QQ-Plots for the runs created Variable that results from this calculation.

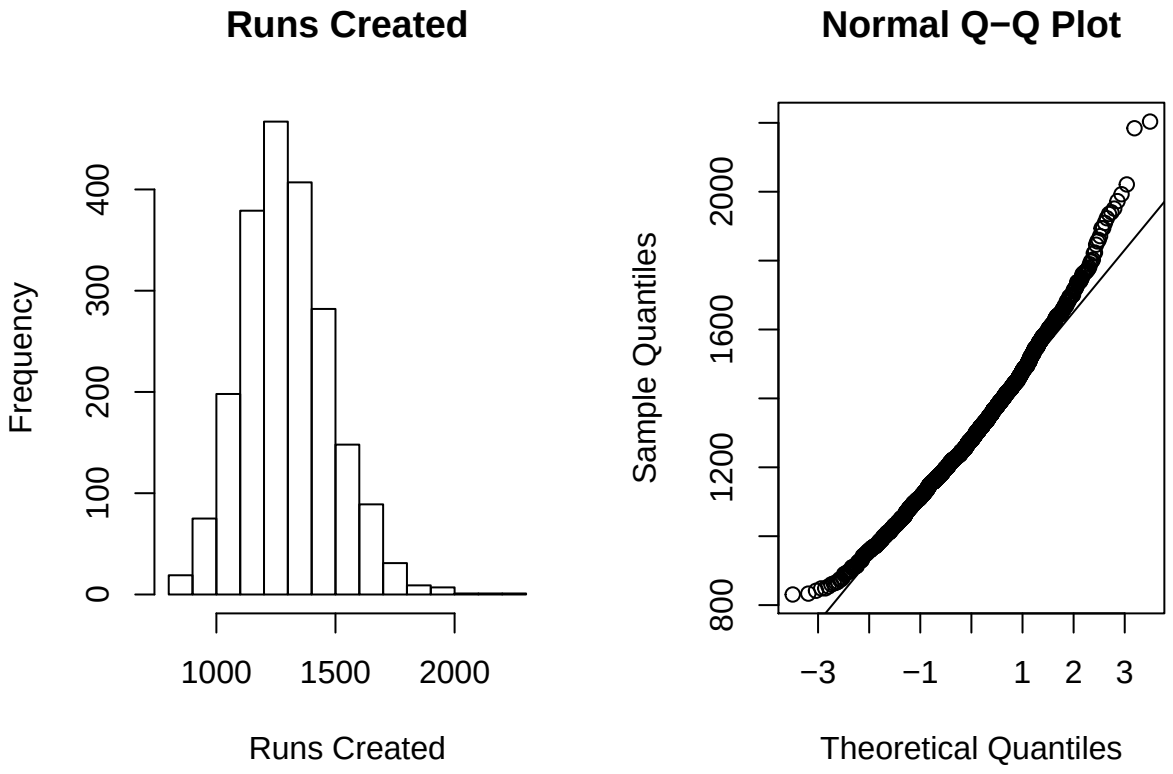


Table 2 shows the runs created (rc) statistic has the following correlation values for wins, hits, doubles, triples, and home runs. We expect there to be a relatively high correlation given that the runs created

variable is constructed from these variables. We do note that the correlation between wins and the runs created variable is the highest correlation that we have seen in the data set. This may bode well for this variable being a predictor in the data set.

Table 2: Correlation Values for the Runs Created Statistic

Variable	Correlation
TARGET_WINS	0.367
TEAM_BATTING_H	0.948
TEAM_BATTING_2B	0.633
TEAM_BATTING_3B	0.35
TEAM_BATTING_HR	0.212
TEAM_BATTING_BB	0.19

3 Building the Models

Now that we have prepared the data we will begin building our models. For this analysis we will construct three different models using our modified data set. The first model will be constructed using a forward selection method and the second model will be constructed using backwards selection. From our previous coursework in statistics we expect that these two models will end up with different sets of variables and it will be interesting to see the outcome. While this selection process is controversial it will be interesting to see the structure of the models that are generated. Our final model will include variables that we think may have an impact on wins and will be based on our best guess.

3.1 Forward Selection

In the forward selection process we will fit all possible models with a single variable, we will then select the variable with the smallest p-value to include in the model. Our model will be complete when we can no longer add variable with a significant p-value, $\alpha = .05$. To expedite the process we will use the *mixlm()* package in R and report the final model, the code for the process can be found in the Appendix. Table 3 contains the output for the forward selection model.

Table 3: Linear Regression using Forward Selection $\alpha = 0.05$

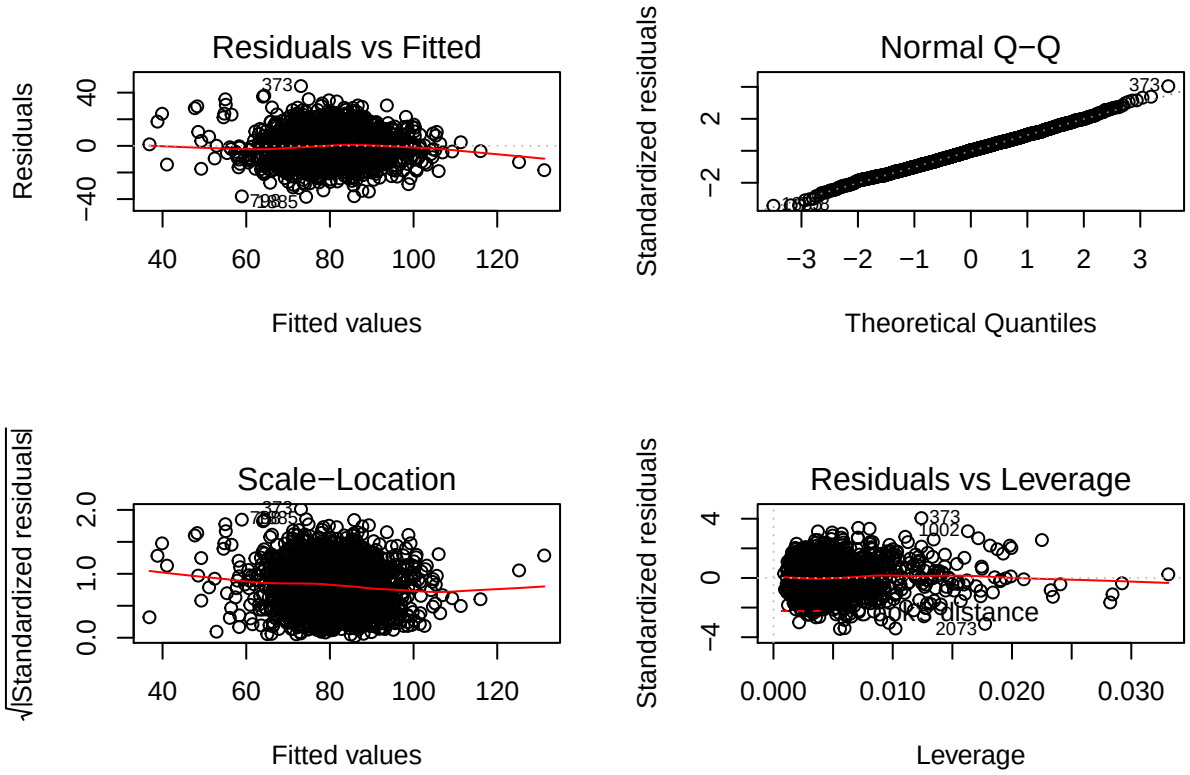
Variable	Coefficient
Intercept	45.475
rc	0.034
TEAM_BATTING_BB	0.038
TEAM_FIELDING_DP	-0.11
TEAM_FIELDING_E	-0.078
TEAM_BASERUN_SB	0.077
TEAM_BATTING_HR	0.021
TEAM_BATTING_2B	-0.063
TEAM_BATTING_3B	0.112

Given the construction of the model all of the predictor variables are significant (p-value < 0.05). The model has a residual standard error (RSE) of 11.2 (2105 degrees of freedom) with an R^2 value of 0.3817. The coefficients of the model show some interesting information. We first see that the number of runs created (rc), triples (TEAM_BATTING_3B), home runs (TEAM_BATTING_HR), walks (TEAM_BATTING_BB), stolen bases (TEAM_BASERUN_SB) all have a positive impact on the number of wins. For the most part,

when the other variables are held constant, an increase of one in these variables is related to an increase of 3 to 8 wins per hundred games played. The largest impact on wins is caused by an increase in the number of triples. This is quite possibly the most difficult hit to get in a game and even the best teams do not average a triple a game.

Next we see that the number of errors made (TEAM_FIELDING_E), double plays turned (TEAM_FIELDING_DP), and doubles hit (TEAM_BATTING_2B) all have a similar negative impact on wins to the impact of the positive variables. There are some odd effects here. The first is that the numbers of double plays turned should not have a negative impact on wins. There is a potential explanation for this, a team can only turn double plays if they allow their opponents to a high number of base runners. We can easily believe that allowing a high number of base runners would lead to a team giving up a large number of runs and losing games. The second variable is less easy to explain. We have no idea why a team hitting an increased number of doubles would lead to losses. I would want to do more research on this data to attempt to explain why this would happen.

In the figure below we can see the diagnostic plots for the model. We note that the diagnostic plots look fairly good with some potential for inconsistent variance in residuals give the spread of points to the left of the graph. We also note that there are a few high leverage points in the data.



3.2 Backward Selection

The second model that we will be constructing we will use a backward selection process. This method requires that we fit the full model and then remove the largest non-significant p-value predictor. For this model we will again use the *mixlm()* package and use a significance level of $\alpha = 0.05$. Table 4 contains the output for the backward selection model.

Table 4: Linear Regression using Backward Selection $\alpha = 0.05$

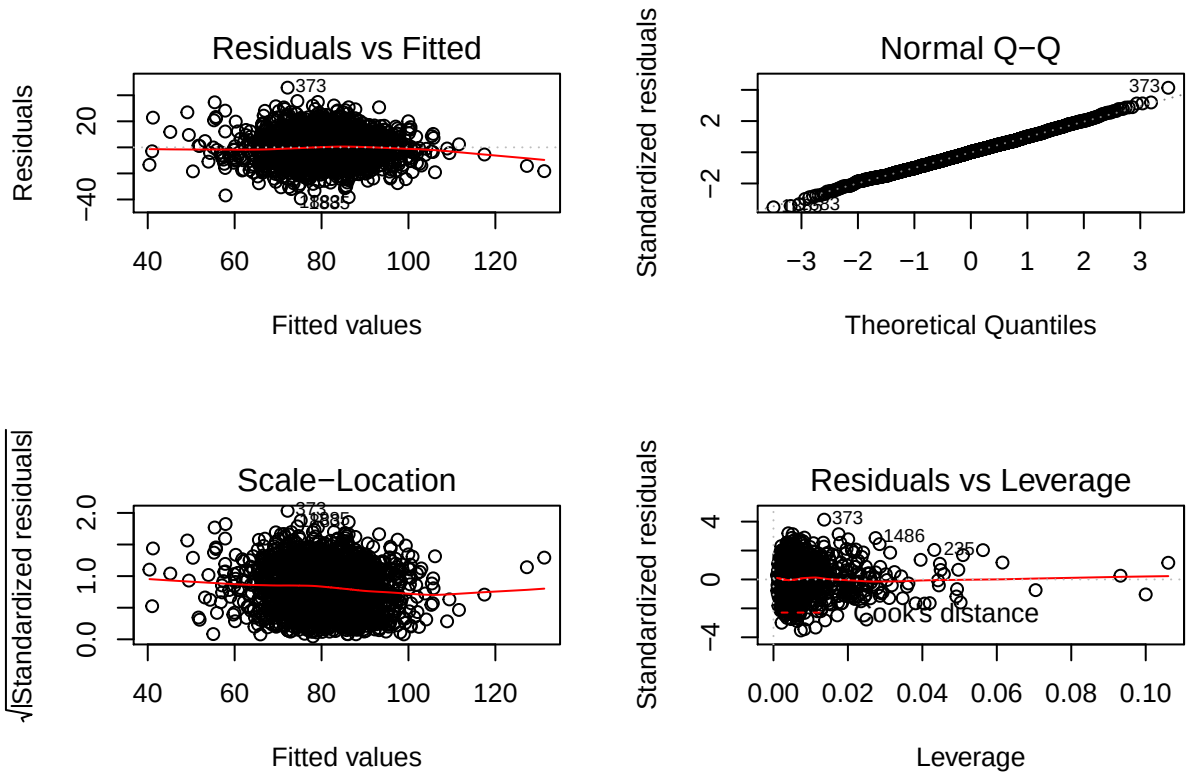
Variable	Coefficient
Intercept	43.64
TEAM_BATTING_H	0.057
TEAM_BATTING_3B	0.247
TEAM_BATTING_HR	0.201
TEAM_BATTING_BB	0.122
TEAM_BATTING_SO	-0.042
TEAM_BASERUN_SB	0.076
TEAM_PITCHING_H	0.033
TEAM_PITCHING_BB	-0.094
TEAM_FIELDING_E	-0.079
TEAM_FIELDING_DP	-0.107
rc	-0.7

Given the construction of the model all of the predictor variables are significant (p-value < 0.05). The model has a residual standard error (RSE) of 11.15 (2104 degrees of freedom) with an R^2 value of 0.3887. This is slightly better than the forward selection model above.

The coefficients of our second model also show some interesting information. We first see that the number of hits (TEAM_BATTING_H), number of triples (TEAM_BATTING_3B), home runs (TEAM_BATTING_HR), walks (TEAM_BATTING_BB), stolen bases (TEAM_BASERUN_SB), hits given up (TEAM_PITCHING_H) all have a positive impact on the number of wins. We notice that a number of these variables have larger coefficients then were seen in the previous model. This could indicate some colinearity in the models. We also have one unusual coefficient, in the hits given up. It appears that teams that give up more hits win more often. This could be explained if there is a corresponding decrease in the number of home runs and other multi-base hits while keeping the number of strikeouts and walks consistent.

Next we see that the number of strike outs (TEAM_BATTING_SO), double plays turned (TEAM_FIELDING_DP), errors (TEAM_FIELDING_E), walks given up (TEAM_PITCHING_BB), and runs created (rc) all have a similar negative impact on wins to the impact of the positive variables. Once again we see the odd effect of teams turning double plays and runs created have a negative impact on the wins. The biggest unknown is why the sign switches on the runs created variable in the first model it has a positive impact and in the second model has a negative impact. We would want to do more research with these two variables before moving forward with model in a production situation.

In the figure below we can see the diagnostic plots for the model. We note that there are a few high leverage points in the data.



3.3 Hand Picked Variables

For our final model we will hand pick some of our variables based on our intuition about what might effect the number of wins. The first variable that we will pick is the runs created (rc) variable that we constructed. It had different effects in the automatically picked models so we would like to explore it's effect in a third model. The other offensive variable that we will take is stolen bases (TEAM_BASERUN_SB) given that this variable should indicate an aggressive team that manufactures runs. Continuing with expected positive variables we will take the defensive variable strike outs by pitchers (TEAM_PITCHING_SO). For negative impact variables we will take errors (TEAM_FIELDING_E), hits given up (TEAM_PITCHING_H), and home runs given up pitchers (TEAM_PITCHING_HR). These three variables should give us an idea of the teams ability to limit the number of base runners and potentially the runs. Table 5 contains the results of the model run.

Table 5: Linear Regression Selection by Hand (@ significant at $\alpha = 0.05$)

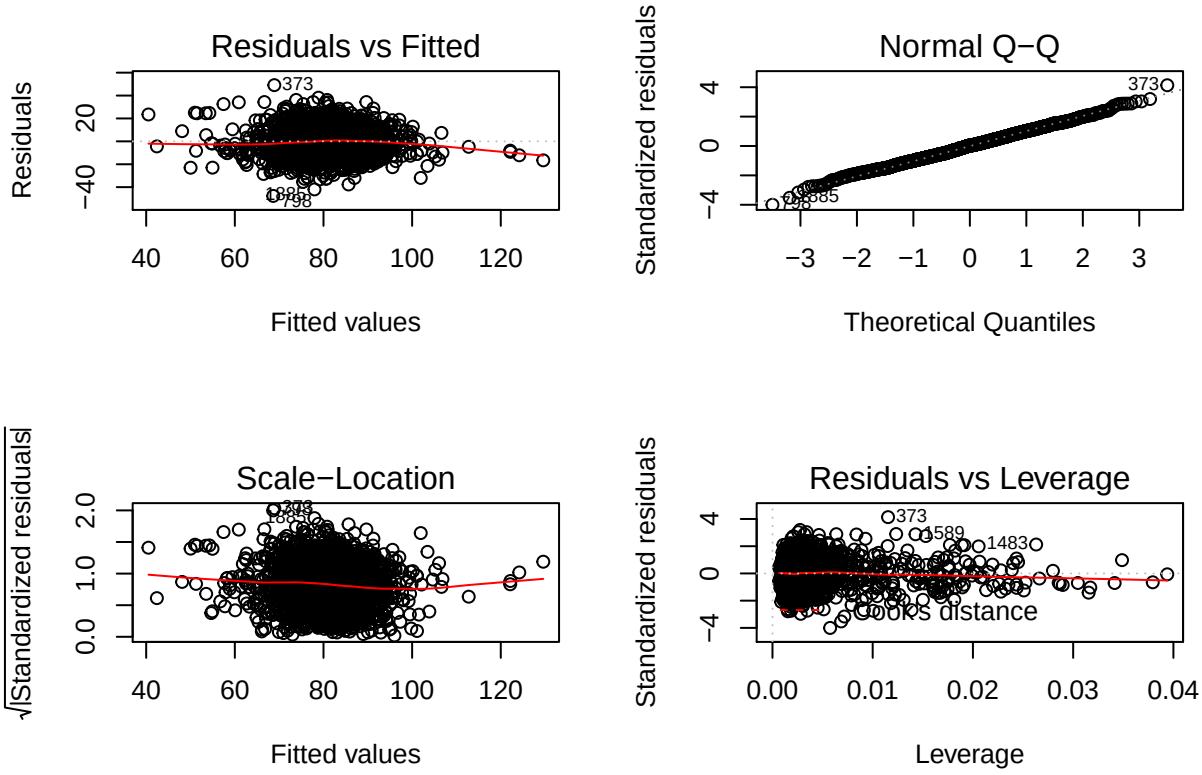
Variable	Coefficient
Intercept@	57.636
rc	0.009
TEAM_BASERUN_SB@	0.092
TEAM_BATTING_SO	0.017
TEAM_PITCHING_SO@	-0.033
TEAM_PITCHING_H@	0.013
TEAM_PITCHING_HR@	0.051
TEAM_FIELDING_E@	-0.07

All of our predictor variables except for the runs created (rc) and strike outs (TEAM_BATTING_SO) are significant (p-value < 0.05). The model has a residual standard error (RSE) of 11.94 (2106 degrees of freedom) with an R^2 value of 0.2966. This model is not as good as the two that were fit automatically.

The coefficients of our second model also show some interesting information. We first see that the stolen bases (TEAM_BASERUN_SB), home runs (TEAM_BATTING_HR), hits given up (TEAM_PITCHING_H) all have a positive impact on the number of wins. We also have an unusual coefficient, in the hits given up. It appears that teams that give up more hits win more often. This could be explained if there is a corresponding decrease in the number of home runs and other multi-base hits while keeping the number of strikeouts and walks consistent.

Next we see that the number of strike outs (TEAM_BATTING_SO) and errors (TEAM_FIELDING_E), have a similar negative impact on wins to the impact of the positive variables. Once again we see the sign switch on the runs created variable although this time it is not significant. We would want to do more research with these two variables before moving forward with model in a production situation.

In the figure below we can see the diagnostic plots for the model. We note that there are a few high leverage points in the data.



4 Selecting a Model

Finally we need to select a final model and make our predictions. To decide on the best model we will be looking at the mean square error, the residual standard error and adjusted R^2 values and validity of the model assumptions. These are calculated by using our model to predict the values in the training data set and comparing them to the actual values. Our best model ended up being the model chosen by forward selection as can bee seen in Table 3.

Using this full model we get the following values; The Adjusted R^2 = 0.3794, MSE =124.93, RSE = 11.2. Using this model we also get an Adjusted R^2 value of 0.3817 and an F-statistic of 162.5 with a p-value of ~0. The backward fit model actually had slightly better values in all of these statistics: Adjusted R^2 = 0.3855, MSE =123.53, RSE = 11.15. However, this model suffers multi-coliniarity with a number of variables having a variance inflation factor (VIF) much greater than 10. Our forward fit model performs almost as well and does not suffer from the multi-coliniarity issue.

As we have seen in the figures above there is no discernible pattern to the residuals and the variance of the residuals by observation appear to be consistent. The biggest problem that we see with the models is the relatively low values for the coefficients and the fact that a number of the coefficients change sign depending on how the model is built. Overall this leads me to believe that we may not be able to get good predictions of the number of wins that a team will achieve from the data or models that we have. This being said we will go ahead and predict the number of wins for the test data set even though we are unsure of the values. Table 6 shows the first 10 predicted wins and Table 7 shows the last 10 predicted wins.

Table 6: First 10 Predicted Wins

1	2	3	4	5	6	7	8	9	10
64	68	73	84	35	41	76	70	75	70

Table 7: Last 10 Predicted Wins

250	251	252	253	254	255	256	257	258	259
84	79	8	90	-9	71	80	85	88	71

5 Concluding Remarks

The most note worthy thing that we can see in the data is also adds to our lack of trust in this model. We can see that team 254 is predicted to win -9 games. This does not make sense and indicates that a linear model may not be the best predictor for this type of data even though there were no indications in the residuals that there may be an underlying pattern. The problems that we have seen in this analysis indicate that predicting the number of wins is a difficult process and hint at many different underlying factors that are not taken into account in this data.

6 Appendix

In this section, we provide the R code used in the analysis.

```
library(dplyr)
bsbl <- read.csv("DATA621/moneyball-training-data.csv")
bsbl_test <- read.csv("DATA621/moneyball-evaluation-data.csv")


# Generating Summary statistics for the variables.
summary(bsbl)
summary(bsbl_test)

# Removing the two variables with a larger than 30% of the values missing.
bt <- bsbl %>%
  select(-TEAM_BATTING_HBP, -TEAM_BASERUN_CS)
```

```r
bt_test <- bsbl_test %>%
  select(-TEAM_BATTING_HBP, -TEAM_BASERUN_CS)


par(mfrow = c(1,2))
hist(bsbl$TARGET_WINS, main = "Target Wins", xlab = "Target Wins")
qqnorm(bsbl$TARGET_WINS, main = "QQ-Plot of Wins")
qqline(bsbl$TARGET_WINS)


# Graph of the highly skewed data that makes no sense.
# https://www.teamrankings.com/mlb/stat/at-bats-per-game
par(mfrow = c(3,2))
hist(bsbl$TEAM_BATTING_H, main = "Hits", xlab = "Team Hits")
hist(bsbl$TEAM_PITCHING_H, main = "Hits Given Up", xlab = "Hits Given Up")
hist(bsbl$TEAM_PITCHING_BB, main = "Walks Given Up", xlab = "Walks Given Up")
hist(bsbl$TEAM_PITCHING_SO, main = "Strike Outs", xlab = "Strik Outs")
hist(bsbl$TEAM_FIELDING_E, main = "Errors Committed", xlab = "Errors Committed")

# Checking the correlations in the data set
cor(bt, use="complete.obs")

# https://en.wikipedia.org/wiki/List_of_worst_Major_League_Baseball_season_records
# Filtering the data set to remove the 0 win season, we will leave in the 120+ win seasons
# these we caused by the extension of the shorter seasons to 162 games.
bt <- bsbl %>%
  select(-TEAM_BATTING_HBP, -TEAM_BASERUN_CS) %>% # Removing the two variables with a larger than 30% o
  filter(TARGET_WINS > 0) %>% # Filtering Wins
  filter(TEAM_BATTING_H < 2192) %>% # Filtering Hits
  filter(TEAM_BASERUN_SB >= 13) %>% # Filter stolen bases under 13
  filter(TEAM_PITCHING_H <= 4000) %>% # Filter the teams that have given up more than 4000 hits.
  filter(TEAM_PITCHING_HR >= 5) # Filter Teams with less than 4 home runs

# Letslook at a summary of the new data set
summary(bt)

# Filling in the missing values for the Strike out and double play data
bt$TEAM_BATTING_SO[is.na(bt$TEAM_BATTING_SO)] <- median(bt$TEAM_BATTING_SO, na.rm = TRUE)
bt$TEAM_PITCHING_SO[is.na(bt$TEAM_PITCHING_SO)] <- median(bt$TEAM_PITCHING_SO, na.rm = TRUE)
bt$TEAM_FIELDING_DP[is.na(bt$TEAM_FIELDING_DP)] <- median(bt$TEAM_FIELDING_DP, na.rm = TRUE)

# Filling in the missing data in the test set
summary(bt_test)
bt_test$TEAM_BATTING_SO[is.na(bt_test$TEAM_BATTING_SO)] <-
  median(bt_test$TEAM_BATTING_SO, na.rm = TRUE)
bt_test$TEAM_BASERUN_SB[is.na(bt_test$TEAM_BASERUN_SB)] <-
  median(bt_test$TEAM_BASERUN_SB, na.rm = TRUE)
bt_test$TEAM_PITCHING_SO[is.na(bt_test$TEAM_PITCHING_SO)] <-
  median(bt_test$TEAM_PITCHING_SO, na.rm = TRUE)
bt_test$TEAM_FIELDING_DP[is.na(bt_test$TEAM_FIELDING_DP)] <-
  median(bt_test$TEAM_FIELDING_DP, na.rm = TRUE)

# Constructing the runs created variable
```

```r
ab <- bt$TEAM_BATTING_H + bt$TEAM_BATTING_BB + bt$TEAM_BATTING_SO
singles <- bt$TEAM_BATTING_H - (bt$TEAM_BATTING_2B + bt$TEAM_BATTING_3B + bt$TEAM_BATTING_HR)
total_bases <- singles + 2*bt$TEAM_BATTING_2B + 3*bt$TEAM_BATTING_3B + 4*bt$TEAM_BATTING_HR
bt$rc <- ((bt$TEAM_BATTING_H+bt$TEAM_BATTING_BB)*total_bases)/(ab + bt$TEAM_BATTING_BB)
hist(bt$rc, main = "Runs Created", xlab = "Runs Created")
qqnorm(bt$rc)
qqline(bt$rc)

# Constructing runs created for the test data set
ab <- bt_test$TEAM_BATTING_H + bt_test$TEAM_BATTING_BB + bt_test$TEAM_BATTING_SO
singles <- bt_test$TEAM_BATTING_H -
  (bt_test$TEAM_BATTING_2B + bt_test$TEAM_BATTING_3B + bt_test$TEAM_BATTING_HR)
total_bases <- singles + 2*bt_test$TEAM_BATTING_2B +
  3*bt_test$TEAM_BATTING_3B + 4*bt_test$TEAM_BATTING_HR
bt_test$rc <- ((bt_test$TEAM_BATTING_H+bt_test$TEAM_BATTING_BB)*total_bases)/(ab + bt_test$TEAM_BATTING

# Fitting the full model
# The mixlm() package requires that the model be fully defined. The first attemp failed with the
# model defined as lm(TARGET_WINS ~ ., data = bt)
library(car)
fit_tbh <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B +
                TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
                TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_HR +
                TEAM_PITCHING_BB + TEAM_PITCHING_SO + TEAM_FIELDING_E +
                TEAM_FIELDING_DP + rc, data = bt)
vif(fit_tbh)


# Fitting the model with forward selection, alpha level is the sgnificance level.
library(mixlm)
for_fit <- forward(fit_tbh, alpha = .05)
coefficients(for_fit)
summary(for_fit)
plot(for_fit)
vif(for_fit)  #Checking for Collinearity, all less than 10
mean((bt$TARGET_WINS - predict(for_fit, bt))^2)
sd((bt$TARGET_WINS - predict(for_fit, bt))^2)/sqrt(n)

# Fitting the model with backwards selection, alpha level of 0.05
back_fit <- backward(fit_tbh, alpha = 0.05)
coefficients(back_fit)
summary(back_fit)
plot(back_fit)
vif(back_fit)  #Checking for Collinearity, all less than 10
mean((bt$TARGET_WINS - predict(back_fit, bt))^2)
sd((bt$TARGET_WINS - predict(back_fit, bt))^2)/sqrt(n)

# Fitting the hand chosen model.
fit_hand <- lm(TARGET_WINS ~ rc + TEAM_BASERUN_SB + TEAM_BATTING_SO + TEAM_PITCHING_SO +
                TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_FIELDING_E, data = bt)
coefficients(fit_hand)
summary(fit_hand)
plot(fit_hand)
```

```r
vif(fit_hand)  #Checking for Collinearity, all less than 10
mean((bt$TARGET_WINS - predict(fit_hand, bt))^2)
sd((bt$TARGET_WINS - predict(fit_hand, bt))^2)/sqrt(n)

# Using our best model to predict on our test data
test <- predict(for_fit, bt_test) # Predicting the data
head(test)
tail(test)
test
```