CUNY DATA 621

Homework 1 (Moneyball)

Ahsanul Choudhury

1. DATA EXPLORATION

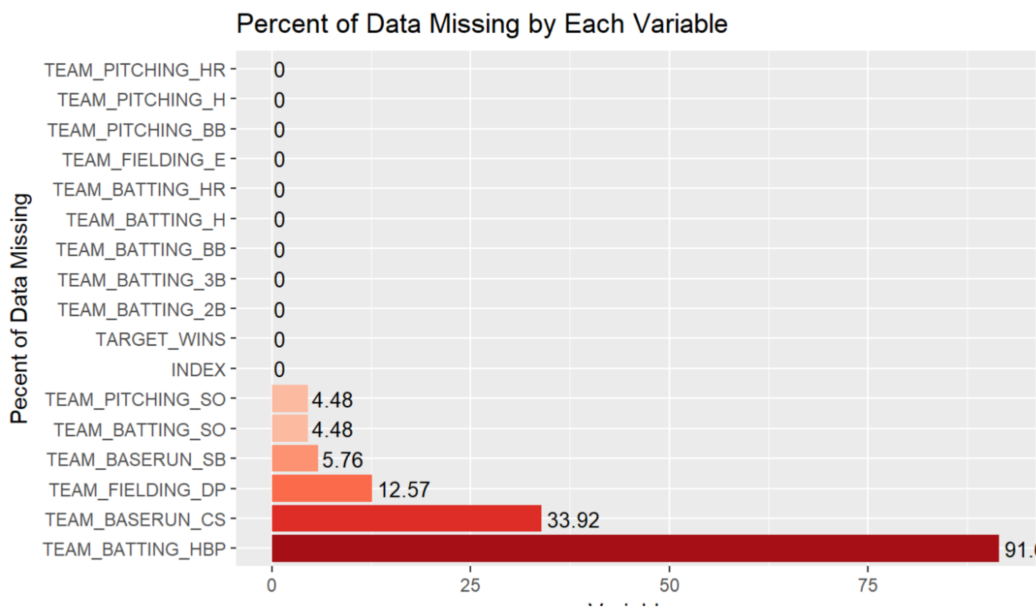
The Moneyball data set contains 2276 observations and 17 variables. Each observation represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season. Out of the 17 variables the INDEX variable is the index value for each observation. Below is a description of each variable:

Variable	**Description**
INDEX	Index
TARGET_WINS	Number of 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	Homeruns by batters (4B)
TEAM_BATTING_BB	Walks by batters
TEAM_BATTING_HBP	Batters hit by pitch
TEAM_BATTING_SO	Strikeouts by batters
TEAM_BASERUN_SB	Stolen bases
TEAM_BASERUN_CS	Caught stealing
TEAM_FIELDING_E	Errors
TEAM_FIELDING_DP	Double Plays
TEAM_PITCHING_BB	Walks allowed
TEAM_PITCHING_H	Hits Allowed
TEAM_PITCHING_HR	Homeruns Allowed
TEAM_PITCHING_SO	Strikeouts by pitcher

TARGET_WINS is our response variable and the remaining 15 variable are potential predictor variables. There are 6 variables with a total of 3478 missing values in the dataset. The following table and graph shows the missing values:

Percentage of Missing Values in Each Variable

Varibles	Percent of Data Missing
INDEX	0.00
TARGET_WINS	0.00
TEAM_BATTING_H	0.00
TEAM_BATTING_2B	0.00
TEAM_BATTING_3B	0.00
TEAM_BATTING_HR	0.00
TEAM_BATTING_BB	0.00
TEAM_BATTING_SO	4.48
TEAM_BASERUN_SB	5.76
TEAM_BASERUN_CS	33.92
TEAM_BATTING_HBP	91.61
TEAM_PITCHING_H	0.00
TEAM_PITCHING_HR	0.00
TEAM_PITCHING_BB	0.00
TEAM_PITCHING_SO	4.48
TEAM_FIELDING_E	0.00
TEAM_FIELDING_DP	12.57



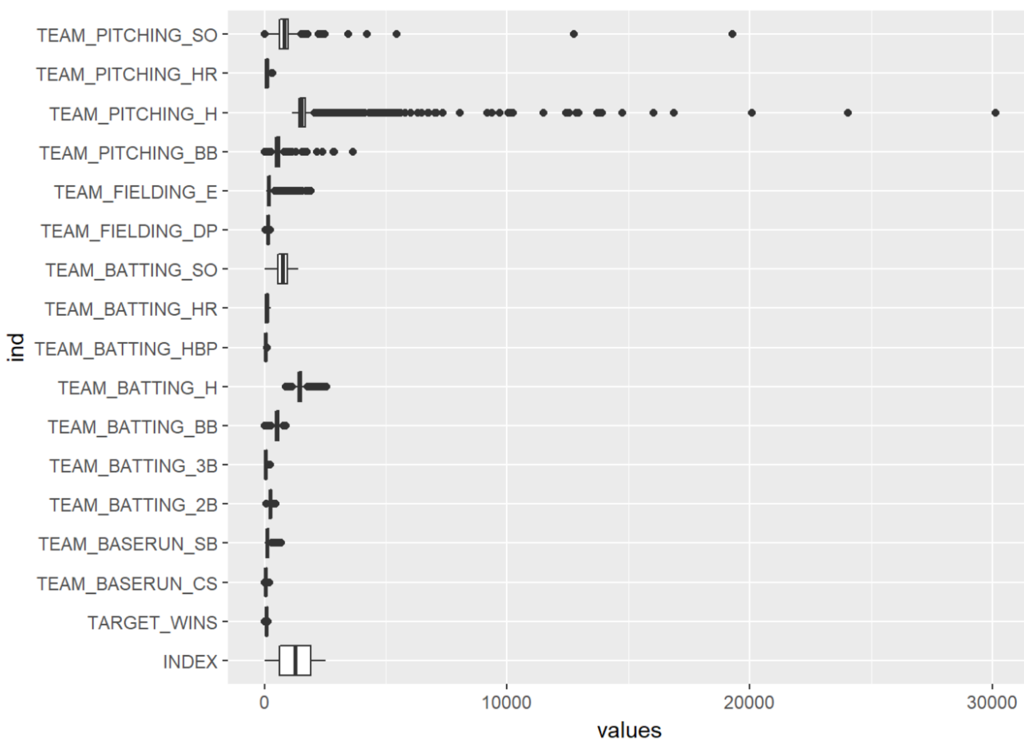
The table below shows the summary statistics of the variables in the dataset:

Summary Statistics

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
INDEX	1	2276	1268.46353	736.34904	1270.5	1268.56970	952.5705	1	2535	2534	0.0042149	-1.2167564	15.4346788
TARGET_WINS	2	2276	80.79086	15.75215	82.0	81.31229	14.8260	0	146	146	-0.3987232	1.0274757	0.3301823
TEAM_BATTING_H	3	2276	1469.26977	144.59120	1454.0	1459.04116	114.1602	891	2554	1663	1.5713335	7.2785261	3.0307891
TEAM_BATTING_2B	4	2276	241.24692	46.80141	238.0	240.39627	47.4432	69	458	389	0.2151018	0.0061609	0.9810087
TEAM_BATTING_3B	5	2276	55.25000	27.93856	47.0	52.17563	23.7216	0	223	223	1.1094652	1.5032418	0.5856226
TEAM_BATTING_HR	6	2276	99.61204	60.54687	102.0	97.38529	78.5778	0	264	264	0.1860421	-0.9631189	1.2691285
TEAM_BATTING_BB	7	2276	501.55888	122.67086	512.0	512.18331	94.8864	0	878	878	-1.0257599	2.1828544	2.5713150
TEAM_BATTING_SO	8	2174	735.60534	248.52642	750.0	742.31322	284.6592	0	1399	1399	-0.2978001	-0.3207992	5.3301912
TEAM_BASERUN_SB	9	2145	124.76177	87.79117	101.0	110.81188	60.7866	0	697	697	1.9724140	5.4896754	1.8955584
TEAM_BASERUN_CS	10	1504	52.80386	22.95634	49.0	50.35963	17.7912	0	201	201	1.9762180	7.6203818	0.5919414
TEAM_BATTING_HBP	11	191	59.35602	12.96712	58.0	58.86275	11.8608	29	95	66	0.3185754	-0.1119828	0.9382681
TEAM_PITCHING_H	12	2276	1779.21046	1406.84293	1518.0	1555.89517	174.9468	1137	30132	28995	10.3295111	141.8396985	29.4889618
TEAM_PITCHING_HR	13	2276	105.69859	61.29875	107.0	103.15697	74.1300	0	343	343	0.2877877	-0.6046311	1.2848886
TEAM_PITCHING_BB	14	2276	553.00791	166.35736	536.5	542.62459	98.5929	0	3645	3645	6.7438995	96.9676398	3.4870317
TEAM_PITCHING_SO	15	2174	817.73045	553.08503	813.5	796.93391	257.2311	0	19278	19278	22.1745535	671.1891292	11.8621151
TEAM_FIELDING_E	16	2276	246.48067	227.77097	159.0	193.43798	62.2692	65	1898	1833	2.9904656	10.9702717	4.7743279
TEAM_FIELDING_DP	17	1990	146.38794	26.22639	149.0	147.57789	23.7216	52	228	176	-0.3889390	0.1817397	0.5879114

From the summary statistics of the dataset we can see there are few variable with high degree of skewness and Kurtosis. This indicates presence of outliers in those variable, a look at the boxplot of the variables will give us a clearer picture.

Boxplot of Variables



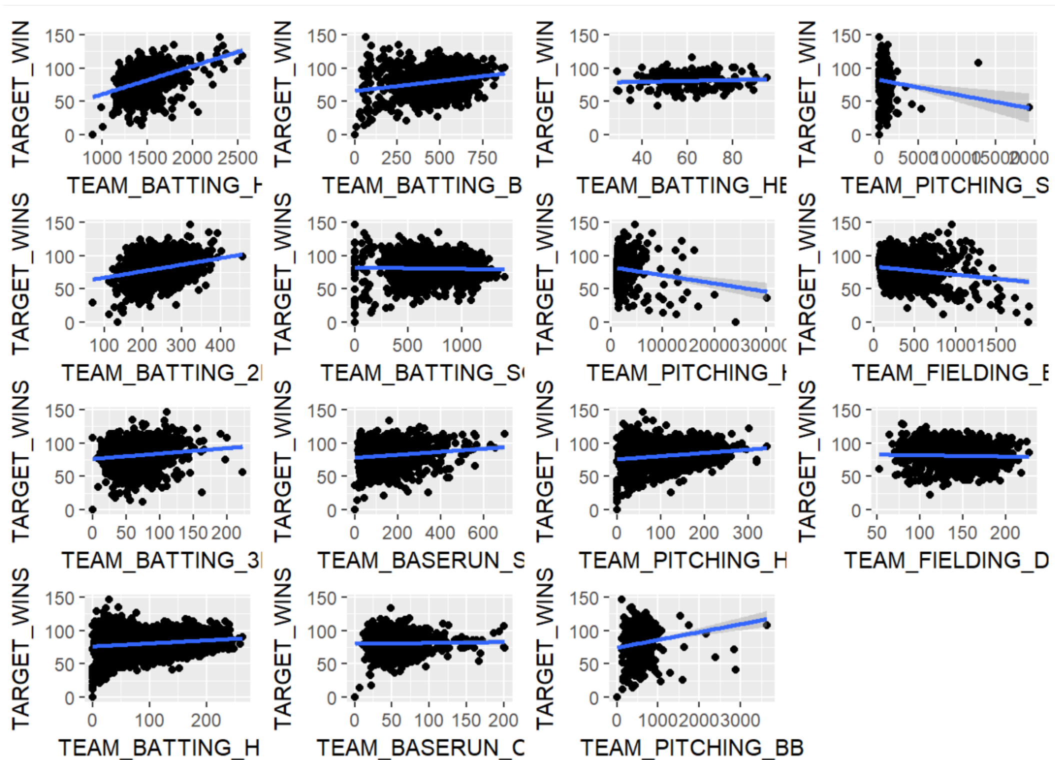
The boxplot above confirms what we have seen in the summary statistics, TEAM_PITICHING_SO and TEAM_PITICHING_HR has some significant outliers.

The following table shows correlation between TARGET_WINS and the remaining variables:

INDEX	-0.02
TARGET_WINS	1.00
TEAM_BATTING_H	0.39
TEAM_BATTING_2B	0.29
TEAM_BATTING_3B	0.14
TEAM_BATTING_HR	0.18
TEAM_BATTING_BB	0.23
TEAM_BATTING_SO	NA
TEAM_BASERUN_SB	NA
TEAM_BASERUN_CS	NA
TEAM_BATTING_HBP	NA
TEAM_PITCHING_H	-0.11
TEAM_PITCHING_HR	0.19
TEAM_PITCHING_BB	0.12
TEAM_PITCHING_SO	NA
TEAM_FIELDING_E	-0.18
TEAM_FIELDING_DP	NA

We can see from the table TEAM_BATTING_H (0.39), TEAM_BATTING_2B (0.29) and TEAM_BATTING_BB(0.23) has the highest correlation. The plots below show the correlation between TARGET_WINS and the other variables:

Relationship Between TARGET_WIN and Other Variables



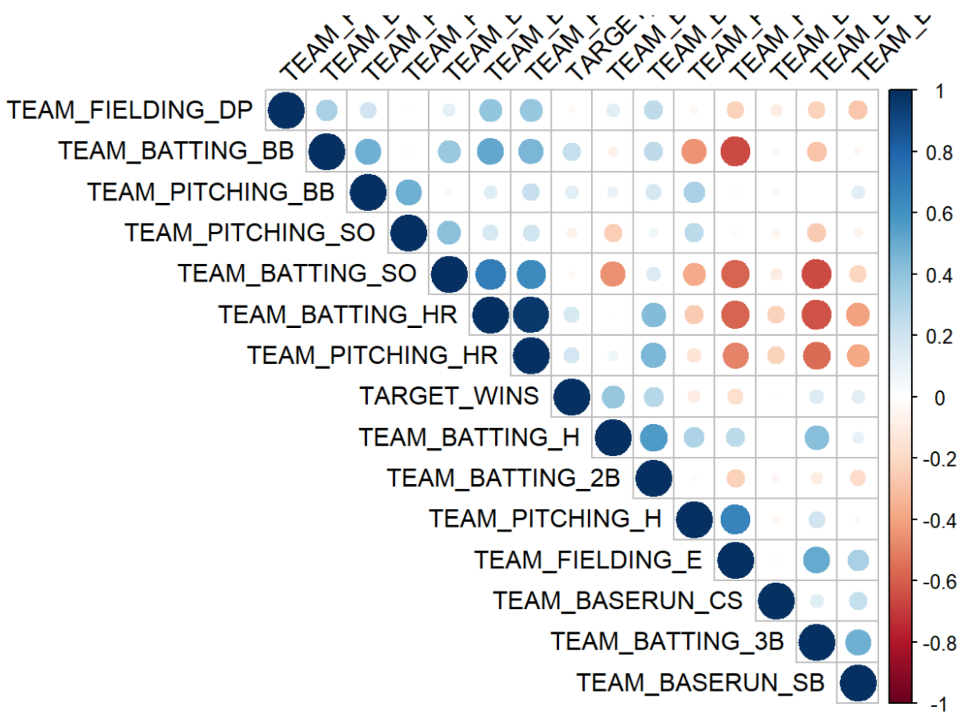
2. DATA PREPARATION

As we have seen in the Data Exploration part TEAM_BATTING_HBP has about 92% of the data missing we will take out the variable entirely. For the remaining 5 variables with missing values we will use Median Imputation. After imputation and removal of TEAM_BATTING_HBP we find the following correlation between TARGER_WINS and the rest of the variables:

Correlation After Median Imputaion

variables	correlation1
TARGET_WINS	1.00
TEAM_BATTING_H	0.39
TEAM_BATTING_2B	0.29
TEAM_BATTING_3B	0.14
TEAM_BATTING_HR	0.18
TEAM_BATTING_BB	0.23
TEAM_BATTING_SO	-0.03
TEAM_BASERUN_SB	0.12
TEAM_BASERUN_CS	0.02
TEAM_PITCHING_H	-0.11
TEAM_PITCHING_HR	0.19
TEAM_PITCHING_BB	0.12
TEAM_PITCHING_SO	-0.08
TEAM_FIELDING_E	-0.18
TEAM_FIELDING_DP	-0.03

Correlation Matrix



3. Build Model

Model 1: Backward Selection

For our first model I chose backward selection model, we will start with fitting a model with all the variable of interest then we will start dropping the least significant variables. We will continue doing so until only the significant variable remains.

Using all the remaining 15 variables we get following results:

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.752  -8.626   0.120   8.395  58.561
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      23.6414208  5.3902300   4.386 1.21e-05 ***
## TEAM_BATTING_H    0.0489152  0.0036949  13.239  < 2e-16 ***
## TEAM_BATTING_2B  -0.0209571  0.0091783  -2.283 0.022503 *
## TEAM_BATTING_3B   0.0644777  0.0168040   3.837 0.000128 ***
## TEAM_BATTING_HR   0.0527287  0.0274912   1.918 0.055234 .
## TEAM_BATTING_BB   0.0104509  0.0058376   1.790 0.073547 .
## TEAM_BATTING_SO  -0.0084337  0.0025461  -3.312 0.000940 ***
## TEAM_BASERUN_SB   0.0254237  0.0043565   5.836 6.12e-09 ***
## TEAM_BASERUN_CS  -0.0110004  0.0157842  -0.697 0.485920
## TEAM_PITCHING_H  -0.0008456  0.0003674  -2.302 0.021440 *
## TEAM_PITCHING_HR  0.0129688  0.0243894   0.532 0.594958
## TEAM_PITCHING_BB  0.0007775  0.0041571   0.187 0.851654
## TEAM_PITCHING_SO  0.0028164  0.0009219   3.055 0.002278 **
## TEAM_FIELDING_E  -0.0195320  0.0024609  -7.937 3.23e-15 ***
## TEAM_FIELDING_DP -0.1217768  0.0129420  -9.409  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.07 on 2261 degrees of freedom
## Multiple R-squared:  0.3154, Adjusted R-squared:  0.3111
## F-statistic:  74.4 on 14 and 2261 DF,  p-value: < 2.2e-16
```

We get a we get adjusted R-squared value of 0.3111 and a F-statistic is 74.4. For our next model we have dropped the variable with the highest p-value, TEAM_PITCHING_BB and got the following results:

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
##     TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_PITCHING_H + TEAM_PITCHING_HR +
##     TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.699  -8.633   0.129   8.398  58.543
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      23.5994132  5.3843996   4.383 1.22e-05 ***
## TEAM_BATTING_H    0.0488784  0.0036888  13.250  < 2e-16 ***
## TEAM_BATTING_2B  -0.0209330  0.0091754  -2.281 0.022616 *
## TEAM_BATTING_3B   0.0644693  0.0168004   3.837 0.000128 ***
## TEAM_BATTING_HR   0.0502599  0.0241097   2.085 0.037215 *
## TEAM_BATTING_BB   0.0113395  0.0033909   3.344 0.000839 ***
## TEAM_BATTING_SO  -0.0085602  0.0024542  -3.488 0.000496 ***
## TEAM_BASERUN_SB   0.0255474  0.0043050   5.934 3.40e-09 ***
## TEAM_BASERUN_CS  -0.0111088  0.0157702  -0.704 0.481243
## TEAM_PITCHING_H  -0.0008148  0.0003283  -2.482 0.013140 *
## TEAM_PITCHING_HR  0.0152725  0.0210461   0.726 0.468118
## TEAM_PITCHING_SO  0.0029342  0.0006730   4.360 1.36e-05 ***
## TEAM_FIELDING_E  -0.0195164  0.0024589  -7.937 3.23e-15 ***
## TEAM_FIELDING_DP -0.1217364  0.0129375  -9.410  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.07 on 2262 degrees of freedom
## Multiple R-squared:  0.3154, Adjusted R-squared:  0.3114
## F-statistic: 80.15 on 13 and 2262 DF,  p-value: < 2.2e-16
```

After removing TEAM_PITCHING_BB our adjusted R-squared value improves to 0.3114 and F-statistic to 80.15. In our next model we have dropped TEAM_BASERUN_CS AND TEAM_PITCHING_HR which have the new highest p-values values and got the following results:

```
##
## Call:
## lm(formula = 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_SO + TEAM_FIELDING_E +
##     TEAM_FIELDING_DP, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.598  -8.593   0.085   8.445  58.581
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      22.3435812  5.2338329   4.269 2.04e-05 ***
## TEAM_BATTING_H    0.0490923  0.0036699  13.377  < 2e-16 ***
## TEAM_BATTING_2B  -0.0213740  0.0091625  -2.333 0.019747 *
## TEAM_BATTING_3B   0.0665751  0.0166230   4.005 6.40e-05 ***
## TEAM_BATTING_HR   0.0674074  0.0096316   6.999 3.39e-12 ***
## TEAM_BATTING_BB   0.0115464  0.0033748   3.421 0.000634 ***
## TEAM_BATTING_SO  -0.0085222  0.0024530  -3.474 0.000522 ***
## TEAM_BASERUN_SB   0.0249206  0.0042092   5.920 3.70e-09 ***
## TEAM_PITCHING_H  -0.0007772  0.0003209  -2.421 0.015538 *
## TEAM_PITCHING_SO  0.0029667  0.0006719   4.415 1.06e-05 ***
## TEAM_FIELDING_E  -0.0190097  0.0023919  -7.947 2.98e-15 ***
## TEAM_FIELDING_DP -0.1217860  0.0129295  -9.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.07 on 2264 degrees of freedom
## Multiple R-squared:  0.3151, Adjusted R-squared:  0.3117
## F-statistic: 94.68 on 11 and 2264 DF,  p-value: < 2.2e-16
```

Our R-squared value improves to 0.3117 and F-statistic to 94.68, at this point all our remaining variables look statistically significant, p-value is lower than .05 so we will conclude the model here.

Model 2: Simple Model

For our next model we will use the simple model. The simple model takes only the most important variables and ignores the rest. For this model we used four significant predictors (TEAM_BATTING_H, TEAM_BASERUN_SB, TEAM_FIELDING_E and TEAM_FIELDING_DP) and got the following results:

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BASERUN_SB +
##     TEAM_FIELDING_DP + TEAM_FIELDING_E, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.940  -8.906   0.032   8.540  55.823
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      17.927279   3.198192   5.605 2.33e-08 ***
## TEAM_BATTING_H    0.053588   0.002051  26.131  < 2e-16 ***
## TEAM_BASERUN_SB   0.029683   0.003554   8.353  < 2e-16 ***
## TEAM_FIELDING_DP -0.087792   0.012244  -7.170 1.01e-12 ***
## TEAM_FIELDING_E  -0.026998   0.001367 -19.744  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.34 on 2271 degrees of freedom
## Multiple R-squared:  0.2841, Adjusted R-squared:  0.2829
## F-statistic: 225.3 on 4 and 2271 DF,  p-value: < 2.2e-16
```

We get a R-squared value of 0.2841 and F-statistics of 225.3.

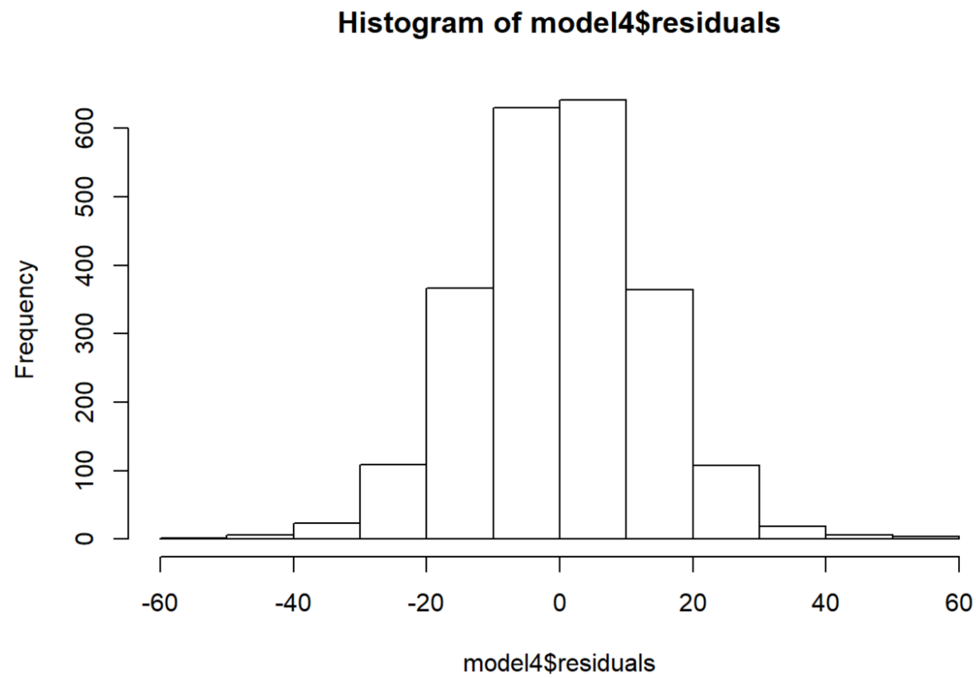
4. SELECT MODEL

The chart below summarizes our two models:

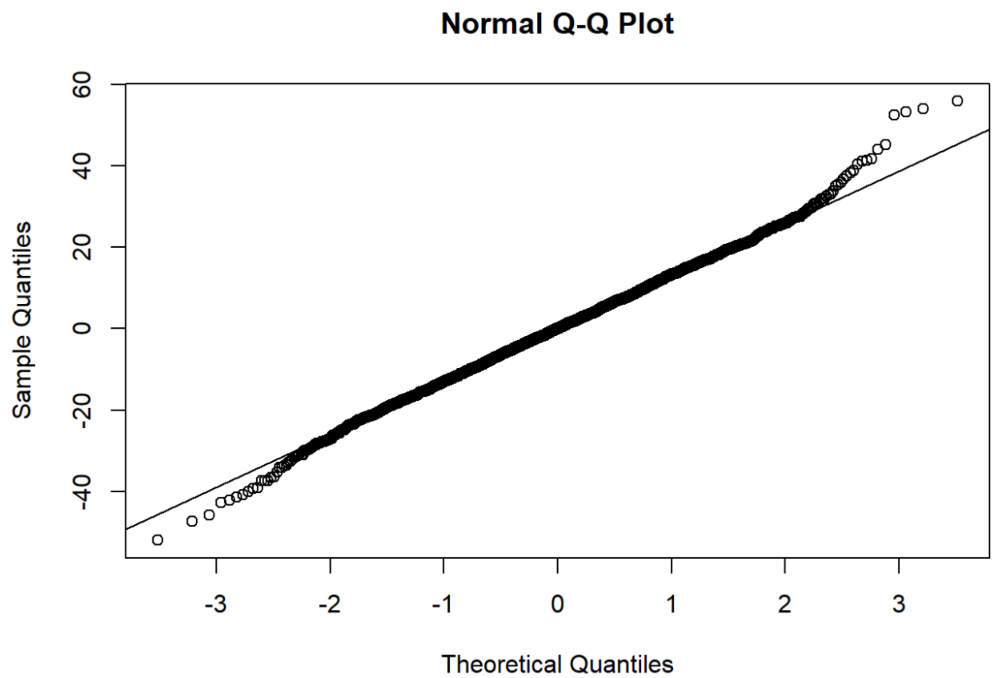
Metric	Backward Selection	Simple Model
R-squared	0.3117	0.2841
F-statistic	94.68	225.3

The backward selection model yields a larger R-squared value than the simple model, but the simple model yields a much larger F-statistics. The Larger F-statistics indicates the simple model explains more of the variability in the training data. So, for our project we will select the **Simple Model**.

Let’s look at some plots for our selected model:


Histogram of model4$residuals
Frequency
0
100
200
300
400
500
600
-60
-40
-20
0
20
40
60
model4$residuals


model4$residuals
-40
-20
0
20
40
60
0
50
100
150
training$TARGET_WINS



From diagnostic plots of our Simple Model we can see the residuals mimics a normal distribution and are random, there may be some issues with outliers which we need to be careful with.

Predictions

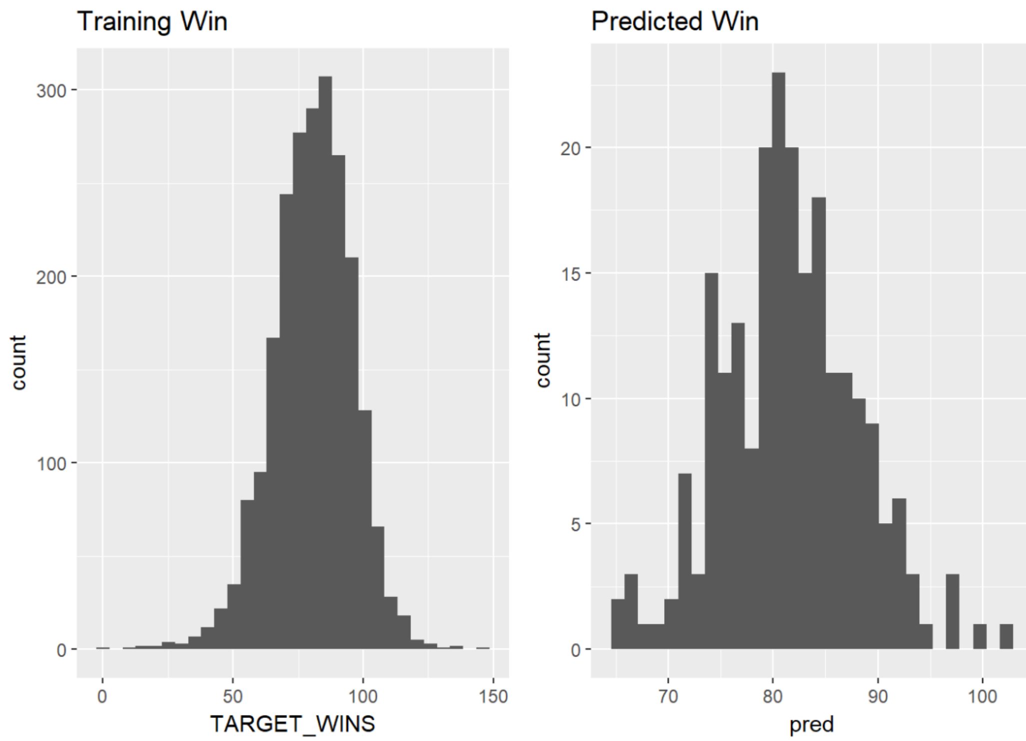
Below is a summary of our win prediction using the simple model:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   65.12   77.11   81.23   81.55   85.64  102.20      36
```

The full prediction can be found in the following link:

https://github.com/choudhury1023/DATA-621/blob/master/HW%201/mb_predictions.csv

Histogram Comparing Training Win and Predicted Win Distribution



Appendix: R code

```
if (!require('psych')) install.packages('psych')
if (!require('knitr')) install.packages('knitr')
if (!require('ggplot2')) install.packages('ggplot2')
if (!require('RColorBrewer')) install.packages('RColorBrewer')
if (!require('gridExtra')) install.packages('gridExtra')
if (!require('corrplot')) install.packages('corrplot')
# Load data
training_data <- read.csv("https://raw.githubusercontent.com/choudhury1023/DATA-621/master/HW%201/moneyball-training-data.csv")
eval_data <- read.csv("https://raw.githubusercontent.com/choudhury1023/DATA-621/master/HW%201/moneyball-evaluation-data.csv")

# Data exploration

dim(training_data)
```

```
missing_train <- colSums(is.na(training_data))
total_missing <- sum(missing_train)
total_missing

missing_train <- round(missing_train/dim(training_data)[1]*100,2)
df_missing <- data.frame(missing_train)
df_missing <- cbind(variables = rownames(df_missing), df_missing)
df_missing_table <- df_missing
colnames(df_missing_table) <- c("Varibles", "Percent of Data Missing")
rownames(df_missing_table) <- NULL

kable(df_missing_table)


# Missing data plot

ggplot(df_missing, aes(x = reorder(variables, -missing_train), y = missing_train,
fill=factor(missing_train))) +
  scale_fill_brewer(palette="Reds") +
  geom_bar(stat = "identity") + coord_flip() +
  geom_text(aes(label = missing_train), hjust= -0.1, size=3.5) +
  xlab("Pecent of Data Missing") + ylab("Variables") +
  ggtitle("Percent of Data Missing by Each Variable") +
  theme(legend.position="bottom")



# Data Summary

des_train <- describe(training_data)
knitr::kable(des_train)


# Boxplot

ggplot(stack(training_data), aes(x = ind, y = values)) + geom_boxplot() + coord_flip()


# Correlation plot

correlation <- round(apply(training_data,2, function(col)cor(col, training_data$TARGET_WINS)),2)
df_correlation <- data.frame(correlation)
df_correlation <- cbind(variables = rownames(df_correlation), df_correlation)
rownames(df_correlation) <- NULL
kable(df_correlation)
```

```
plots <- list() # empty list for plots

for(i in 3:17){
  plots[[i-2]] <-
    ggplot(training_data,
           aes_string(colnames(training_data)[i],colnames(training_data)[2])) +
    geom_point() + stat_smooth(method="lm")
}

source("http://peterhaschke.com/Code/multiplot.R")


multiplot(plotlist = plots, cols = 4)

######

## Data prep
# Remove INDEX, TEAM_BATTING_HBP

training <- subset(training_data, select = -c(INDEX, TEAM_BATTING_HBP) )

# Median imputation

cs <- round(median(training$TEAM_BASERUN_CS, na.rm=T))
dp <- round(median(training$TEAM_FIELDING_DP, na.rm=T))
sb <- round(median(training$TEAM_BASERUN_SB, na.rm=T))
bso <- round(median(training$TEAM_BATTING_SO, na.rm=T))
pso <- round(median(training$TEAM_PITCHING_SO, na.rm=T))


training[['TEAM_BASERUN_CS']][is.na(training[['TEAM_BASERUN_CS']])] <- cs
training[['TEAM_FIELDING_DP']][is.na(training[['TEAM_FIELDING_DP']])] <- dp
training[['TEAM_BASERUN_SB']][is.na(training[['TEAM_BASERUN_SB']])] <- sb
training[['TEAM_BATTING_SO']][is.na(training[['TEAM_BATTING_SO']])] <- bso
training[['TEAM_PITCHING_SO']][is.na(training[['TEAM_PITCHING_SO']])] <- pso


# Correlation after imputation

correlation1 <- round(apply(training,2, function(col)cor(col, training$TARGET_WINS)),2)
df_correlation1 <- data.frame(correlation1)
df_correlation1 <- cbind(variables = rownames(df_correlation1), df_correlation1)
rownames(df_correlation1) <- NULL
kable(df_correlation1)


# correlation matrix plot
```

```
cm <- cor(training)
corrplot(cm, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)


#######


## Build Model
# All variables
model1 <- lm(data = training, TARGET_WINS ~ .)
summary(model1)


# Drop TEAM_PITCHING_BB
model2 <- 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_BASERUN_CS + TEAM_PITCHING_H +
TEAM_PITCHING_HR + TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP,
data = training)
summary(model2)

# Drop TEAM_BASERUN_CS and TEAM_PITCHING_HR

model3 <- 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_SO + TEAM_FIELDING_E
+ TEAM_FIELDING_DP, data = training)
summary(model3)


# Simple Model
model4 <- lm(TARGET_WINS ~  TEAM_BATTING_H + TEAM_BASERUN_SB +
TEAM_FIELDING_DP + TEAM_FIELDING_E, training)
summary(model4)


#######

## Select Model

# Selected model plots

hist(model4$residuals)

plot(model4$residuals~training$TARGET_WINS)
```

```
abline(h=0,lty=3)

qqnorm(model4$residuals)
qqline(model4$residuals)

pred <- predict(model4,eval_data)
summary(pred)

pred <- data.frame(pred)
write.csv(pred, "mb_predictions.csv")

p_train <- ggplot(training, aes(TARGET_WINS)) + geom_histogram() + ggtitle("Training Win")
p_pred <- ggplot(pred, aes(pred)) + geom_histogram() + ggtitle("Predicted Win")

grid.arrange(p_train, p_pred, ncol=2)
```

URL Link: https://github.com/choudhury1023/DATA-621/blob/master/HW%201/Ahsanul_Choudhury_HW1.R