DATA 621 Homework 1

Bin Lin

2017-3-1

Overview:

The data set contains approximately 2200 records. Each record 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.

DATA EXPLORATION:

This dataset contains total 17 variables, each of which is numerical data. TARGET_WINS is our response variable. Its median is 82, mean is 80.79, and standard deviation is 15.752. According to the histogram of TARGET_WINS, we can tell it is bell shaped, symmetric and unimodal. The reasonable assumption is that it is normally distributed. The qq-plot has further prove that since most of the data points form a straight line along qqline. The number of explanatory variables are as many as 15. The scatterplots between TARGET_WINS and all other explanatory variables did not display obvious relationship. From the summary statistics, we have noticed that many variables have missing data. This will be taken care of in the data properation section.

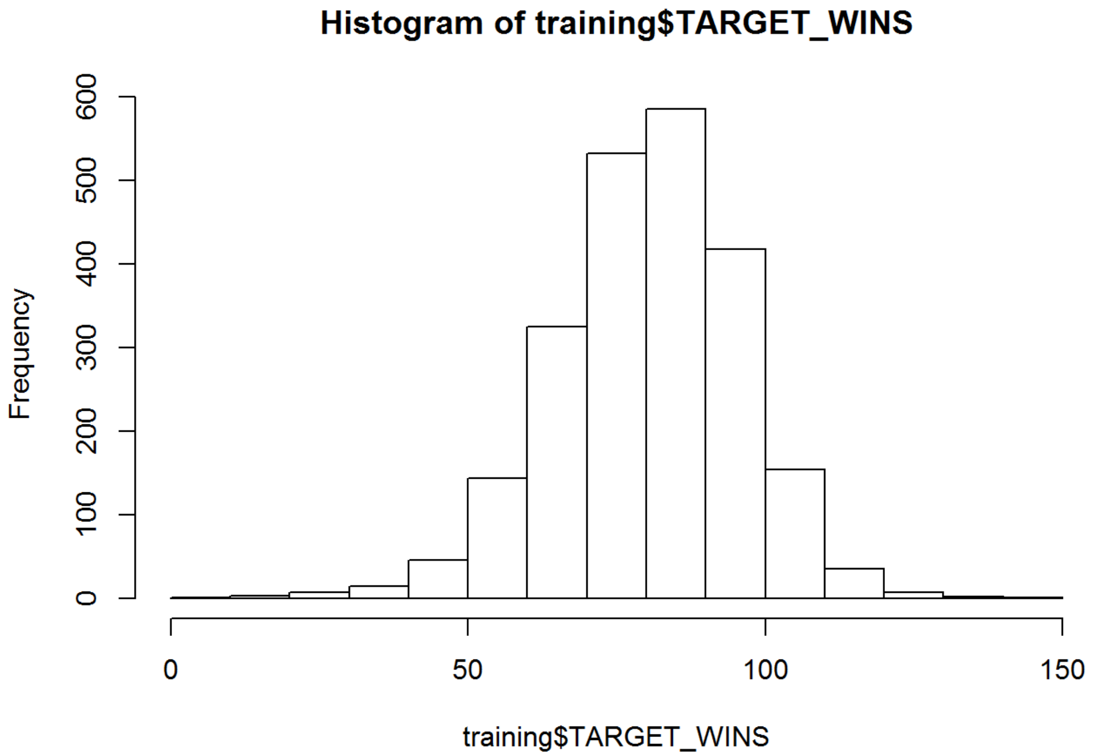
```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

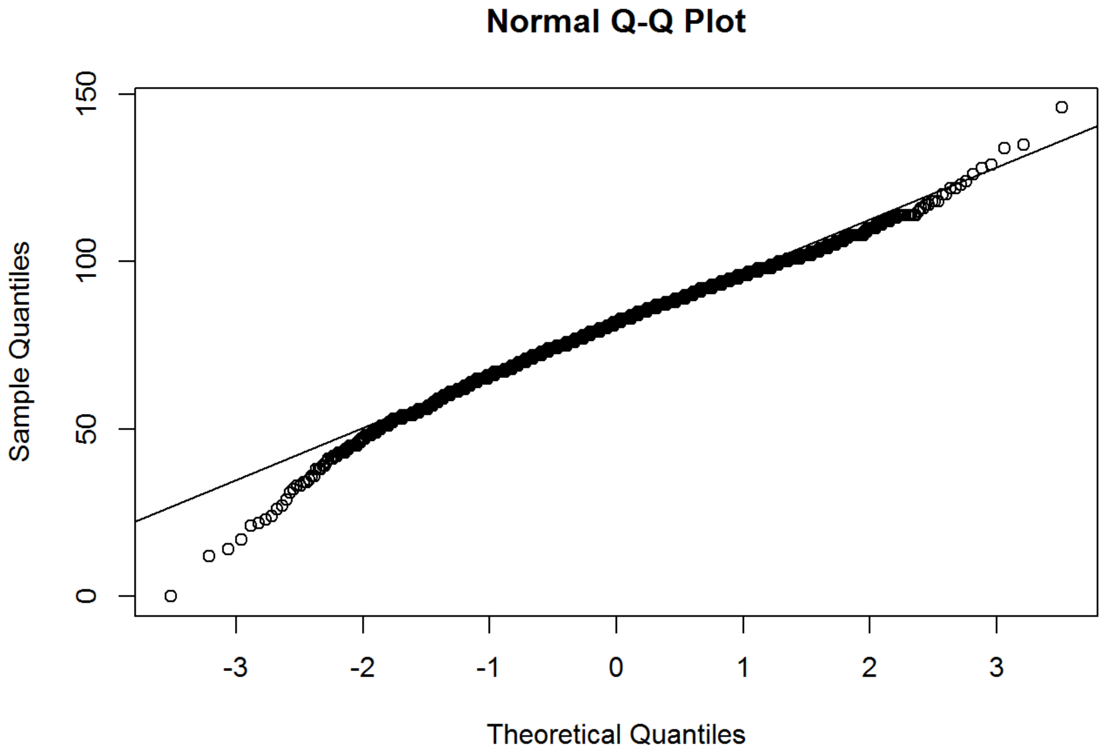
```
## 
## Attaching package: 'psych'
```

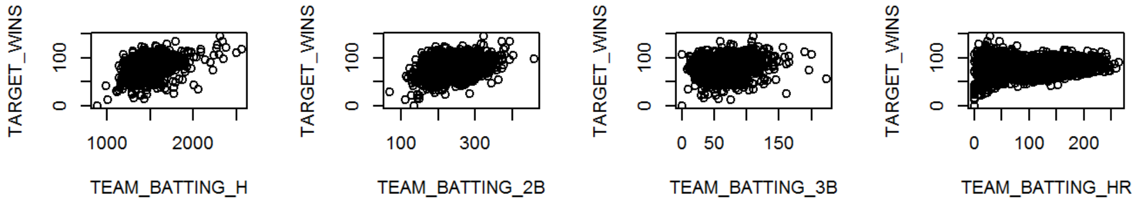
```
## The following objects are masked from 'package:ggplot2':
## 
##     %+%, alpha
```

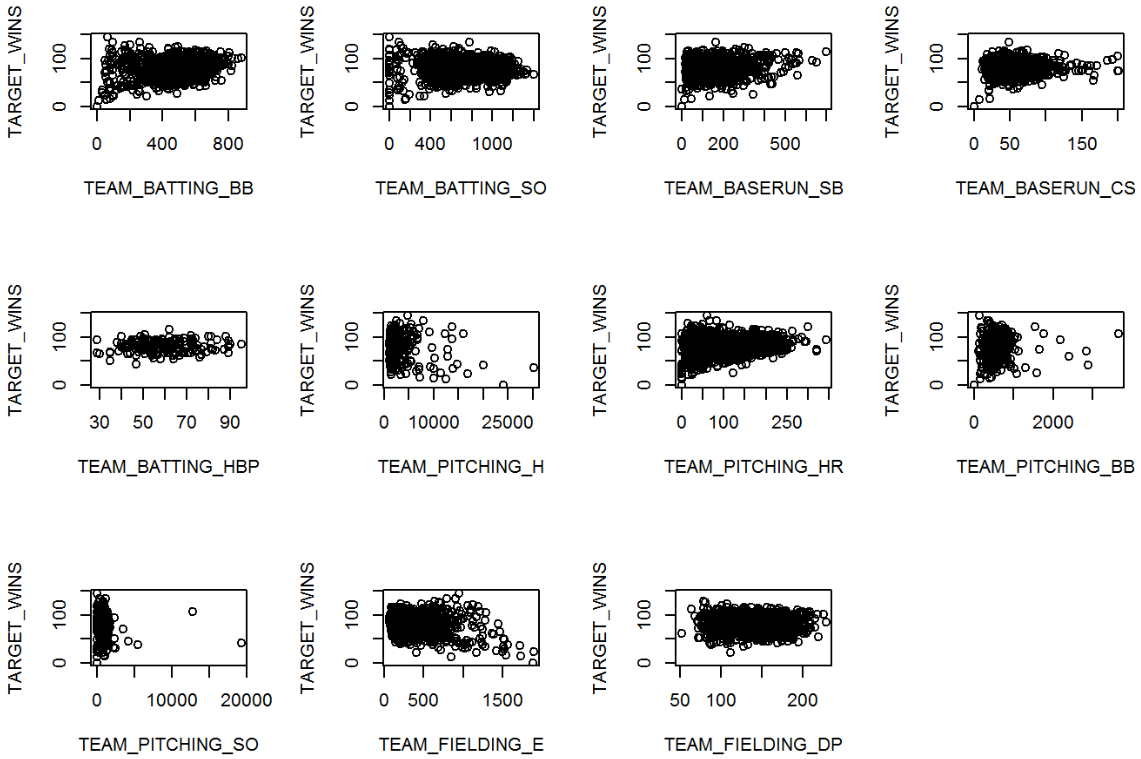
```
## 'data.frame':    2276 obs. of  17 variables:
##  $ INDEX           : int  1 2 3 4 5 6 7 8 11 12 ...
##  $ TARGET_WINS     : int  39 70 86 70 82 75 80 85 86 76 ...
##  $ TEAM_BATTING_H  : int  1445 1339 1377 1387 1297 1279 1244 1273 1391 1271 ...
##  $ TEAM_BATTING_2B : int  194 219 232 209 186 200 179 171 197 213 ...
##  $ TEAM_BATTING_3B : int  39 22 35 38 27 36 54 37 40 18 ...
##  $ TEAM_BATTING_HR : int  13 190 137 96 102 92 122 115 114 96 ...
##  $ TEAM_BATTING_BB : int  143 685 602 451 472 443 525 456 447 441 ...
##  $ TEAM_BATTING_SO : int  842 1075 917 922 920 973 1062 1027 922 827 ...
##  $ TEAM_BASERUN_SB : int  NA 37 46 43 49 107 80 40 69 72 ...
##  $ TEAM_BASERUN_CS : int  NA 28 27 30 39 59 54 36 27 34 ...
##  $ TEAM_BATTING_HBP: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TEAM_PITCHING_H : int  9364 1347 1377 1396 1297 1279 1244 1281 1391 1271 ...
##  $ TEAM_PITCHING_HR: int  84 191 137 97 102 92 122 116 114 96 ...
##  $ TEAM_PITCHING_BB: int  927 689 602 454 472 443 525 459 447 441 ...
##  $ TEAM_PITCHING_SO: int  5456 1082 917 928 920 973 1062 1033 922 827 ...
##  $ TEAM_FIELDING_E : int  1011 193 175 164 138 123 136 112 127 131 ...
##  $ TEAM_FIELDING_DP: int  NA 155 153 156 168 149 186 136 169 159 ...
##   TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
##  Min.   :  0.00   Min.   : 891   Min.   : 69.0   Min.   :  0.00
##  1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0   1st Qu.: 34.00
##  Median : 82.00   Median :1454   Median :238.0   Median : 47.00
##  Mean   : 80.79   Mean   :1469   Mean   :241.2   Mean   : 55.25
##  3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0   3rd Qu.: 72.00
##  Max.   :146.00   Max.   :2554   Max.   :458.0   Max.   :223.00
##                                                                
##  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB
##  Min.   :  0.00   Min.   :  0.0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0   1st Qu.: 66.0
##  Median :102.00   Median :512.0   Median : 750.0   Median :101.0
##  Mean   : 99.61   Mean   :501.6   Mean   : 735.6   Mean   :124.8
##  3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0   3rd Qu.:156.0
##  Max.   :264.00   Max.   :878.0   Max.   :1399.0   Max.   :697.0
##                                   NA's   :102      NA's   :131
##  TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
##  Min.   :  0.0   Min.   :29.00    Min.   : 1137   Min.   :  0.0
##  1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419   1st Qu.: 50.0
##  Median : 49.0   Median :58.00    Median : 1518   Median :107.0
##  Mean   : 52.8   Mean   :59.36    Mean   : 1779   Mean   :105.7
##  3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682   3rd Qu.:150.0
##  Max.   :201.0   Max.   :95.00    Max.   :30132   Max.   :343.0
##  NA's   :772     NA's   :2085
##  TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E  TEAM_FIELDING_DP
##  Min.   :   0.0   Min.   :    0.0   Min.   :  65.0   Min.   : 52.0
##  1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0   1st Qu.:131.0
##  Median : 536.5   Median :  813.5   Median : 159.0   Median :149.0
##  Mean   : 553.0   Mean   :  817.7   Mean   : 246.5   Mean   :146.4
##  3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2   3rd Qu.:164.0
##  Max.   :3645.0   Max.   :19278.0   Max.   :1898.0   Max.   :228.0
##                   NA's   :102                        NA's   :286
```

```
## [1] 15.75215
```


Histogram of training$TARGET_WINS
Frequency
0
100
200
300
400
500
600
0
50
100
150
training$TARGET_WINS


Normal Q-Q Plot
Sample Quantiles
0
50
100
150
-3
-2
-1
0
1
2
3
Theoretical Quantiles

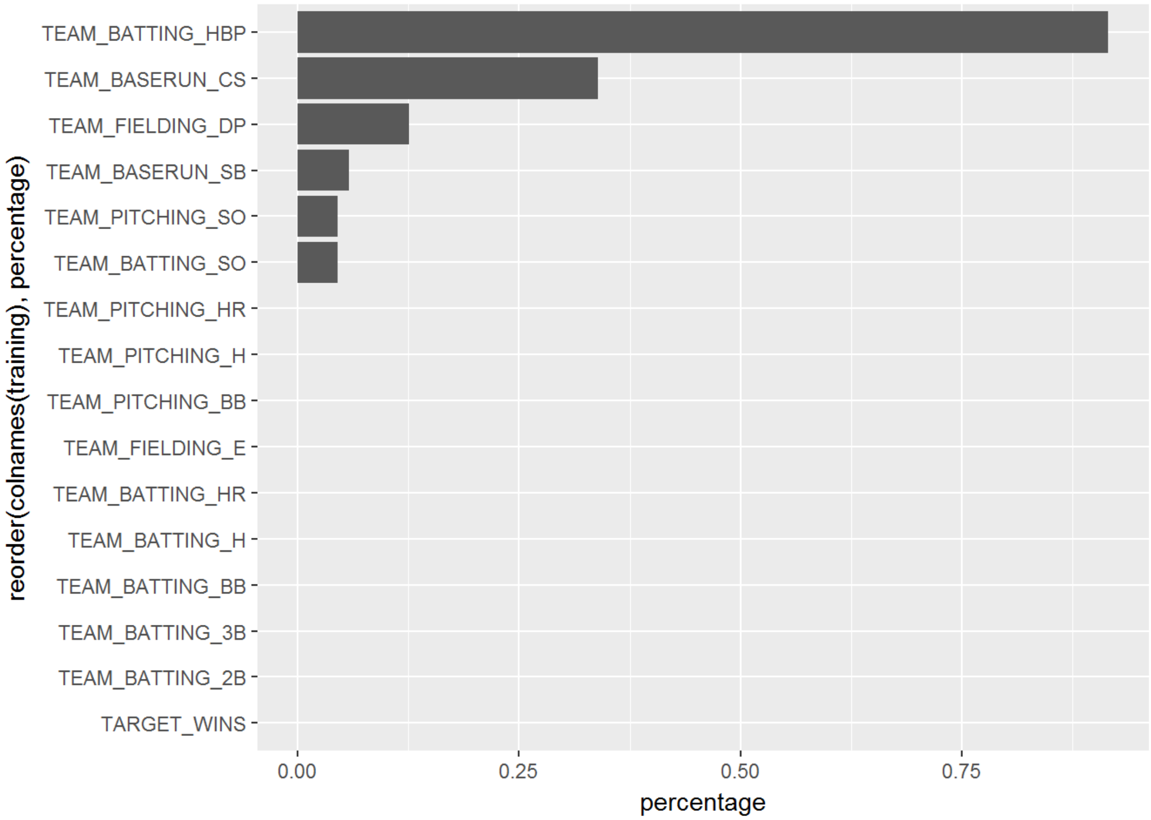

TARGET_WINS
0
100
1000
2000
TEAM_BATTING_H
TARGET_WINS
0
100
100
300
TEAM_BATTING_2B
TARGET_WINS
0
100
0
50
150
TEAM_BATTING_3B
TARGET_WINS
0
100
0
100
200
TEAM_BATTING_HR

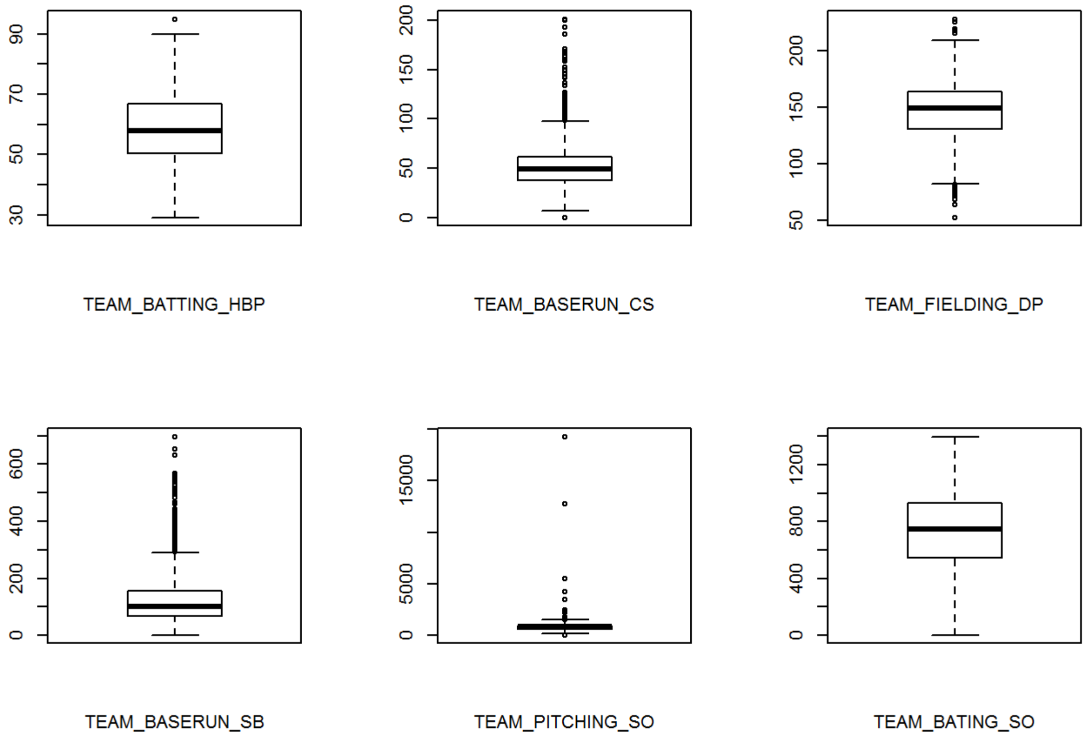


#DATA

PREPARATION:

After summing up all the missing values for each explanatory variables, I created a data frame which is going to show the frequency and percentage for occured missing data. The barplot clearly shows there are 6 variables which have large amount of data that is missing. By creating the histogram for each of these 6 variables, we are able to find the pattern for them, so that we will be able to replace the null values with some values that are more meaningful in this case. Apparently, TEAM_BATTING_HBP, TEAM_BASERUN_CS and TEAM_FIELDING_DP have missing more than 10% of the data, they have been excluded from the analysis. TEAM_BASERUN_SB and TEAM_PITCHING_SO are both is skewed to the left with many outliers, so median can better represent these two variables compare to mean. TEAM_BATING_SO is highly symmetric, therefore I use mean to resplace its missing values.





BUILD MODELS:

Model 1:

The first model I created include all the variables that were left in the training dataset. The coefficient for each variable is shown in the following.

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.474  -8.937   0.118   8.640  56.858
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.6020478  5.1785786   1.661  0.09684 .
## TEAM_BATTING_H   0.0481025  0.0037642  12.779  < 2e-16 ***
## TEAM_BATTING_2B -0.0230727  0.0093442  -2.469  0.01361 *
## TEAM_BATTING_3B  0.0724491  0.0170933   4.238 2.34e-05 ***
## TEAM_BATTING_HR  0.0378938  0.0278312   1.362  0.17347
## TEAM_BATTING_BB  0.0045910  0.0059125   0.777  0.43753
## TEAM_BATTING_SO -0.0050877  0.0025696  -1.980  0.04783 *
## TEAM_BASERUN_SB  0.0304943  0.0042949   7.100 1.66e-12 ***
## TEAM_PITCHING_H -0.0008241  0.0003738  -2.205  0.02758 *
## TEAM_PITCHING_HR  0.0108311  0.0248386   0.436  0.66284
## TEAM_PITCHING_BB  0.0002679  0.0042333   0.063  0.94954
## TEAM_PITCHING_SO  0.0026423  0.0009393   2.813  0.00495 **
## TEAM_FIELDING_E -0.0203043  0.0024446  -8.306  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.32 on 2263 degrees of freedom
## Multiple R-squared:  0.2883, Adjusted R-squared:  0.2845
## F-statistic:  76.4 on 12 and 2263 DF,  p-value: < 2.2e-16
```

Model2:

For the second model, I used backward elimination process. I got rid of the variables that have the highest p-value. I stopped eliminating variables once all the variables have p-value less than 0.05, which making them statistically significant at α equals 0.5 level. Eventually, TEAM_PITCHING_BB, TEAM_PITCHING_HR, and TEAM_BATTING_BB were aliminated. The coefficient for each variable is shown in the following.

```
## 'data.frame':    2276 obs. of  13 variables:
##  $ TARGET_WINS     : int  39 70 86 70 82 75 80 85 86 76 ...
##  $ TEAM_BATTING_H  : int  1445 1339 1377 1387 1297 1279 1244 1273 1391 1271 ...
##  $ TEAM_BATTING_2B : int  194 219 232 209 186 200 179 171 197 213 ...
##  $ TEAM_BATTING_3B : int  39 22 35 38 27 36 54 37 40 18 ...
##  $ TEAM_BATTING_HR : int  13 190 137 96 102 92 122 115 114 96 ...
##  $ TEAM_BATTING_BB : int  143 685 602 451 472 443 525 456 447 441 ...
##  $ TEAM_BATTING_SO : num  842 1075 917 922 920 ...
##  $ TEAM_BASERUN_SB : int  101 37 46 43 49 107 80 40 69 72 ...
##  $ TEAM_PITCHING_H : int  9364 1347 1377 1396 1297 1279 1244 1281 1391 1271 ...
##  $ TEAM_PITCHING_HR: int  84 191 137 97 102 92 122 116 114 96 ...
##  $ TEAM_PITCHING_BB: int  927 689 602 454 472 443 525 459 447 441 ...
##  $ TEAM_PITCHING_SO: num  5456 1082 917 928 920 ...
##  $ TEAM_FIELDING_E : int  1011 193 175 164 138 123 136 112 127 131 ...
```

```
##
## 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_HR + TEAM_PITCHING_BB +
##     TEAM_PITCHING_SO + TEAM_FIELDING_E, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.474  -8.937   0.118   8.640  56.858
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.6020478  5.1785786   1.661  0.09684 .
## TEAM_BATTING_H    0.0481025  0.0037642  12.779  < 2e-16 ***
## TEAM_BATTING_2B  -0.0230727  0.0093442  -2.469  0.01361 *
## TEAM_BATTING_3B   0.0724491  0.0170933   4.238 2.34e-05 ***
## TEAM_BATTING_HR   0.0378938  0.0278312   1.362  0.17347
## TEAM_BATTING_BB   0.0045910  0.0059125   0.777  0.43753
## TEAM_BATTING_SO  -0.0050877  0.0025696  -1.980  0.04783 *
## TEAM_BASERUN_SB   0.0304943  0.0042949   7.100 1.66e-12 ***
## TEAM_PITCHING_H  -0.0008241  0.0003738  -2.205  0.02758 *
## TEAM_PITCHING_HR  0.0108311  0.0248386   0.436  0.66284
## TEAM_PITCHING_BB  0.0002679  0.0042333   0.063  0.94954
## TEAM_PITCHING_SO  0.0026423  0.0009393   2.813  0.00495 **
## TEAM_FIELDING_E  -0.0203043  0.0024446  -8.306  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.32 on 2263 degrees of freedom
## Multiple R-squared:  0.2883, Adjusted R-squared:  0.2845
## F-statistic:  76.4 on 12 and 2263 DF,  p-value: < 2.2e-16
```

```
##
## 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_HR + TEAM_PITCHING_SO +
##     TEAM_FIELDING_E, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.473  -8.938   0.124   8.637  56.852
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.5868474  5.1718675   1.660   0.0970 .
## TEAM_BATTING_H    0.0480898  0.0037580  12.797  < 2e-16 ***
## TEAM_BATTING_2B  -0.0230651  0.0093413  -2.469   0.0136 *
## TEAM_BATTING_3B   0.0724469  0.0170896   4.239 2.33e-05 ***
## TEAM_BATTING_HR   0.0370498  0.0244221   1.517   0.1294
## TEAM_BATTING_BB   0.0048987  0.0033639   1.456   0.1455
## TEAM_BATTING_SO  -0.0051316  0.0024734  -2.075   0.0381 *
## TEAM_BASERUN_SB   0.0305342  0.0042474   7.189 8.83e-13 ***
## TEAM_PITCHING_H  -0.0008135  0.0003342  -2.434   0.0150 *
## TEAM_PITCHING_HR  0.0116242  0.0214407   0.542   0.5878
## TEAM_PITCHING_SO  0.0026829  0.0006851   3.916 9.27e-05 ***
## TEAM_FIELDING_E  -0.0202975  0.0024418  -8.313  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.32 on 2264 degrees of freedom
## Multiple R-squared:  0.2883, Adjusted R-squared:  0.2849
## F-statistic: 83.38 on 11 and 2264 DF,  p-value: < 2.2e-16
```

```
## 
## 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, 
##     data = training)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -51.467  -8.931   0.106   8.611  56.900 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       8.1517183  5.1084165   1.596   0.1107    
## TEAM_BATTING_H    0.0482894  0.0037393  12.914  < 2e-16 ***
## TEAM_BATTING_2B  -0.0232074  0.0093362  -2.486   0.0130 *  
## TEAM_BATTING_3B   0.0737222  0.0169242   4.356 1.38e-05 ***
## TEAM_BATTING_HR   0.0492202  0.0096171   5.118 3.35e-07 ***
## TEAM_BATTING_BB   0.0048903  0.0033633   1.454   0.1461    
## TEAM_BATTING_SO  -0.0051034  0.0024725  -2.064   0.0391 *  
## TEAM_BASERUN_SB   0.0305335  0.0042468   7.190 8.77e-13 ***
## TEAM_PITCHING_H  -0.0007764  0.0003271  -2.374   0.0177 *  
## TEAM_PITCHING_SO  0.0027011  0.0006842   3.948 8.12e-05 ***
## TEAM_FIELDING_E  -0.0201977  0.0024344  -8.297  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 13.32 on 2265 degrees of freedom
## Multiple R-squared:  0.2882, Adjusted R-squared:  0.2851 
## F-statistic: 91.72 on 10 and 2265 DF,  p-value: < 2.2e-16
```

```
## 
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + 
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_SO + TEAM_BASERUN_SB + 
##     TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_E, data = training)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -52.037  -8.928   0.095   8.640  56.944 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      11.3751217  4.6034462   2.471   0.0135 *  
## TEAM_BATTING_H    0.0476514  0.0037144  12.829  < 2e-16 ***
## TEAM_BATTING_2B  -0.0219380  0.0092976  -2.360   0.0184 *  
## TEAM_BATTING_3B   0.0752126  0.0168973   4.451 8.96e-06 ***
## TEAM_BATTING_HR   0.0539718  0.0090470   5.966 2.82e-09 ***
## TEAM_BATTING_SO  -0.0058372  0.0024210  -2.411   0.0160 *  
## TEAM_BASERUN_SB   0.0325401  0.0040172   8.100 8.88e-16 ***
## TEAM_PITCHING_H  -0.0007876  0.0003271  -2.408   0.0161 *  
## TEAM_PITCHING_SO  0.0027052  0.0006844   3.953 7.96e-05 ***
## TEAM_FIELDING_E  -0.0217632  0.0021839  -9.965  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 13.32 on 2266 degrees of freedom
## Multiple R-squared:  0.2876, Adjusted R-squared:  0.2847 
## F-statistic: 101.6 on 9 and 2266 DF,  p-value: < 2.2e-16
```

Model 3:

For the third model, I used a different strategy. I use the build-in function called stepAIC to get the model using forward elimination method.

```
## Start:  AIC=11800.62
## 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
```

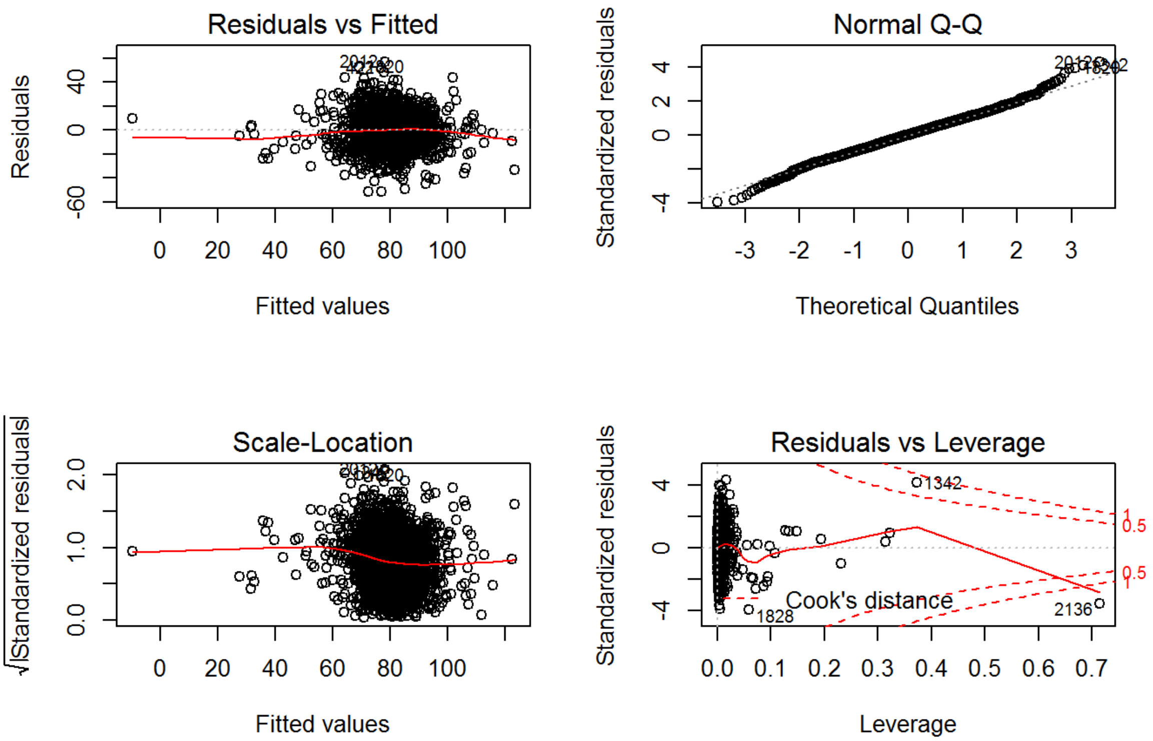
```
## 
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + 
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + 
##     TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_BB + 
##     TEAM_PITCHING_SO + TEAM_FIELDING_E, data = training)
## 
## Coefficients:
##      (Intercept)    TEAM_BATTING_H   TEAM_BATTING_2B   TEAM_BATTING_3B  
##        8.6020478         0.0481025        -0.0230727         0.0724491  
##  TEAM_BATTING_HR   TEAM_BATTING_BB   TEAM_BATTING_SO   TEAM_BASERUN_SB  
##        0.0378938         0.0045910        -0.0050877         0.0304943  
##  TEAM_PITCHING_H  TEAM_PITCHING_HR  TEAM_PITCHING_BB  TEAM_PITCHING_SO  
##       -0.0008241         0.0108311         0.0002679         0.0026423  
##  TEAM_FIELDING_E  
##       -0.0203043
```

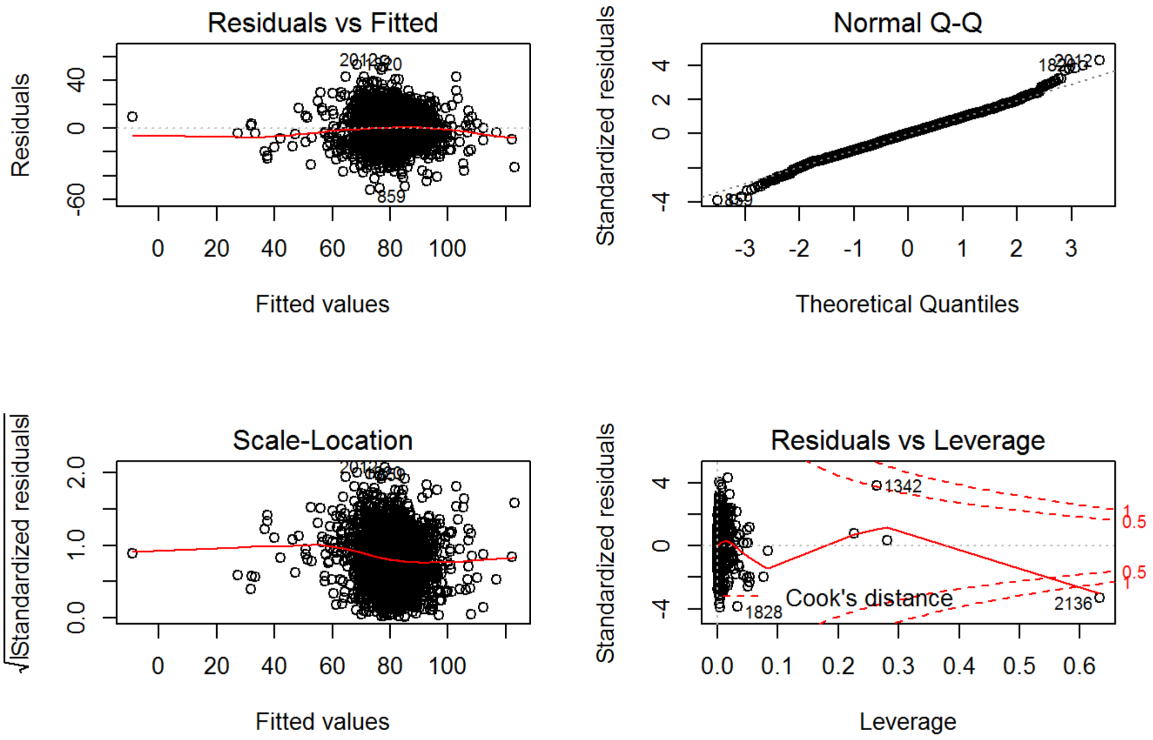
```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + 
##     TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB + 
##     TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_PITCHING_SO + 
##     TEAM_FIELDING_E
## 
## Final Model:
## 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
## 
## 
##   Step Df Deviance Resid. Df Resid. Dev      AIC
## 1                       2263   401739.2 11800.62
```

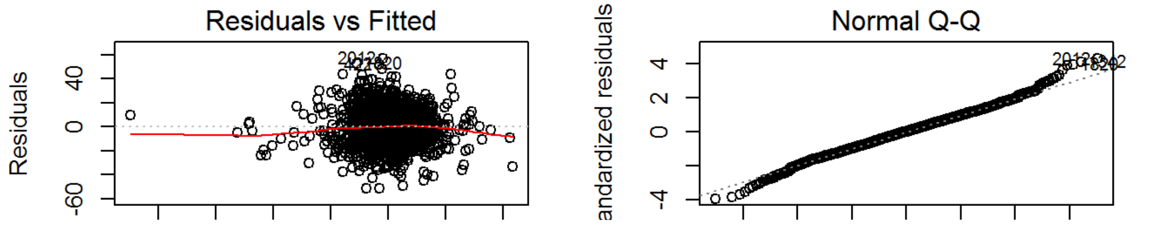
SELECT MODELS:

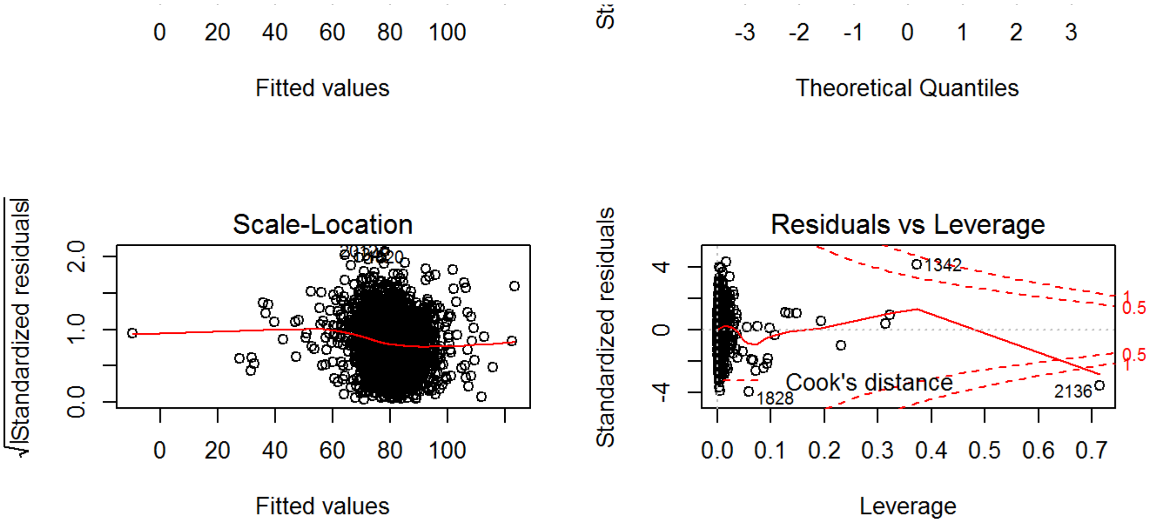
Decide on the criteria for selecting the best multiple linear regression model. Will you select a model with slightly worse performance if it makes more sense or is more parsimonious?

The best model should contains all the variables that have statistically significant correlation with the response variable. Model 1 and model 3 maybe more powerful than model 2, because their models include all the variables. One of the largest problem with powerful model has always been overfitting. Because of that, model 2 is the best model.


Residuals vs Fitted
Residuals
40
0
-60
0
20
40
60
80
100
Fitted values
2012
1820
Normal Q-Q
Standardized residuals
4
2
0
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles
2012
1820
Scale-Location
√|Standardized residuals|
2.0
1.0
0.0
0
20
40
60
80
100
Fitted values
Residuals vs Leverage
Standardized residuals
4
0
-4
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Leverage
1342
1828
2136
Cook's distance
1
0.5
0.5
1


Residuals vs Fitted
Residuals
40
0
-60
0
20
40
60
80
100
Fitted values
2012
1820
859
Normal Q-Q
Standardized residuals
4
2
0
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles
1820
2012
859
Scale-Location
√|Standardized residuals|
2.0
1.0
0.0
0
20
40
60
80
100
Fitted values
Residuals vs Leverage
Standardized residuals
4
0
-4
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Leverage
1342
1828
2136
Cook's distance
1
0.5
0.5
1


Residuals vs Fitted
Residuals
40
0
-60
2012
1820
Normal Q-Q
Standardized residuals
4
2
0
-4
2012
1820



For the evaluation step, I modified the evaluation data set, making it suitable for model 2. The summary statistics for the response variable looks similar to the original training data. Therefore, model 2 has been pretty accurate at predicting moneyball data sets.

```
##   INDEX TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## 1     9           1209             170              33              83
## 2    10           1221             151              29              88
## 3    14           1395             183              29              93
## 4    47           1539             309              29             159
## 5    60           1445             203              68               5
## 6    63           1431             236              53              10
##   TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_BASERUN_CS
## 1             447            1080              62              50
## 2             516             929              54              39
## 3             509             816              59              47
## 4             486             914             148              57
## 5              95             416              NA              NA
## 6             215             377              NA              NA
##   TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
## 1               NA            1209               83              447
## 2               NA            1221               88              516
## 3               NA            1395               93              509
## 4               42            1539              159              486
## 5               NA            3902               14              257
## 6               NA            2793               20              420
##   TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP
## 1             1080             140              156
## 2              929             135              164
## 3              816             156              153
## 4              914             124              154
## 5             1123             616              130
## 6              736             572              105
## 'data.frame':    259 obs. of  16 variables:
##  $ INDEX           : int  9 10 14 47 60 63 74 83 98 120 ...
##  $ TEAM_BATTING_H  : int  1209 1221 1395 1539 1445 1431 1430 1385 1259 1397 ...
##  $ TEAM_BATTING_2B : int  170 151 183 309 203 236 219 158 177 212 ...
##  $ TEAM_BATTING_3B : int  33 29 29 29 68 53 55 42 78 42 ...
##  $ TEAM_BATTING_HR : int  83 88 93 159 5 10 37 33 23 58 ...
##  $ TEAM_BATTING_BB : int  447 516 509 486 95 215 568 356 466 452 ...
##  $ TEAM_BATTING_SO : int  1080 929 816 914 416 377 527 609 689 584 ...
##  $ TEAM_BASERUN_SB : int  62 54 59 148 NA NA 365 185 150 52 ...
##  $ TEAM_BASERUN_CS : int  50 39 47 57 NA NA NA NA NA NA ...
##  $ TEAM_BATTING_HBP: int  NA NA NA 42 NA NA NA NA NA NA ...
##  $ TEAM_PITCHING_H : int  1209 1221 1395 1539 3902 2793 1544 1626 1342 1489 ...
##  $ TEAM_PITCHING_HR: int  83 88 93 159 14 20 40 39 25 62 ...
##  $ TEAM_PITCHING_BB: int  447 516 509 486 257 420 613 418 497 482 ...
##  $ TEAM_PITCHING_SO: int  1080 929 816 914 1123 736 569 715 734 622 ...
##  $ TEAM_FIELDING_E : int  140 135 156 124 616 572 490 328 226 184 ...
##  $ TEAM_FIELDING_DP: int  156 164 153 154 130 105 NA 104 132 145 ...
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   30.59   76.28   80.95   80.60   85.67  111.20
```

Appendix (Code)

library(MASS) library(knitr) library(ggplot2) library(psych)

moneyball_training_data <- read.csv(“C:/Users/blin261/Desktop/DATA621/moneyball-training-data.csv”) str(moneyball_training_data) training <- moneyball_training_data[, -1]

summary(training) sd(training$TARGET_{W}INS)hist(training$TARGET_WINS) qqnorm(training $TARGET_{W}INS)qqline(training$TARGET_WINS)

par(mfrow = c(3, 4)) for (i in 2:ncol(training)) { plot(training[,i], training$TARGET_WINS, xlab= colnames(training)[i], ylab = colnames(training)[1]) }

frequency <- c() total <- c() for (i in 1:ncol(training)) { frequency <- c(frequency, (sum(is.na(training[,i])))) total <- length(training[,i]) }

percentage <- round(frequency/total, 3) missing_values <- data.frame(colnames(training), frequency, percentage)

ggplot(data = missing_values, aes(x = reorder(colnames(training), percentage), y = percentage)) + geom_bar(stat=“identity”) + coord_flip()

par(mfrow = c(2, 3)) boxplot(training $TEAM_{B}ATTING_{H}BP, xlab =” \ TEAM_{B}ATTING_{H}BP \ ”, na.rm = TRUE)boxplot(training$TEAM_BASERUN_CS, xlab= “TEAM_BASERUN_CS”, na.rm = TRUE) boxplot(training $TEAM_{F}IELDING_{D}P, xlab =” \ TEAM_{F}IELDING_{D}P \ ”, na.rm = TRUE)boxplot(training$TEAM_BASERUN_SB, xlab= “TEAM_BASERUN_SB”, na.rm = TRUE) boxplot(training $TEAM_{P}ITCHING_{S}O, xlab =” \ TEAM_{P}ITCHING_{S}O \ ”, na.rm = TRUE)boxplot(training$TEAM_BATTING_SO, xlab= “TEAM_BATING_SO”, na.rm = TRUE)

training$TEAM_{B}ASERUN_{S}B[is.na(training$TEAM_BASERUN_SB)] <- median(training $TEAM_{B}ASERUN_{S}B, na.rm = TRUE)training$TEAM_PITCHING_SO[is.na(training$TEAM_PITCHING_SO)] <- median(training $TEAM_{P}ITCHING_{S}O, na.rm = TRUE)training$TEAM_BATTING_SO[is.na(training$TEAM_BATTING_SO)] <- median(training$TEAM_BATTING_SO, na.rm = TRUE)

training <- training[,-c(9,10,16)]

m_full <- lm(TARGET_WINS ~ ., data = training) summary(m_full) str(training) m_1 <- 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, data = training) summary(m_1)

m_2 <- 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_SO + TEAM_FIELDING_E, data = training) summary(m_2)

m_3 <- 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, data = training) summary(m_3)

m_4 <- lm (TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_E, data = training) summary(m_4)

step1 <- stepAIC(m_full, direction = “forward”) step1 step1$anova

par(mfrow = c(2,2)) plot(m_full) par(mfrow = c(2,2)) plot(m_4) par(mfrow = c(2,2)) plot(step1)

evaluation <- read.csv(“C:/Users/blin261/Desktop/DATA621/moneyball-evaluation-data.csv”) head(evaluation) str(evaluation) eval <- evaluation[,-c(1, 9, 10, 16)]

eval$TEAM_BASERUN_SB[is.na(eval$TEAM_BASERUN_SB)] <- median(eval $TEAM_BASERUN_SB, na.rm = TRUE)eval$TEAM_PITCHING_SO[is.na(eval$TEAM_PITCHING_SO)] <- median(eval $TEAM_PITCHING_SO, na.rm = TRUE)eval$TEAM_BATTING_SO[is.na(eval$TEAM_BATTING_SO)] <- median(eval$TEAM_BATTING_SO, na.rm = TRUE)

eval$TARGET_WINS < -predict(m_4, newdata = eval)summary(eval$TARGET_WINS)