DATA621 HW 1

Tyler Brown

2023-02-14

Overview In this homework assignment, you will explore, analyze and model a data set containing 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. Your objective is to build a multiple linear regression model on the training data to predict the number of wins for the team.

1. DATA EXPLORATION

Describe the size and the variables in the moneyball training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job.

Looking at the data provided, there are a total of 17 variables with 2276 records relevant to professional baseball teams. The 17 variables are each defined and evaluated based on impact on wins.

```
##                  DEFINITION                                THEORETICAL EFFECT
## INDEX            Indentification Variable (do not use)  None
## TARGET_WINS      Number of wins
## TEAM_BATTING_H   Base Hits by batters (1B,2B,3B,HR)     Positive Impact on Wins
## TEAM_BATTING_2B  Doubles by batters (2B)                Positive Impact on Wins
## TEAM_BATTING_3B  Triples by batters (3B)                Positive Impact on Wins
## TEAM_BATTING_4B  Homeruns by batters (4B)               Positive Impact on Wins
## TEAM_BATTING_BB  Walks by batters                       Positive Impact on Wins
## TEAM_BATTING_HBP Batters hit by pitch (get a free base) Positive Impact on Wins
## TEAM_BATTING_SO  Strikeouts by batters                  Negative Impact on Wins
## TEAM_BASERUN_SB  Stolen bases                           Positive Impact on Wins
## TEAM_BASERUN_CS  Caught stealing                        Negative Impact on Wins
## TEAM_FIELDING_E  Errors                                 Negative Impact on Wins
## TEAM_FIELDING_DP Double Plays                           Positive Impact on Wins
## TEAM_PITCHING_BB Walks allowed                          Negative Impact on Wins
## TEAM_PITCHING_H  Hits allowed                           Negative Impact on Wins
## TEAM_PITCHING_HR Homeruns allowed                       Negative Impact on Wins
## TEAM_PITCHING_SO Strikeouts by pitchers                 Positive Impact on Wins
```

We can see the minimum value, 1st and 3rd quantile, median value, average value (mean), and the maximum value for each variable.

```
##      INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
##  Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
##  1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
```

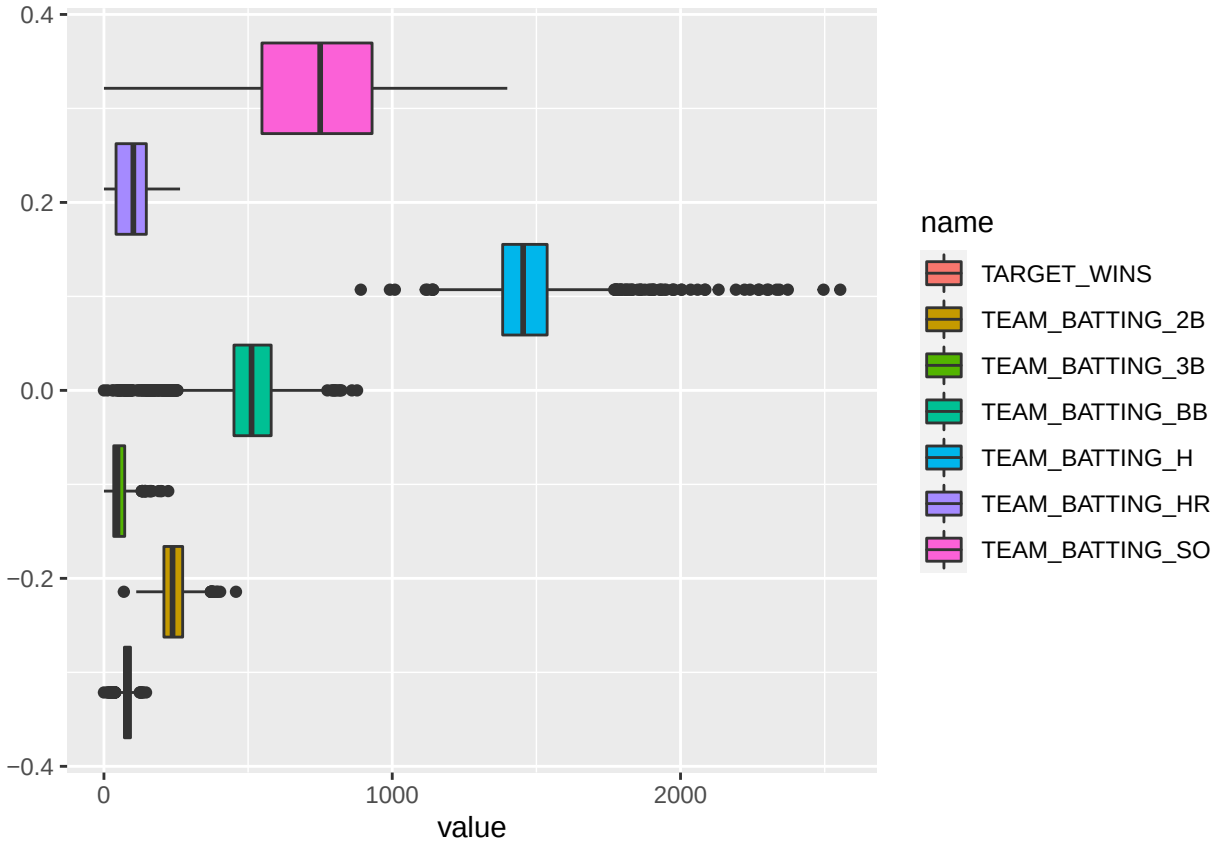
```
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
##  Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0
##
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0
##  Median : 47.00   Median :102.00   Median :512.0   Median : 750.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 735.6
##  3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
##                                                    NA's   :102
##  TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H
##  Min.   :  0.0   Min.   :  0.0   Min.   :29.00    Min.   : 1137
##  1st Qu.: 66.0   1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419
##  Median :101.0   Median : 49.0   Median :58.00    Median : 1518
##  Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779
##  3rd Qu.:156.0   3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682
##  Max.   :697.0   Max.   :201.0   Max.   :95.00    Max.   :30132
##  NA's   :131     NA's   :772     NA's   :2085
##  TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##  Min.   :  0.0    Min.   :   0.0   Min.   :    0.0   Min.   :  65.0
##  1st Qu.: 50.0    1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0
##  Median :107.0    Median : 536.5   Median :  813.5   Median : 159.0
##  Mean   :105.7    Mean   : 553.0   Mean   :  817.7   Mean   : 246.5
##  3rd Qu.:150.0    3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2
##  Max.   :343.0    Max.   :3645.0   Max.   :19278.0   Max.   :1898.0
##                                    NA's   :102
##  TEAM_FIELDING_DP
##  Min.   : 52.0
##  1st Qu.:131.0
##  Median :149.0
##  Mean   :146.4
##  3rd Qu.:164.0
##  Max.   :228.0
##  NA's   :286
```

The missing values are within the following variables and need to be addressed to make a predictive model:

```
##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
##                0                0                0                0
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##                0                0                0              102
##  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H
##              131              772             2085                0
## TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##                0                0              102                0
## TEAM_FIELDING_DP
##              286
```

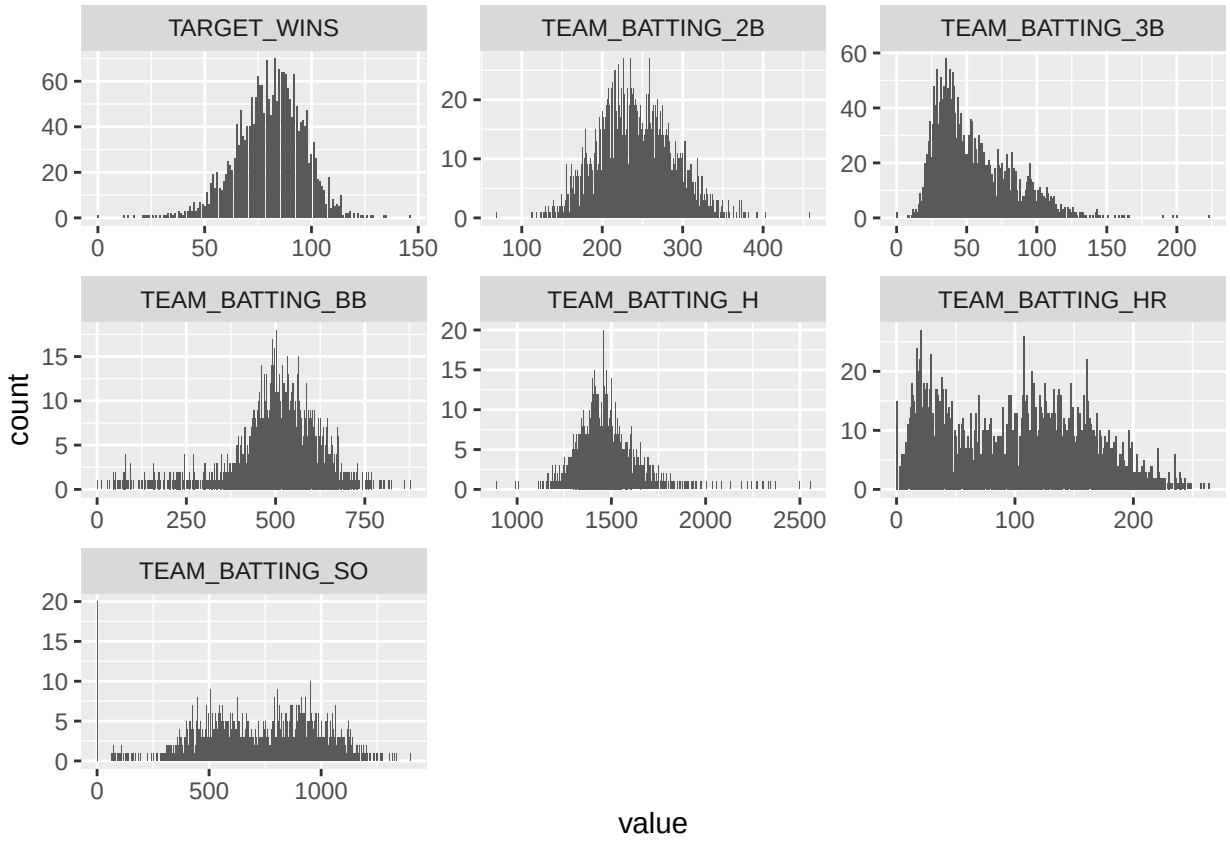
Here are boxplots of the variables in the data set. As we can see, the median, upper quartile, lower quartile, upper whisker, lower whisker, and outliers can be determined based on the plots.

```
## Warning: Removed 102 rows containing non-finite values (stat_boxplot).
```



Additionally, we can use a barplot to determine the count of each value for each variable.

```
## Warning: Removed 102 rows containing non-finite values (stat_count).
```



Let's determine the correlation of our target variable with each remaining variable, where values range from -1 (negative linear correlation) and 1 (positive linear correlation).

```
##                        [,1]
## INDEX           -0.02105643
## TEAM_BATTING_H   0.38876752
## TEAM_BATTING_2B  0.28910365
## TEAM_BATTING_3B  0.14260841
## TEAM_BATTING_HR  0.17615320
## TEAM_BATTING_BB  0.23255986
## TEAM_BATTING_SO          NA
## TEAM_BASERUN_SB          NA
## TEAM_BASERUN_CS          NA
## TEAM_BATTING_HBP         NA
## TEAM_PITCHING_H -0.10993705
## TEAM_PITCHING_HR 0.18901373
## TEAM_PITCHING_BB 0.12417454
## TEAM_PITCHING_SO         NA
## TEAM_FIELDING_E -0.17648476
## TEAM_FIELDING_DP         NA
```

2. Data Preparation

'Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this. Here are some possible transformations.

First, we need to address the missing values. From what we can recall, TEAM_BATTING_HBP have over 90% of missing values and should not be included in the model. Additionally, the INDEX variable has no relevance to the model and therefore will be removed as well. TEAM_BASERUN_CS is highly correlated with TEAM_BASERUN_SB and has a large amount of missing values. I will remove this variable from the model. In baseball, stolen bases can be derived from the batting and/or pitching rates. Therefore TEAM_BASERUN_SB can be removed from the model. The remaining variables (TEAM_BATTING_SO, TEAM_PITCHING_SO, TEAM_FIELDING_E, and TEAM_FIELDING_DP) will have their missing values replaced with the median values. This is, in my opinion, the best course of action because having a decimal value for each variable when they should be whole numbers does not make sense and will show in the model.

3. Build Models

Using the training data set, build at least three different multiple linear regression models, using different variables (or the same variables with different transformations). Since we have not yet covered automated variable selection methods, you should select the variables manually (unless you previously learned Forward or Stepwise selection, etc.). Since you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done.

The first multiple linear regression model is based on only batting variables.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO,
##     data = prep_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.644  -8.787   0.454   9.020  54.933
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -8.499099   5.023839  -1.692 0.090830 .
## TEAM_BATTING_H   0.044182   0.003702  11.933  < 2e-16 ***
## TEAM_BATTING_2B -0.015660   0.009321  -1.680 0.093104 .
## TEAM_BATTING_3B  0.099801   0.016369   6.097 1.27e-09 ***
## TEAM_BATTING_HR  0.031772   0.009378   3.388 0.000716 ***
## TEAM_BATTING_BB  0.028425   0.002805  10.135  < 2e-16 ***
## TEAM_BATTING_SO  0.007086   0.002184   3.244 0.001195 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.76 on 2269 degrees of freedom
## Multiple R-squared:  0.2391, Adjusted R-squared:  0.2371
## F-statistic: 118.8 on 6 and 2269 DF,  p-value: < 2.2e-16
```

Since TEAM_BATTING_2B has a p-value greater than 0.05, I will remove it from the model. This is a better model as all coefficients are positive, which means there is a positive correlation in relation to winning. Unfortunately, I was not expecting Batting Strike Outs being positively correlated to winnings, which does not make much sense.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B +
##     TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO, data = prep_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -65.329  -8.805   0.471   8.973  52.511
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -5.530298   4.704651  -1.175 0.239920
## TEAM_BATTING_H   0.040190   0.002840  14.149  < 2e-16 ***
## TEAM_BATTING_3B  0.104229   0.016162   6.449 1.37e-10 ***
## TEAM_BATTING_HR  0.031142   0.009374   3.322 0.000908 ***
## TEAM_BATTING_BB  0.027709   0.002773   9.992  < 2e-16 ***
## TEAM_BATTING_SO  0.006129   0.002109   2.906 0.003700 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.76 on 2270 degrees of freedom
## Multiple R-squared:  0.2382, Adjusted R-squared:  0.2365
## F-statistic: 141.9 on 5 and 2270 DF,  p-value: < 2.2e-16
```

The next model is made only off the pitching variables.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_PITCHING_H + TEAM_PITCHING_HR +
##     TEAM_PITCHING_BB + TEAM_PITCHING_SO, data = prep_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.284  -9.842   0.483   9.679  74.699
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      71.8253764  1.1577568  62.038  < 2e-16 ***
## TEAM_PITCHING_H  -0.0011826  0.0002476  -4.776 1.90e-06 ***
## TEAM_PITCHING_HR  0.0419494  0.0055003   7.627 3.52e-14 ***
## TEAM_PITCHING_BB  0.0197724  0.0022794   8.674  < 2e-16 ***
## TEAM_PITCHING_SO -0.0052582  0.0006818  -7.712 1.84e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.11 on 2271 degrees of freedom
## Multiple R-squared:  0.08207,    Adjusted R-squared:  0.08046
## F-statistic: 50.76 on 4 and 2271 DF,  p-value: < 2.2e-16
```

From the model, it seems pitching has little to no correlation to winning the game as the coefficients are close to 0, whether negative or positive.

The final model is based on fielding only.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_FIELDING_E + TEAM_FIELDING_DP,
##     data = prep_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.653  -9.992   0.632  10.038  74.737
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      91.060363   2.127771  42.796  < 2e-16 ***
## TEAM_FIELDING_E  -0.013370   0.001462  -9.143  < 2e-16 ***
## TEAM_FIELDING_DP -0.047535   0.013574  -3.502 0.000471 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.47 on 2273 degrees of freedom
## Multiple R-squared:  0.03635,    Adjusted R-squared:  0.0355
## F-statistic: 42.87 on 2 and 2273 DF,  p-value: < 2.2e-16
```

The fielding error variable has a negative correlation to winning the game, which makes sense. However, the fielding double play should have had a positive correlation, though it is close to 0.

Select Model

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? Discuss why you selected your model.

All three models have similar characteristics that would allow them to be implemented. The MSE and residual plots where fairly similar across the models. I have decided, however, to choose the batting multiple regression model because the F Stat and R-squared scores were significantly higher than the others. The F Stat explains the variability more than the other models and the R-squared explains better model fitting.

Implement Prediction to Evaluation Data

```
##          fit       lwr       upr
## 1   68.08942  66.64072  69.53811
## 2   69.29685  68.11673  70.47696
## 3   75.55903  74.59663  76.52142
## 4   83.36517  82.32353  84.40681
## 5   64.96974  62.94783  66.99165
## 6   66.08535  64.34267  67.82803
## 7   77.79507  76.53715  79.05300
## 8   69.13533  67.87089  70.39977
## 9   71.05055  69.69488  72.40622
```

```
## 10   72.90298  71.85225  73.95370
## 11   75.18580  74.02157  76.35003
## 12   82.34208  80.86427  83.81989
## 13   82.03545  80.20805  83.86284
## 14   79.02673  77.46249  80.59097
## 15   75.37674  74.01718  76.73629
## 16   76.59812  75.34915  77.84710
## 17   72.72117  71.69116  73.75118
## 18   82.39601  81.41707  83.37495
## 19         NA        NA        NA
## 20   91.54848  90.14901  92.94795
## 21   84.38659  83.04238  85.73080
## 22   86.54548  85.06329  88.02767
## 23   85.20318  84.08788  86.31847
## 24   75.15469  73.99257  76.31680
## 25   80.10412  78.94021  81.26803
## 26   82.71288  81.48628  83.93948
## 27   60.02618  56.75330  63.29906
## 28   74.42073  73.39384  75.44762
## 29   85.21959  83.62002  86.81916
## 30   75.46200  73.96388  76.96011
## 31   92.25479  90.84458  93.66499
## 32   86.95118  85.80719  88.09517
## 33   88.79667  87.49142  90.10191
## 34   91.83855  90.37263  93.30448
## 35   83.33418  82.42539  84.24297
## 36   83.68520  82.32668  85.04372
## 37   77.03349  76.23771  77.82927
## 38   90.00488  88.36854  91.64122
## 39   85.29554  84.08406  86.50701
## 40   88.03578  86.86580  89.20577
## 41   82.29417  81.06071  83.52763
## 42   87.17920  86.08031  88.27809
## 43   44.12510  39.93466  48.31554
## 44   98.88584  96.58767 101.18401
## 45   85.89691  84.92403  86.86980
## 46   93.22319  91.54780  94.89858
## 47   95.04292  93.61661  96.46922
## 48   72.19682  71.11182  73.28181
## 49   70.26227  69.07241  71.45213
## 50   75.14097  74.08225  76.19968
## 51   77.06988  76.06775  78.07200
## 52   83.53489  82.02354  85.04624
## 53   80.30853  79.21703  81.40002
## 54   72.74436  71.53790  73.95083
## 55   76.79893  75.92452  77.67334
## 56   75.60789  74.79366  76.42212
## 57   87.21737  85.82777  88.60697
## 58   68.52212  67.25246  69.79179
## 59         NA        NA        NA
## 60         NA        NA        NA
## 61   82.49638  81.45479  83.53797
## 62   84.02306  82.26443  85.78168
## 63   84.34261  83.42354  85.26169
```

```
## 64   83.64790  81.92827  85.36754
## 65   79.56597  78.12004  81.01189
## 66   86.02321  84.76045  87.28597
## 67   76.69226  75.65744  77.72708
## 68   81.53356  80.31551  82.75160
## 69         NA        NA        NA
## 70   86.21488  84.46725  87.96250
## 71   89.29295  87.64841  90.93749
## 72   74.60917  73.49096  75.72737
## 73   82.57260  81.30381  83.84140
## 74   84.95533  82.99559  86.91506
## 75   80.99296  79.41214  82.57378
## 76   86.30953  85.01612  87.60295
## 77   82.51679  81.55144  83.48213
## 78   79.10895  78.12165  80.09625
## 79         NA        NA        NA
## 80         NA        NA        NA
## 81   86.40439  85.06827  87.74051
## 82   90.19683  89.18432  91.20934
## 83   97.97194  96.44221  99.50166
## 84   82.70965  81.78967  83.62962
## 85   86.99759  85.92605  88.06914
## 86   77.31055  75.87592  78.74518
## 87   75.73002  74.64869  76.81136
## 88   81.12983  80.42372  81.83595
## 89   81.66588  80.32692  83.00485
## 90   89.21333  88.02150  90.40516
## 91   77.62391  76.57258  78.67525
## 92   94.50043  92.43066  96.57020
## 93   72.47554  71.09452  73.85657
## 94         NA        NA        NA
## 95         NA        NA        NA
## 96         NA        NA        NA
## 97   86.76518  84.86046  88.66990
## 98  100.27256  98.43955 102.10557
## 99   91.12815  89.63324  92.62306
## 100  92.39052  90.89663  93.88442
## 101  84.64627  83.82141  85.47112
## 102  74.42346  73.04834  75.79859
## 103  83.97725  83.11662  84.83788
## 104  81.59866  80.40856  82.78876
## 105  82.70115  81.24141  84.16089
## 106  76.78864  75.25292  78.32437
## 107  65.38717  63.41417  67.36017
## 108  81.52514  80.18784  82.86244
## 109  85.11409  84.09137  86.13680
## 110  68.92951  67.57369  70.28534
## 111  82.21495  81.26315  83.16674
## 112  80.45662  79.77893  81.13431
## 113  88.25270  87.34528  89.16012
## 114  85.94376  84.98162  86.90589
## 115  79.47902  78.57030  80.38773
## 116  80.74175  79.85312  81.63038
## 117  89.34286  88.24729  90.43843
```

```
## 118  80.24073  79.49758  80.98387
## 119  77.93898  76.79564  79.08232
## 120  72.97312  71.80593  74.14031
## 121  85.86022  84.43120  87.28923
## 122        NA        NA        NA
## 123        NA        NA        NA
## 124        NA        NA        NA
## 125  70.15680  68.72547  71.58812
## 126  82.94453  81.66910  84.21996
## 127  89.56987  88.39985  90.73988
## 128  73.79284  72.71109  74.87458
## 129  88.40188  87.33274  89.47102
## 130  93.92047  92.65171  95.18923
## 131  87.97776  86.77158  89.18394
## 132  79.16626  77.63084  80.70168
## 133  74.32412  73.20772  75.44052
## 134  83.61427  82.30577  84.92278
## 135  83.73923  82.49900  84.97946
## 136  69.78458  68.27238  71.29677
## 137  76.39372  75.50268  77.28476
## 138  75.92216  75.00281  76.84151
## 139  78.96605  77.86885  80.06325
## 140  79.37017  78.40316  80.33719
## 141  65.02154  63.42657  66.61652
## 142        NA        NA        NA
## 143  93.40640  92.01065  94.80215
## 144  81.02677  80.23983  81.81371
## 145  75.95016  74.64104  77.25929
## 146  76.21359  75.30355  77.12362
## 147  80.91402  80.07156  81.75647
## 148  82.01046  80.99287  83.02806
## 149  83.62807  82.86302  84.39312
## 150  80.21488  79.40566  81.02411
## 151  82.62385  81.31041  83.93729
## 152  79.98538  78.67536  81.29539
## 153  60.31141  56.98412  63.63871
## 154  71.54299  70.31605  72.76993
## 155  76.43235  75.32176  77.54295
## 156  71.87775  70.71345  73.04205
## 157  85.73202  84.45313  87.01091
## 158  72.94072  71.66300  74.21844
## 159  90.69042  88.87673  92.50411
## 160        NA        NA        NA
## 161 105.10975 102.83732 107.38219
## 162 104.70676 102.67172 106.74180
## 163  91.14299  89.81007  92.47591
## 164 105.37722 103.25832 107.49613
## 165  98.49251  96.60207 100.38296
## 166  91.05878  89.30814  92.80943
## 167  85.80615  84.76832  86.84398
## 168  80.12362  78.68624  81.56101
## 169  72.28575  71.08617  73.48533
## 170  80.56660  79.55834  81.57486
## 171        NA        NA        NA
```

```
## 172  83.70763  82.57633  84.83892
## 173  81.46669  80.41953  82.51385
## 174  90.07446  88.60931  91.53961
## 175  83.79660  82.94920  84.64400
## 176  78.86884  77.59812  80.13956
## 177  79.94570  78.26951  81.62188
## 178  77.33317  76.58710  78.07925
## 179  76.69705  75.89659  77.49751
## 180  81.20086  80.30876  82.09295
## 181  76.35058  75.39064  77.31053
## 182  84.56267  83.41446  85.71089
## 183  82.60796  81.64134  83.57459
## 184  85.10092  83.94347  86.25838
## 185  99.04875  96.48779 101.60970
## 186  87.00154  85.75493  88.24815
## 187  91.58599  90.00473  93.16725
## 188  69.98346  68.53005  71.43686
## 189  66.12218  64.82235  67.42200
## 190 106.76010 104.13804 109.38216
## 191        NA        NA        NA
## 192        NA        NA        NA
## 193  73.20437  72.01471  74.39403
## 194  77.08226  75.90128  78.26325
## 195  80.55491  79.02408  82.08574
## 196  69.54963  68.04272  71.05654
## 197  76.21057  75.31578  77.10536
## 198  82.65042  81.21464  84.08620
## 199  80.72763  79.61255  81.84271
## 200  87.19532  86.24386  88.14677
## 201  80.83480  79.56419  82.10541
## 202  81.48662  80.56821  82.40503
## 203  77.26073  75.91779  78.60366
## 204  82.24614  81.25423  83.23804
## 205  76.62681  75.69447  77.55915
## 206  80.67658  79.71049  81.64268
## 207  81.49961  80.21715  82.78206
## 208  78.30078  77.24688  79.35468
## 209  81.48989  80.63939  82.34038
## 210  77.69007  76.57125  78.80889
## 211 102.09699  99.64678 104.54720
## 212  91.88123  90.13394  93.62852
## 213  83.68423  81.98636  85.38210
## 214  70.41563  69.27764  71.55362
## 215  75.31877  74.17984  76.45771
## 216  86.65858  85.87931  87.43785
## 217  84.62488  83.49822  85.75155
## 218  85.36282  84.43411  86.29152
## 219  75.00080  74.09362  75.90798
## 220  78.13146  77.23682  79.02610
## 221  80.82165  79.63426  82.00905
## 222  74.90177  73.52816  76.27538
## 223  85.41220  84.36683  86.45757
## 224  78.90019  77.63416  80.16623
## 225  93.08378  89.28187  96.88568
```

```
## 226  75.76272  74.88305  76.64240
## 227  78.29652  77.37794  79.21509
## 228  83.89904  82.59321  85.20487
## 229  82.10592  81.20322  83.00862
## 230  81.15634  79.74136  82.57132
## 231        NA        NA        NA
## 232  90.48406  89.23190  91.73622
## 233  83.84944  82.52407  85.17481
## 234  84.33990  82.93444  85.74536
## 235  79.76023  79.00411  80.51635
## 236  73.88387  72.99846  74.76928
## 237  81.54950  80.14457  82.95443
## 238  76.93411  75.84156  78.02666
## 239  93.31488  91.10435  95.52541
## 240  72.67397  71.51547  73.83247
## 241  88.81799  87.81270  89.82329
## 242  86.80489  85.73392  87.87586
## 243  82.85278  81.54910  84.15646
## 244  81.93886  81.09569  82.78202
## 245  64.98955  63.29840  66.68071
## 246  83.54372  82.09603  84.99140
## 247  76.67608  75.69917  77.65300
## 248  82.74580  81.60182  83.88979
## 249  72.99194  71.93834  74.04554
## 250  83.86636  82.48596  85.24676
## 251  84.27467  82.98045  85.56889
## 252  63.70279  61.55413  65.85145
## 253  92.86270  91.39385  94.33155
## 254  47.72203  44.22261  51.22144
## 255  69.00722  67.85843  70.15602
## 256  78.83549  77.43877  80.23220
## 257  75.81659  74.80303  76.83016
## 258  78.97870  78.27716  79.68024
## 259  78.84236  77.49893  80.18578
```

Appendix

```
library(tidyverse)

df = read.csv("https://raw.githubusercontent.com/AlphaCurse/DATA621/main/moneyball-training-data.csv")
dict <- matrix(c("Indentification Variable (do not use)","None","Number of wins","","Base Hits by batte
colnames(dict) <- c("DEFINITION","THEORETICAL EFFECT")
rownames(dict) <- c("INDEX","TARGET_WINS","TEAM_BATTING_H","TEAM_BATTING_2B","TEAM_BATTING_3B","TEAM_BA

dict <- as.table(dict)
dict
summary(df)
colSums(is.na(df))
plot_df = pivot_longer(df, c("TARGET_WINS","TEAM_BATTING_H","TEAM_BATTING_2B","TEAM_BATTING_3B","TEAM_B

ggplot(plot_df, aes(x=value, fill=name)) +
  geom_boxplot()
```

```
ggplot(plot_df, aes(x=value)) +
  geom_bar() +
  facet_wrap(name ~ ., scales = "free")
cor(df[ ,colnames(df) != "TARGET_WINS"],
    df$TARGET_WINS)
prep_df = df
prep_df=subset(prep_df, select= (-TEAM_BATTING_HBP))
prep_df=subset(prep_df, select= (-INDEX))
prep_df=subset(prep_df, select= (-TEAM_BASERUN_CS))
prep_df=subset(prep_df, select= (-TEAM_BASERUN_SB))
prep_df$TEAM_BATTING_SO[is.na(prep_df$TEAM_BATTING_SO)]=median(prep_df$TEAM_BATTING_SO, na.rm=TRUE)
prep_df$TEAM_PITCHING_SO[is.na(prep_df$TEAM_PITCHING_SO)]=median(prep_df$TEAM_PITCHING_SO, na.rm=TRUE)
prep_df$TEAM_FIELDING_DP[is.na(prep_df$TEAM_FIELDING_DP)]=median(prep_df$TEAM_FIELDING_DP, na.rm=TRUE)
prep_df$TEAM_FIELDING_E[is.na(prep_df$TEAM_FIELDING_E)]=median(prep_df$TEAM_FIELDING_E, na.rm=TRUE)
bm1 = lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTI
summary(bm1)
bm2 = lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTI
summary(bm2)
pm1 = lm(TARGET_WINS ~ TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_PITCHING_SO, data=pr
summary(pm1)
fm1 = lm(TARGET_WINS ~ TEAM_FIELDING_E + TEAM_FIELDING_DP, data=prep_df)
summary(fm1)
eval_data = read.csv("https://raw.githubusercontent.com/AlphaCurse/DATA621/main/moneyball-evaluation-dat

predict(bm2, newdata = eval_data, interval='confidence')
```