Oscar Hernandez
MSDS 411
Fall 2018
Unit 1 – Moneyball

Contents

Introduction

The purpose of this report is to cover all the analysis that was completed for the Moneyball data set. The Moneyball data set includes a variety of baseball statistics for different baseball teams. The statistics have been normalized to cover a 162-game season. Specifically, the data set includes 16 variables such as the amount of home runs a team hit, the amount of hits a team's pitchers allowed and the number of stolen bases a team achieved. This report will cover the Exploratory Data Analysis that was conducted to better understand the data, how the data was prepared, the creation of three different multiple linear regression models and how the "best" model was determined.
This analysis was done with the goal of developing a model that can accurately predict the amount of wins a baseball team should earn based on the various statistics such, as the ones mentioned previously.

Section 1 – Exploratory Data Analysis

Prior to building any model, it is prudent to better understand the data that will be used to train the model. Exploring data is important because we are looking to see the relationship amongst the variables included in our data, check to see if we have erroneous values and understand why we have missing values. As mentioned, our goal is to build a multiple linear regression model that can accurately predict the amount of wins a baseball team should earn based on the variables that are in our data.

Our first step will be to review the distribution of the response variable to see if it follows a normal distribution and if there are outliers. Figure 1 displays the necessary graphics to check the distribution and see if any outliers exist. From the histogram, we can see some positive signs that our response variable – TARGE_WINS – seems to follow a normal distribution. This indicates that we may not necessarily need to make any transformation in the data preparation step. Furthermore, the boxplot does show the presence of outliers. Specifically, we can see that there is a 0 value that appeared in one of the observations which will need to be dealt with because it won't help in building an accurate model.

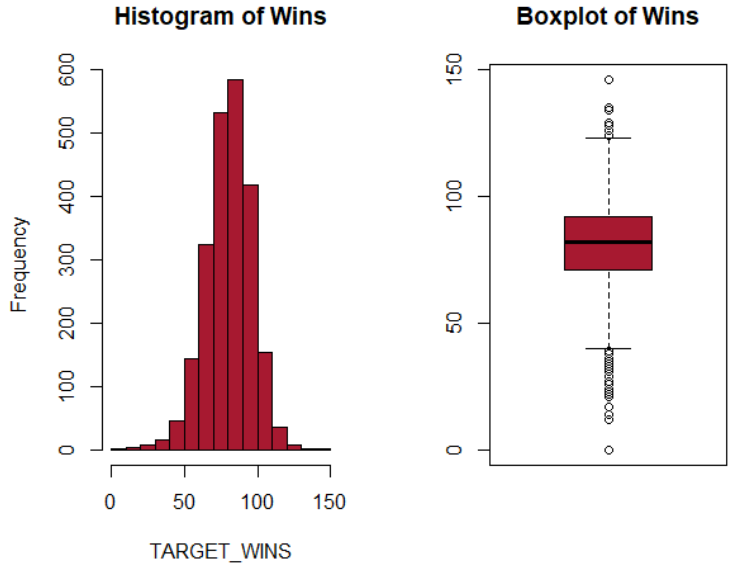

Figure 1: Distribution of TARGET_WINS response variable

The next part of our analysis will include a review of all the variables that are in our data set. Specifically, we want to check the distribution of these variables to see if there are some clearly erroneous values and see how much of our data set has missing values. Figure 2 gives a robust overview of all the variables. The table suggests that we will have to come up with a way to impute for missing values since six of the variables have missing values. Furthermore, there are several variables that have 0 as a minimum value which is considered to be an error given that it is highly unlikely a team would play 162 games without getting at least 1 win or 1 walk for example. On the other hand, we can see there are some erroneous values from the Max column given that it is also highly unlikely that a team would allow over 30,000 hits in a season. Overall, Figure 2 provides us an idea of which variables may need to be inputted.

Variable	Missing	Min	Q1	Median	Mean	Q3	Max
TARGET_WINS	0	0	71	82	80.79	92	146
TEAM_BATTING_H	0	891	1383	1454	1469	1537	2554
TEAM_BATTING_2B	0	69	208	238	241	273	458
TEAM_BATTING_3B	0	0	34	47	55.25	72	223
TEAM_BATTING_HR	0	0	42	102	99.61	147	264
TEAM_BATTING_BB	0	0	451	512	501.6	580	878
TEAM_BATTING_HBP	2085	29	50.5	58	59.36	67	96
TEAM_BATTING_SO	102	0	548	750	735.6	930	1399
TEAM_BASERUN_SB	131	0	66	101	124.8	156	697
TEAM_BASERUN_CS	772	0	38	49	52.8	62	201
TEAM_FIELDING_E	0	65	127	159	246.5	249.2	1898
TEAM_FIELDING_DP	286	52	131	149	146.4	164	228
TEAM_PITCHING_BB	0	0	476	536.5	553	611	3645
TEAM_PITCHING_H	0	1137	1419	1518	1779	1682	30132
TEAM_PITCHING_HR	0	0	50	107	105.7	150	343
TEAM_PITCHING_SO	102	0	615	813.5	817.7	968	19278

Figure 2: Descriptive Statistics

The next part of our analysis is concerned with reviewing the correlation between the response variable and each of the explanatory variable. Correlation is an important statistic to review since it can give us an idea of how two different variables move in conjunction with each other. This relationship is key with regards to building a multiple linear regression model.

Figure 3 is table that displays the correlation between TARGET_WINS and each explanatory variable. We can see that the amount of hits from batters has the highest positive correlation with the response variable and the amount of fielding errors has the highest negative correlation. The table also suggests that most of the batting statistics appear to be positively correlated with the amount of wins a team has which makes sense given that the more runs a team scores, the better chances they have at winning. One interesting correlation is that the amount of walks a team gives up is positive correlated with the response variable. This is a bit odd given that providing more opportunities for the other team to score more runs will likely decrease the chances of winning for the team that is giving up the walks. Overall, this table provides us an idea of which variables we may want to include in our model.

	Correlation Coefficient		Correlation Coefficient
TEAM_BATTING_H	0.39	**TEAM_BASERUN_SB**	0.14
TEAM_BATTING_2B	0.29	**TEAM_BASERUN_CS**	0.02
TEAM_BATTING_3B	0.14	**TEAM_FIELDING_E**	-0.18
TEAM_BATTING_HR	0.18	**TEAM_FIELDING_DP**	-0.03
TEAM_BATTING_BB	0.23	**TEAM_PITCHING_BB**	0.12
TEAM_BATTING_HBP	0.07	**TEAM_PITCHING_H**	-0.11
TEAM_BATTING_SO	-0.08	**TEAM_PITCHING_HR**	0.19
		TEAM_PITCHING_SO	-0.08

Figure 3: Correlation Table

The last step of our Exploratory Data Analysis is to check scatterplots so that we can see if linearity exists between the response variable and certain explanatory variable. This linear relationship is important to validate given that our model assumes such a relationship. Figure 4 is a scatterplot matrix that shows the relationship between TARGET_WINS and all the batting variables. As mentioned earlier, most of those variables demonstrated a positive correlation with the response variable.

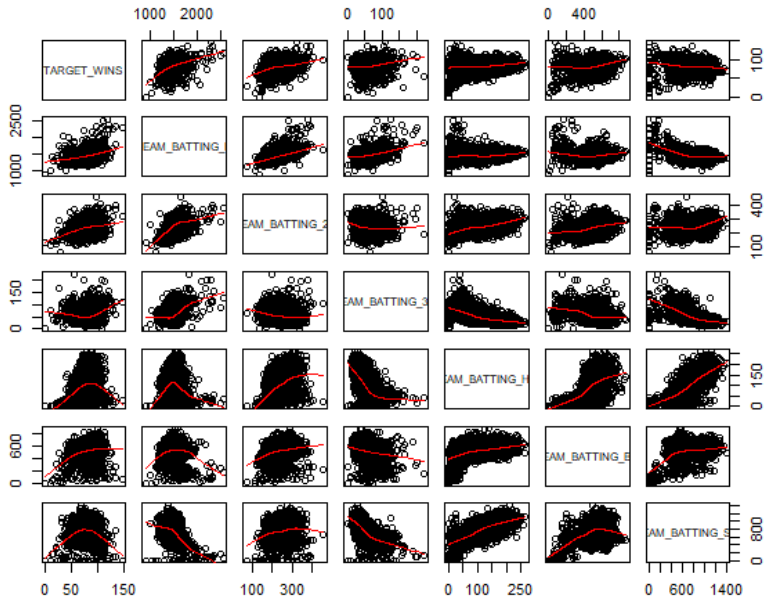

Figure 4: Batting Scatterplot Matrix

From Figure 4, we can see some signs of linearity between the response variable and the batting explanatory variables. This is something that we will keep in mind as we move into building the regression models. Also, Figure 4 shows the relationship amongst each of the batting variables. This is also important to gauge given that it could be an early sign of multicollinearity.

Model Preparation

The Model Preparation section will cover what was done on the training data set prior to fitting any models. As noted, the two main issues that we discovered during our EDA, is that the data set had missing values and outliers.

After reviewing the distribution of the response variable, it was decided that there would not be any transformation necessary. The response variable showed positive signs of being normally distributed which is important. Furthermore, the decision was made to first calculate the 5^{th} and 95^{th} percentile of each variable. Once these values were determined, any value that was less than the 5^{th} percentile and greater than the 95^{th} percentile was inputted with a NA value (e.g. missing value in R). This was done across all the variables so that each one was treated equally. Based on the EDA, it was discovered that most of the variables suffered from this problem, so the decision made to apply this imputation to all of them.

After this was complete, it left each variable with missing values. Therefore, we used a popular R package called "mice" to impute the missing values. The mice package uses an algorithm to replace the missing values with plausible data values. According to a popular mice tutorial, "these plausible values are drawn from a distribution specifically designed for each missing datapoint." The mice package is one of more popular and efficient packages that help with missing value prediction.

After using the mice package to impute missing values, the decision was made to combine variables that would demonstrate important attributes of a baseball team that one variable itself may not necessarily do well. We created 7 new variables that were derived by either adding, subtracting or multiplying original variables from the data set. For example, one of the variables that was created is called TEAM_BATTING_ONBASE which is a combination of TEAM_BATTING_H, TEAM_BATTING_BB, TEAM_BATTING_HBP. The idea is that these three variables combined illustrate how effective a team is at getting on base which in turn increases their chances of scoring more runs. The rest of the combined variables are included in the R script that is placed within the Code section of this report.

Build Model

This section of the report will cover the three regression models that were built and a discussion of some of their key aspects. Figure 5, 6 and 7 show the model summary data for the three models that were built – "Stepwise Model", "Transformed Model" and "Batting Model."

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	61.8363	4.9819	12.41	< 2e-16
TEAM_BATTING_H	0.0118	0.0048	2.49	0.01
TEAM_BATTING_BB	0.0641	0.0113	5.68	0.00
TEAM_BASERUN_SB	0.0697	0.0058	11.98	< 2e-16
TEAM_BATTING_SO	-0.0289	0.0033	-8.73	< 2e-16
TEAM_FIELDING_E	-0.0405	0.0032	-12.84	< 2e-16
TEAM_FIELDING_DP	-0.0828	0.0130	-6.37	0.00
TEAM_BATTING_3B	0.1080	0.0168	6.42	0.00
TEAM_BASERUN_CS	-0.0837	0.0193	-4.33	0.00
TEAM_BATTING_HR	0.0920	0.0084	10.96	< 2e-16
TEAM_BATTING_2B	-0.0268	0.0081	-3.31	0.00
TEAM_PITCHING_H	0.0218	0.0030	7.31	0.00
TEAM_PITCHING_H_SO_RATIO	-4.5768	0.9301	-4.92	0.00
TEAM_PITCHING_BB	-0.0475	0.0104	-4.57	0.00
Model F-statistic: 63.84 (**p-value**: < 2.2e-16) **R-squared:** 0.2684; **Adjusted R-squared:** 0.2642				

Figure 5: Stepwise Model

The Stepwise Model was calculated using the AIC statistic in Stepwise fashion. From Figure 5, we can see that the model has an adjusted R-squared value of 0.2642. There really is no ideal value to target when designing the model but it will be compared with the other three models that were developed. Furthermore, we can see that most of the variables are statistically significant as well as the entire model by reviewing the p-value for the F-statistic. Reviewing the coefficients, we can see that triples and homeruns have the largest positive impact on the response variable. This make sense as those two hits contribute to a team earning more runs. Moreover, our combined pitching variable (hits per strikeout ratio) shows that there is a large negative impact for teams that give up more hits than earning strikeouts. This makes sense given that if a team is allowing more players to get on base rather than retiring them, they are more likely to give up runs.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	27.923	5.252	5.317	0.000
TEAM_BATTING_2B	-0.067	0.017	-3.989	0.000
TEAM_BATTING_H	0.117	0.024	4.834	0.000
TEAM_BATTING_BB	0.033	0.019	1.711	0.087
TEAM_BATTING_H_SO_RATIO	2.068	1.680	1.231	0.218
TEAM_BATTING_1B	-0.057	0.016	-3.566	0.000
TEAM_PITCHING_H_SO_RATIO	-6.837	1.898	-3.602	0.000
TEAM_BATTING_ONBASE	-0.022	0.018	-1.189	0.235
TEAM_PITCHING_NON_HR	-0.011	0.003	-3.305	0.001
TEAM_PITCHING_ADVANCE_BASES	0.015	0.004	4.067	0.000
Model F-statistic: 42.28 (**p-value**: < 2.2e-16) **R-squared:** 0.1438; **Adjusted R-squared:** 0.1404				

Figure 6: Transformed Model

The Transformed Model was determined manually as opposed to using a systematic fashion such as Stepwise. The rational behind the Transformed Model is to see how well a model with mainly combined variables would perform. This model primarily includes combined variables. From Figure 6, we can see that the model has an adjusted R-squared value of 0.1404 which is lower than the Stepwise Model. Furthermore, we can see that most of the variables are statistically significant as well as the entire model by reviewing the p-value for the F-statistic. However, this model has two variables that are not statistically significant which we did not see in the Stepwise Model.
Reviewing the coefficients, we can see that the TEAM_BATTING_H_SO_RATIO appears to have the largest impact on team wins and is one of the variables that was not statistically significant. Nonetheless, this variable is a ratio of the amount of hits per strikeout by a team's batters. We would expect that the better a team is at hits than striking out, then they should be able to score more runs. This make sense as those two hits contribute to a team earning more runs. Moreover, our combined pitching variable (hits per strikeout ratio) shows that there is a large negative impact for teams that give up more hits than earning strikeouts. This makes sense given that if a team is allowing more players to get on base rather than retiring them, they are more likely to give up runs. This was also observed in the Stepwise model.

	Estimate	**Std. Error**	**t value**	**Pr(>|t|)**
(Intercept)	36.528	5.230	6.985	0.000
TEAM_BATTING_H	0.027	0.004	6.563	0.000
TEAM_BATTING_2B	-0.019	0.008	-2.219	0.027
TEAM_BATTING_3B	0.122	0.017	7.311	0.000
TEAM_BATTING_HR	0.078	0.008	9.536	< 2e-16
TEAM_BATTING_BB	0.015	0.003	5.500	0.000
TEAM_BATTING_HBP	-0.007	0.018	-0.376	0.707
TEAM_BATTING_SO	-0.012	0.004	-3.168	0.002
TEAM_BATTING_H_SO_RATIO	-1.496	1.110	-1.347	0.178
Model F-statistic: 57.82 (**p-value**: < 2.2e-16) **R-squared:** 0.1695; **Adjusted R-squared:** 0.1665				

Figure 7: Batting Model

The Batting Model was also derived using a manual process. As indicated earlier in this report, we noticed that most of the batting variables were better correlated to TARGET_WINS than the other ones. Upon developing a scatterplot matrix, it was observed that there were signs of a linear relationship between the response variable and the batting variables. Therefore, the decision was made to construct a model that only incorporated batting variables which also include combined batting variables as well. From Figure 7, we can see that the model has an adjusted R-squared value of 0.1665, which is in between the previous two models discussed.
Furthermore, we can see that most of the variables are statistically significant (except for two) as well as the entire model by reviewing the p-value for the F-statistic. Reviewing the coefficients, we can see that once again triples and homeruns have the largest positive impact on the response variable. This make sense as those two hits contribute to a team earning more runs. Moreover, we see again our combined pitching variable (hits per strikeout ratio) shows that there is a large negative impact for teams that give up more hits than earning strikeouts. This makes sense given that if a team is allowing more players to get on base rather than retiring them, they are more likely to give up runs.

Select Models

The final section of this report will review which model was determined to be our "best" model. In order to make such a validation, we calculated multiple metrics that are commonly used to compare models using training data. Figure 8 display the metrics that were calculated with respect to each model.

	Adj. R^2	Rank	AIC	Rank	BIC	Rank	MSE	Rank	MAE	Rank	Deviance	Rank
Stepwise Model	0.26	1st	17175	1st	17261	1st	109	1st	8.5	1st	249039	1st
Transformed Model	0.14	3rd	17525	3rd	17588	3rd	128	3rd	9.3	3rd	291469	3rd
Batting Model	0.16	2nd	17454	2nd	17511	2nd	124	2nd	9.1	2nd	282729	2nd

Figure 8: Model Comparison (Test Data)

With the exception of Adj. R^2, all of the metrics used are considered more accurate the lower they are when compared to each other. Figure 8 ranks where each model lies with regards to each metric. The data suggests that the Stepwise Model is the most accurate model given than it ranked 1st in every single category. However, choosing it as the "best" model is not solely based on this table. The Stepwise Model was also the only model that had all its variables be statistically significant. Lastly, the coefficients of the variables in the Stepwise Model make sense logically as well (e.g. more home runs, likely contributes to more wins). Prior to selecting it as the "best" model it's prudent to inspect its residuals against the fitted values and check of the residual normality assumption holds.

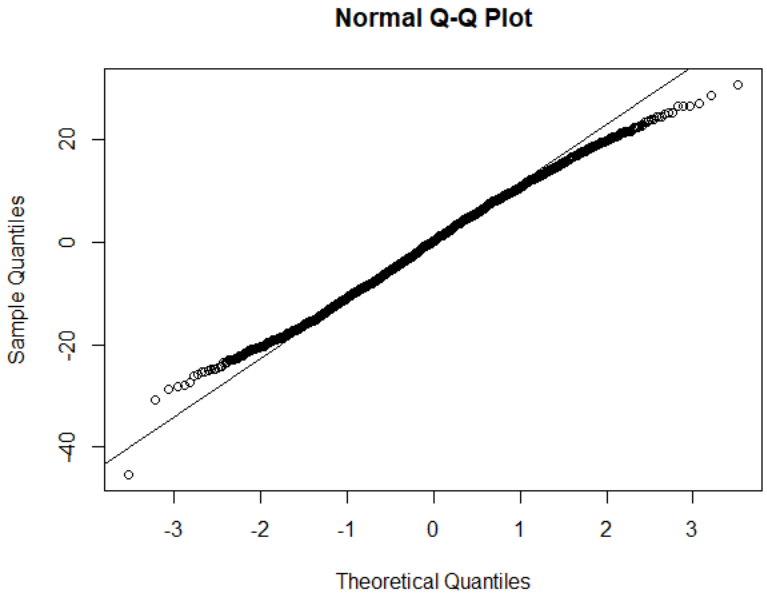

Figure 9: Stepwise Model Residuals Q-Q Plot

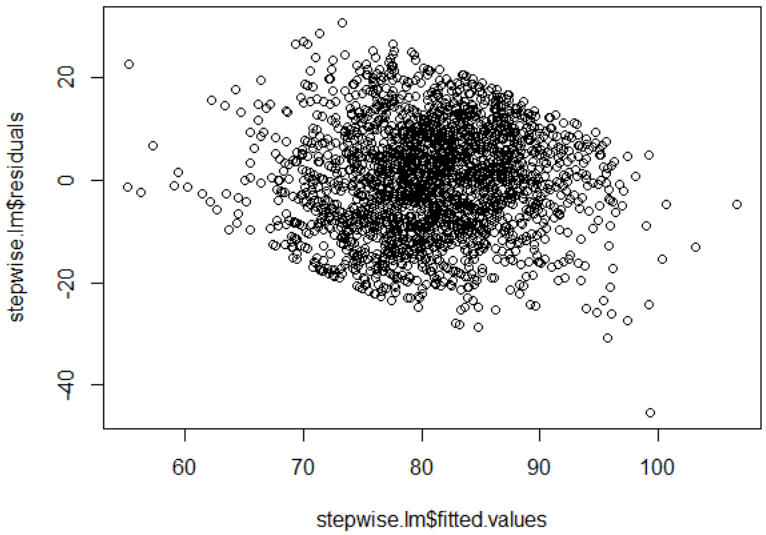

Figure 10: Stepwise Model Residuals vs. Fitted Values

Figure 9 and Figure 10 are scatterplots that allow us to visually inspect the final criteria in determining if the Stepwise Model is our "best" model. Based on the on the shapes demonstrated in both these plots (residuals fitting closely to the theoretical distribution line and the residuals showing constant variance) along with the other discussed aspects of the Stepwise Model, we believe this is our "best model."

Conclusion

The purpose of this report was to explain the analysis and logic behind developing multiple linear regression models to predict the amount of wins a baseball team should have based on a variety of baseball statistics. The Exploratory Data Analysis section was the initial groundwork that allowed us to first notice some of the subsequent work that would have to be done with regards to imputing values and getting an idea of which variables have a linear relationship with the response variable. The subsequent parts explained the logic in preparing the training data, how each model was built and ultimately what criteria was used to select the model that we believe will do the best job at predicting baseball team wins. From this exercise, it appears that it's better to include more variables, combined variables had minimal influence and that for the most part, the coefficients from the variables did not stray from preconceived notions about which is likely to impact wins. In our "best" model it was interesting to see how giving up more hits, contributes positively to earning more wins. This seems counterintuitive, but it doesn't necessarily mean that a team will lose more games by simply giving up more hits.

Code

```
############ Moneyball Unit 1 - Oscar Hernandez
################################################

##### Import necessary packages
library(mice)
library(MASS)
library(car)
#####

##### Set working directory and load test data
setwd("C:/Users/herna_000/Desktop")
moneyball <- read.csv("moneyball.csv", header=TRUE)
moneyball_original <- moneyball
#####

############ PART 1: DATA EXPLORATION
#############################################################
str(moneyball_original)
summary(moneyball_original)

#Inspect the distribution of the response variable
par(mfrow=c(1,2))
hist(moneyball_original$TARGET_WINS, col = "#A71930", xlab = "TARGET_WINS", main = "Histogram of
Wins")
boxplot(moneyball_original$TARGET_WINS, col = "#A71930", main = "Boxplot of Wins")
par(mfrow = c(1,1))

#Check the correlation between the response and each predictor variables
TEAM_BATTING_H.cor <- cor(moneyball_original$TEAM_BATTING_H,
moneyball_original$TARGET_WINS)
TEAM_BATTING_2B.cor <-cor(moneyball_original$TEAM_BATTING_2B,
moneyball_original$TARGET_WINS)
TEAM_BATTING_3B.cor <-cor(moneyball_original$TEAM_BATTING_3B,
moneyball_original$TARGET_WINS)
TEAM_BATTING_HR.cor<-cor(moneyball_original$TEAM_BATTING_HR,
moneyball_original$TARGET_WINS)
TEAM_BATTING_BB.cor <-cor(moneyball_original$TEAM_BATTING_BB,
moneyball_original$TARGET_WINS)
TEAM_BATTING_HBP.cor <-cor(moneyball_original$TEAM_BATTING_HBP,
moneyball_original$TARGET_WINS, use="na.or.complete")
TEAM_BATTING_SO.cor <-cor(moneyball_original$TEAM_BATTING_SO,
moneyball_original$TARGET_WINS, use="na.or.complete")
TEAM_BASERUN_SB.cor <-cor(moneyball_original$TEAM_BASERUN_SB,
moneyball_original$TARGET_WINS, use="na.or.complete")
TEAM_BASERUN_CS.cor <-cor(moneyball_original$TEAM_BASERUN_CS,
moneyball_original$TARGET_WINS, use="na.or.complete")
```

```
TEAM_FIELDING_E.cor <-cor(moneyball_original$TEAM_FIELDING_E,
moneyball_original$TARGET_WINS)
TEAM_FIELDING_DP.cor <-cor(moneyball_original$TEAM_FIELDING_DP,
moneyball_original$TARGET_WINS, use="na.or.complete")
TEAM_PITCHING_BB.cor <-cor(moneyball_original$TEAM_PITCHING_BB,
moneyball_original$TARGET_WINS)
TEAM_PITCHING_H.cor <-cor(moneyball_original$TEAM_PITCHING_H,
moneyball_original$TARGET_WINS)
TEAM_PITCHING_HR.cor <-cor(moneyball_original$TEAM_PITCHING_HR,
moneyball_original$TARGET_WINS)
TEAM_PITCHING_SO.cor <-
cor(moneyball_original$TEAM_PITCHING_SO,moneyball_original$TARGET_WINS, use =
"na.or.complete")

#Check scatterplot of the response and each predictor variable
plot(moneyball_original$TEAM_BATTING_H, moneyball_original$TARGET_WINS, main="Target Wins vs.
Batting Hits") #somewhat linear
plot(moneyball_original$TEAM_BATTING_2B, moneyball_original$TARGET_WINS, main="Target Wins
vs. Batting 2B") #more linear
plot(moneyball_original$TEAM_BATTING_3B, moneyball_original$TARGET_WINS, main="Target Wins
vs. Batting 3B") #somewhat linear
plot(moneyball_original$TEAM_BATTING_HR,  moneyball_original$TARGET_WINS, main="Target Wins
vs. Batting HR") #more linear
plot(moneyball_original$TEAM_BATTING_BB, moneyball_original$TARGET_WINS, main="Target Wins
vs. Batting BB") #kind of linear
plot(moneyball_original$TEAM_BATTING_HBP, moneyball_original$TARGET_WINS, main="Target Wins
vs. Batting HBP") #somewhat linear
plot(moneyball_original$TEAM_BATTING_SO, moneyball_original$TARGET_WINS, main="Target Wins
vs. Batting SO") #somewhat linear
plot(moneyball_original$TEAM_BASERUN_SB, moneyball_original$TARGET_WINS, main="Target Wins
vs. Baserun SB") #not really linear
plot(moneyball_original$TEAM_BASERUN_CS, moneyball_original$TARGET_WINS, main="Target Wins
vs. Baserun CS") #not really linear
plot(moneyball_original$TEAM_FIELDING_E, moneyball_original$TARGET_WINS, main="Target Wins vs.
Fielding E") #not linear
plot(moneyball_original$TEAM_FIELDING_DP, moneyball_original$TARGET_WINS, main="Target Wins
vs. Fielding DP") #somwhat linear
plot(moneyball_original$TEAM_PITCHING_BB, moneyball_original$TARGET_WINS, main="Target Wins
vs. Pitching BB") #not linear
plot(moneyball_original$TEAM_PITCHING_H, moneyball_original$TARGET_WINS, main="Target Wins
vs. Pitching H") #not linear
plot(moneyball_original$TEAM_PITCHING_HR, moneyball_original$TARGET_WINS, main="Target Wins
vs. Pitching HR") #somewhat linear
plot(moneyball_original$TEAM_PITCHING_SO,moneyball_original$TARGET_WINS, main="Target Wins
vs. Pitching SO") #not linear

#Correlation Matrix
panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)
```

```
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

#Batting correlation matrix
pairs(moneyball_original[2:8], lower.panel=panel.smooth, upper.panel = panel.smooth)

#Pitching correlation
pairs(~ moneyball_original$TARGET_WINS + moneyball_original$TEAM_PITCHING_BB +
moneyball_original$TEAM_PITCHING_H +
      moneyball_original$TEAM_PITCHING_HR + moneyball_original$TEAM_PITCHING_SO, lower.panel =
panel.smooth, upper.panel = panel.smooth)

#Calculates the percentage of observations within each variable that's missing
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(moneyball_original,2,pMiss)

############ PART 2: DATA PREPARATION
#############################################################

#Determine the observations that are <= the 5th percentile and >= the 95th percentile for each variable
#These will be considered outliers and will be imputed

#Wins
quantile(moneyball_original$TARGET_WINS, 0.95) #104
quantile(moneyball_original$TARGET_WINS, 0.05) #54

#Batting
quantile(moneyball_original$TEAM_BATTING_H, 0.95) #1695.25
quantile(moneyball_original$TEAM_BATTING_H, 0.05) #1281.5
quantile(moneyball_original$TEAM_BATTING_2B, 0.95)#320
quantile(moneyball_original$TEAM_BATTING_2B, 0.05)#167
quantile(moneyball_original$TEAM_BATTING_3B , 0.95)#108
quantile(moneyball_original$TEAM_BATTING_3B , 0.05)#23
quantile(moneyball_original$TEAM_BATTING_HR  , 0.95)#199
quantile(moneyball_original$TEAM_BATTING_HR  , 0.05)#14
quantile(moneyball_original$TEAM_BATTING_BB , 0.95) #670.25
quantile(moneyball_original$TEAM_BATTING_BB , 0.05) #248.25
quantile(moneyball_original$TEAM_BATTING_HBP, 0.95, na.rm = TRUE) #82.5
quantile(moneyball_original$TEAM_BATTING_HBP , 0.05, na.rm = TRUE) #40
quantile(moneyball_original$TEAM_BATTING_SO   , 0.95, na.rm=TRUE) #1103.35
quantile(moneyball_original$TEAM_BATTING_SO   , 0.05, na.rm=TRUE) #359
```

```
#Baserun
quantile(moneyball_original$TEAM_BASERUN_SB, 0.95, na.rm=TRUE) #301.8
quantile(moneyball_original$TEAM_BASERUN_SB , 0.05, na.rm=TRUE) #35
quantile(moneyball_original$TEAM_BASERUN_CS , 0.95, na.rm=TRUE) #91
quantile(moneyball_original$TEAM_BASERUN_CS  , 0.05, na.rm=TRUE) #24

#Fielding
quantile(moneyball_original$TEAM_FIELDING_E , 0.95) #716
quantile(moneyball_original$TEAM_FIELDING_E , 0.05) #100
quantile(moneyball_original$TEAM_FIELDING_DP , 0.95, na.rm=TRUE) #186
quantile(moneyball_original$TEAM_FIELDING_DP , 0.05, na.rm=TRUE) #98

#Pitching
quantile(moneyball_original$TEAM_PITCHING_BB , 0.95) #757
quantile(moneyball_original$TEAM_PITCHING_BB , 0.05) #377
quantile(moneyball_original$TEAM_PITCHING_H , 0.95) #2563
quantile(moneyball_original$TEAM_PITCHING_H , 0.05) #1316
quantile(moneyball_original$TEAM_PITCHING_HR, 0.95) #209.25
quantile(moneyball_original$TEAM_PITCHING_HR , 0.05) #18
quantile(moneyball_original$TEAM_PITCHING_SO , 0.95, na.rm=TRUE) #1173
quantile(moneyball_original$TEAM_PITCHING_SO , 0.05, na.rm=TRUE) #421.3

#Impute values that are less than 5th percentile or greater than 95th percentile
#Target Wins
moneyball$TARGET_WINS[moneyball$TARGET_WINS>104] <- NA
moneyball$TARGET_WINS[moneyball$TARGET_WINS<54] <- NA
#Batting
moneyball$TEAM_BATTING_H[moneyball$TEAM_BATTING_H>1695] <- NA
moneyball$TEAM_BATTING_H[moneyball$TEAM_BATTING_H<1281]  <-NA
moneyball$TEAM_BATTING_2B[moneyball$TEAM_BATTING_2B>320] <- NA
moneyball$TEAM_BATTING_2B[moneyball$TEAM_BATTING_2B<167] <- NA
moneyball$TEAM_BATTING_3B[moneyball$TEAM_BATTING_3B>108] <- NA
moneyball$TEAM_BATTING_3B[moneyball$TEAM_BATTING_3B<23] <- NA
moneyball$TEAM_BATTING_HR[moneyball$TEAM_BATTING_HR>199] <- NA
moneyball$TEAM_BATTING_HR[moneyball$TEAM_BATTING_HR<14] <- NA
moneyball$TEAM_BATTING_BB[moneyball$TEAM_BATTING_BB>670] <- NA
moneyball$TEAM_BATTING_BB[moneyball$TEAM_BATTING_BB<248] <- NA
moneyball$TEAM_BATTING_HBP[moneyball$TEAM_BATTING_HBP>82] <- NA
moneyball$TEAM_BATTING_HBP[moneyball$TEAM_BATTING_HBP<40] <- NA
moneyball$TEAM_BATTING_SO[moneyball$TEAM_BATTING_SO>1103] <- NA
moneyball$TEAM_BATTING_SO[moneyball$TEAM_BATTING_SO<359] <- NA

#Baserun
moneyball$TEAM_BASERUN_SB[moneyball$TEAM_BASERUN_SB>301] = NA
moneyball$TEAM_BASERUN_SB[moneyball$TEAM_BASERUN_SB<35] = NA
moneyball$TEAM_BASERUN_CS[moneyball$TEAM_BASERUN_CS>91] = NA
moneyball$TEAM_BASERUN_CS[moneyball$TEAM_BASERUN_CS<24] = NA
```

```
#Fielding
moneyball$TEAM_FIELDING_E[moneyball$TEAM_FIELDING_E>716] <- NA
moneyball$TEAM_FIELDING_E[moneyball$TEAM_FIELDING_E<100] <- NA
moneyball$TEAM_FIELDING_DP[moneyball$TEAM_FIELDING_DP>186] <- NA
moneyball$TEAM_FIELDING_DP[moneyball$TEAM_FIELDING_DP<98] <- NA

#Pitching
moneyball$TEAM_PITCHING_BB[moneyball$TEAM_PITCHING_BB>757] <- NA
moneyball$TEAM_PITCHING_BB[moneyball$TEAM_PITCHING_BB<377] <- NA
moneyball$TEAM_PITCHING_H[moneyball$TEAM_PITCHING_H>2563] <- NA
moneyball$TEAM_PITCHING_H[moneyball$TEAM_PITCHING_H<1316] <- NA
moneyball$TEAM_PITCHING_HR[moneyball$TEAM_PITCHING_HR>209] <- NA
moneyball$TEAM_PITCHING_HR[moneyball$TEAM_PITCHING_HR<18] <- NA
moneyball$TEAM_PITCHING_SO[moneyball$TEAM_PITCHING_SO>1173] <- NA
moneyball$TEAM_PITCHING_SO[moneyball$TEAM_PITCHING_SO<421] <- NA

#Impute missing values using MICE package
index_column <- data.frame(moneyball$INDEX)

#using PMM
#don't include Index column
moneyball_temp1<-mice(moneyball[, !names(moneyball) %in% "INDEX"], m=5, maxit=50, meth="pmm",
seed = 500)
summary(moneyball_temp1)

moneyball_imp1 <- complete(moneyball_temp1)
moneyball_imp1 <-cbind(index_column, moneyball_imp1)
colnames(moneyball_imp1)[colnames(moneyball_imp1)=="moneyball.INDEX"] <- "INDEX"

#using RF
#don't include Index column
moneyball_temp2<-mice(moneyball[, !names(moneyball) %in% "INDEX"], m=5, maxit=50, meth="rf",
seed = 750)
summary(moneyball_temp2)

moneyball_imp2 <- complete(moneyball_temp2)
moneyball_imp2 <-cbind(index_column, moneyball_imp2)

#Check for correlation using the first imputed data frame
cor(moneyball_imp1$TEAM_BATTING_H, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_BATTING_2B, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_BATTING_3B, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_BATTING_HR,  moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_BATTING_BB, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_BATTING_HBP, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_BATTING_SO, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_BASERUN_SB, moneyball_imp1$TARGET_WINS)
```

```
cor(moneyball_imp1$TEAM_BASERUN_CS, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_FIELDING_E, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_FIELDING_DP, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_PITCHING_BB, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_PITCHING_H, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_PITCHING_HR, moneyball_imp1$TARGET_WINS)
cor(moneyball_imp1$TEAM_PITCHING_SO,moneyball_imp1$TARGET_WINS)

#Create new variables
moneyball_imp1$TEAM_BATTING_H_SO_RATIO <-
moneyball_imp1$TEAM_BATTING_H/moneyball_imp1$TEAM_BATTING_SO
moneyball_imp1$TEAM_BATTING_1B <- moneyball_imp1$TEAM_BATTING_H-
moneyball_imp1$TEAM_BATTING_2B - moneyball_imp1$TEAM_BATTING_3B
-moneyball_imp1$TEAM_BATTING_HR
moneyball_imp1$TEAM_PITCHING_H_SO_RATIO <-
moneyball_imp1$TEAM_PITCHING_H/moneyball_imp1$TEAM_PITCHING_SO
moneyball_imp1$TEAM_BATTING_ONBASE <-
moneyball_imp1$TEAM_BATTING_H+moneyball_imp1$TEAM_BATTING_BB+moneyball_imp1$TEAM_B
ATTING_HBP
moneyball_imp1$TEAM_BATTING_ADVANCE_BASES <-
moneyball_imp1$TEAM_BATTING_H+moneyball_imp1$TEAM_BATTING_BB+moneyball_imp1$TEAM_B
ATTING_HBP
+moneyball_imp1$TEAM_BASERUN_SB-moneyball_imp1$TEAM_BASERUN_CS
moneyball_imp1$TEAM_PITCHING_NON_HR <- moneyball_imp1$TEAM_PITCHING_H-
moneyball_imp1$TEAM_PITCHING_HR
moneyball_imp1$TEAM_PITCHING_ADVANCE_BASES <- moneyball_imp1$TEAM_PITCHING_H
+moneyball_imp1$TEAM_PITCHING_BB-moneyball_imp1$TEAM_PITCHING_SO
-moneyball_imp1$TEAM_FIELDING_DP-moneyball_imp1$TEAM_FIELDING_E

############ PART 3: BUILD MODELS
#####################################################################
#Build model using automated variable selection - Stepwise Approach
#Need to specifiy the upper model and lower models
# Define the upper model as the FULL model
upper.lm <- lm(TARGET_WINS ~ .,data=moneyball_imp1[, !names(moneyball) %in% "INDEX"])

# Define the lower model as the Intercept model
lower.lm <- lm(TARGET_WINS ~ 1,data=moneyball_imp1[, !names(moneyball) %in% "INDEX"])

# Need a simple linear regression model to initialize stepwise selection
initialize_stepwise.lm <- lm(TARGET_WINS ~ TEAM_BATTING_H,data=moneyball_imp1[,
!names(moneyball) %in% "INDEX"])

#Create stepwise.lm
stepwise.lm <- stepAIC(object=initialize_stepwise.lm,scope=list(upper=formula(upper.lm),lower=~1),
direction=c('both'))
summary(stepwise.lm)
```

```
#Check for normality with residuals
qqnorm(stepwise.lm$res)
qqline(stepwise.lm$res)

#Check for constant variance
plot(stepwise.lm$fitted.values, stepwise.lm$residuals)

#Build model using  transformed variables along with three highest correlated variables
transformed_lm <- lm(TARGET_WINS ~ TEAM_BATTING_2B+ +TEAM_BATTING_H +TEAM_BATTING_BB
+ TEAM_BATTING_H_SO_RATIO + TEAM_BATTING_1B + TEAM_PITCHING_H_SO_RATIO +
TEAM_BATTING_ONBASE +
                TEAM_PITCHING_NON_HR + TEAM_PITCHING_ADVANCE_BASES, data = moneyball_imp1)
summary(transformed_lm)

#Check for normality with residuals
qqnorm(transformed_lm$res)
qqline(transformed_lm$res)

#Check for constant variance
plot(transformed_lm$fitted.values, transformed_lm$residuals)

#Build model using only batting variables
batting_lm <- lm(TARGET_WINS ~ TEAM_BATTING_H  +TEAM_BATTING_2B
        +TEAM_BATTING_3B  +TEAM_BATTING_HR  +TEAM_BATTING_BB  +TEAM_BATTING_HBP
+TEAM_BATTING_SO
        + TEAM_BATTING_H_SO_RATIO, data = moneyball_imp1)

summary(batting_lm)

#Check for normality with residuals
qqnorm(batting_lm$res)
qqline(batting_lm$res)

#Check for constant variance
plot(batting_lm$fitted.values, batting_lm$residuals)


#Create forward.lm
forward.lm <- stepAIC(object=lower.lm,scope=list(upper=formula(upper.lm),lower=~1),
direction=c('forward'))
summary(forward.lm)

#Create backward.lm
backward.lm <- stepAIC(object=upper.lm,direction=c('backward'))
summary(backward.lm)
```

```
############ PART 4: SELECT MODELS
#####################################################################

#Look at VIFs
sort(vif(stepwise.lm),decreasing=TRUE)
sort(vif(transformed_lm),decreasing=TRUE)
sort(vif(batting_lm),decreasing=TRUE)

#Deviance
deviance(stepwise.lm)
deviance(transformed_lm)
deviance(batting_lm)

#AIC and BIC
AIC(stepwise.lm)
AIC(transformed_lm)
AIC(batting_lm)

BIC(stepwise.lm)
BIC(transformed_lm)
BIC(batting_lm)

#MSE AND MAE
mse.stepwise <- mean(stepwise.lm$residuals^2)
mse.stepwise
mae.stepwise <- mean(abs(stepwise.lm$residuals))
mae.stepwise

mse.transformed <- mean(transformed_lm$residuals^2)
mse.transformed
mae.transformed <- mean(abs(transformed_lm$residuals))
mae.transformed

mse.batting <- mean(batting_lm$residuals^2)
mse.batting
mae.batting <- mean(abs(batting_lm$residuals))
mae.batting
```