Data 621 Homework 1

Anthony Pagan

September 25, 2019

Table of Contents

Data Exploration

In this exercise we will go with 2 approaches. One approach would be to remove data with NA values and the second approach would be to replace the NA data with a value. We will attempt both approaches and use the one with the best predictions.

Data Explore - Replace NA Values

The initial review of the data shows 6 columns with incomplete data for 6 Columns

```
## [1] 2276

##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
##               NA               NA               NA               NA
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##               NA               NA               NA  "NA's   :102  "
##  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H
##  "NA's   :131  "  "NA's   :772  " "NA's   :2085  "               NA
## TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
```

```
##               NA               NA  "NA's   :102  "               NA
## TEAM_FIELDING_DP
##  "NA's   :286  "
```

THis is a summary of values for each column that has NA data values. For the most part the mean and median values are close enough to theorize that data of the six columns with NA value are fairly normal. We will attempt a replacement as one approachin in the data preparation section.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0   548.0   750.0   735.6   930.0  1399.0     102

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    66.0   101.0   124.8   156.0   697.0     131

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    38.0    49.0    52.8    62.0   201.0     772

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   29.00   50.50   58.00   59.36   67.00   95.00    2085

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0   615.0   813.5   817.7   968.0 19278.0     102

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    52.0   131.0   149.0   146.4   164.0   228.0     286
```

Data Explore - Remove NA Values

If we remove all rows with incomplete rows, there will be a total of 191 rows. We need to decide if using only 0.09% of the data suffice

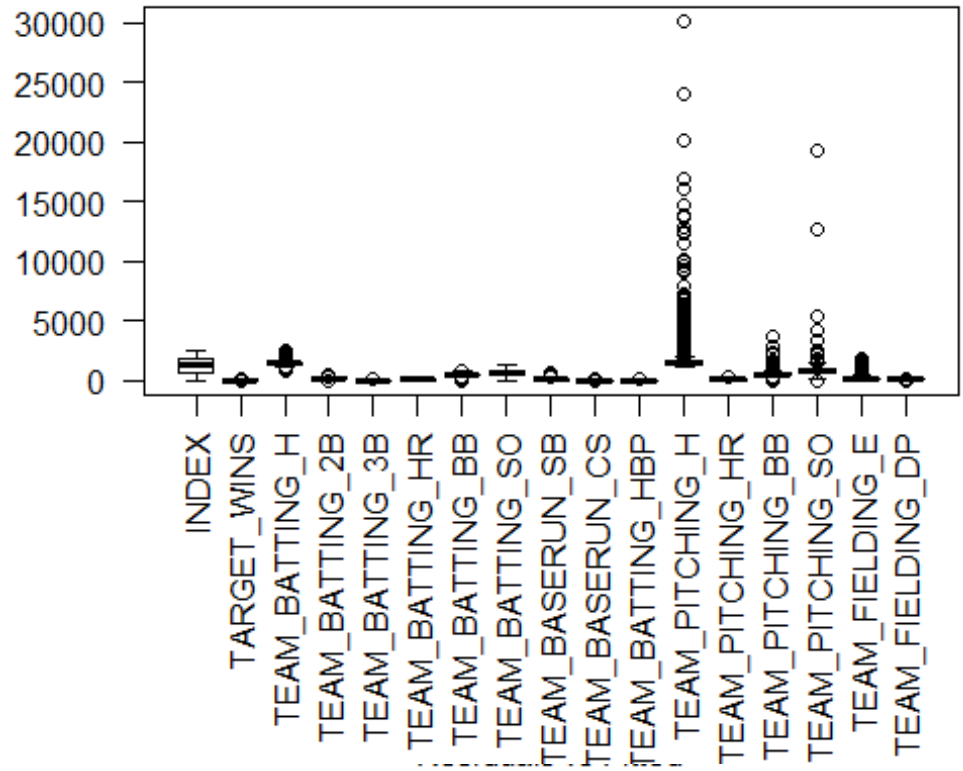
```
##    Mode   FALSE    TRUE
## logical    2085     191
```

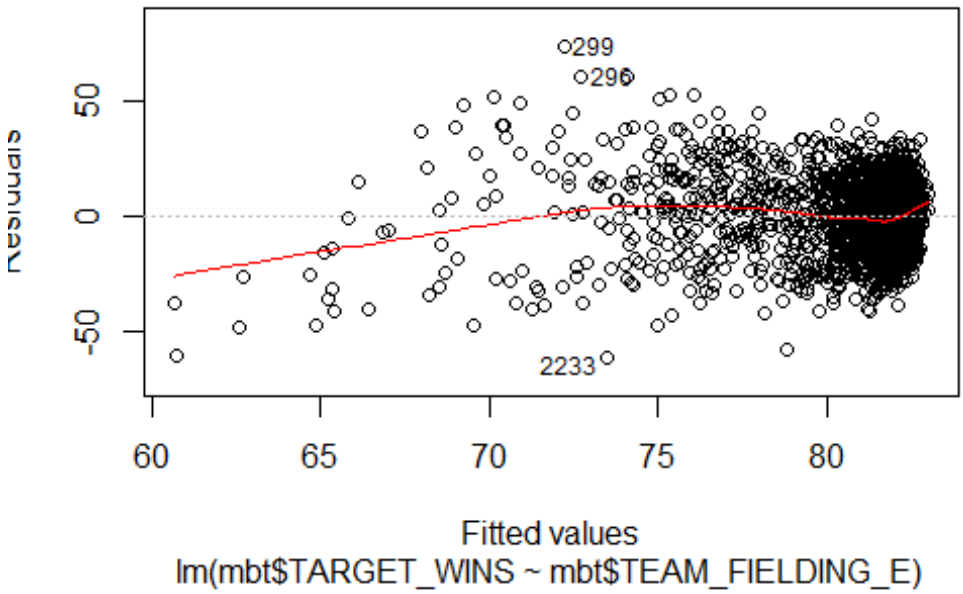
A look at the mean and median values show a larger spread in TB_SO, TP_H and TP_E. The expectation is there will be more outliers in these groups

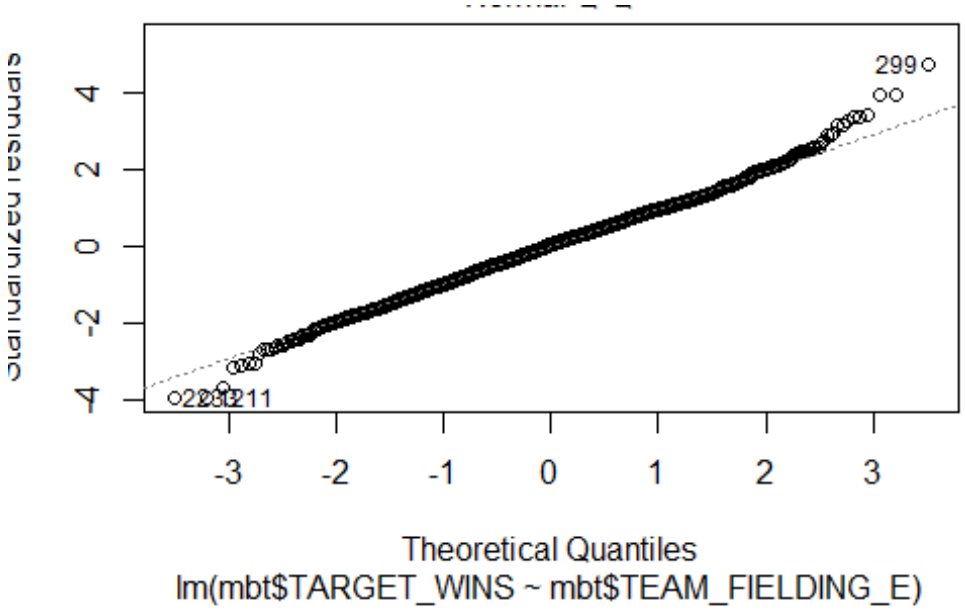
```
##      INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Median : 47.00   Median :102.00   Median :512.0   Median : 750.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 735.6
##  TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H
##  Median :101.0   Median : 49.0   Median :58.00    Median : 1518
##  Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779
##  TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##  Median :107.0    Median : 536.5   Median :  813.5   Median : 159.0
##  Mean   :105.7    Mean   : 553.0   Mean   :  817.7   Mean   : 246.5
##  TEAM_FIELDING_DP
```

```
##  Median :149.0
##  Mean   :146.4
```

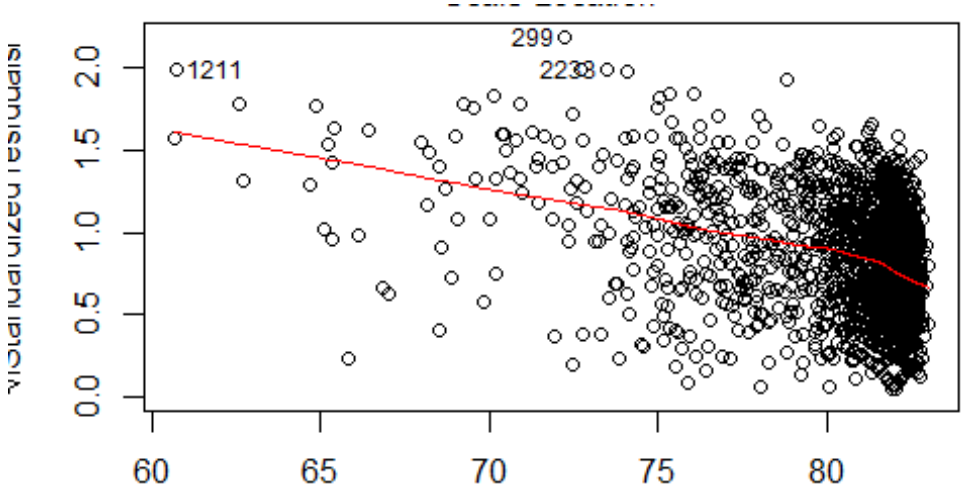
The box plots below confirms the outliers as expected ,but TP_E is not as drastic as the others. However te


30000
25000
20000
15000
10000
5000
0
INDEX
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_BATTING_HBP
TEAM_PITCHING_H
TEAM_PITCHING_HR
TEAM_PITCHING_BB
TEAM_PITCHING_SO
TEAM_FIELDING_E
TEAM_FIELDING_DP

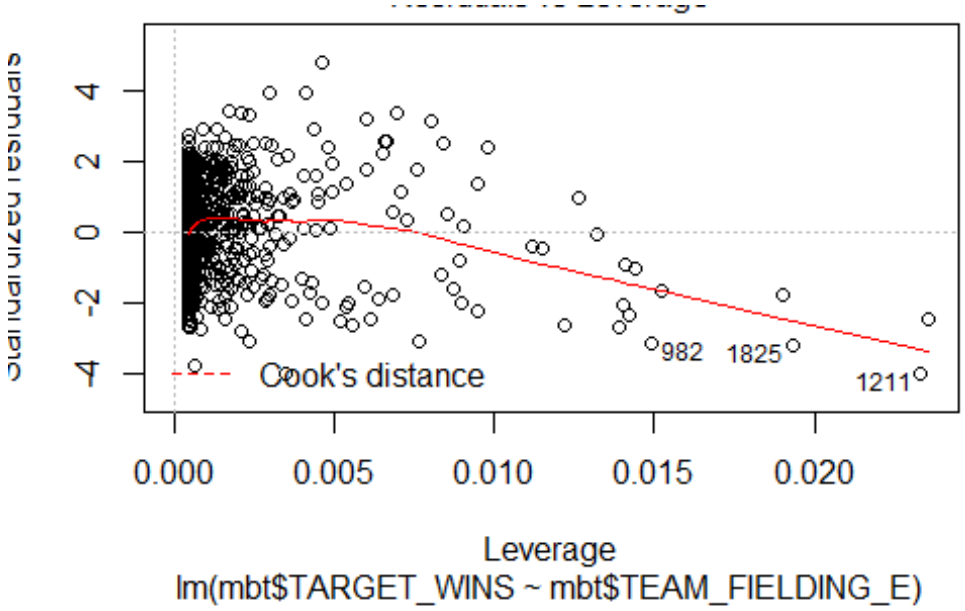

Residuals
50
0
-50
299
296
2233
60
65
70
75
80
Fitted values
lm(mbt$TARGET_WINS ~ mbt$TEAM_FIELDING_E)

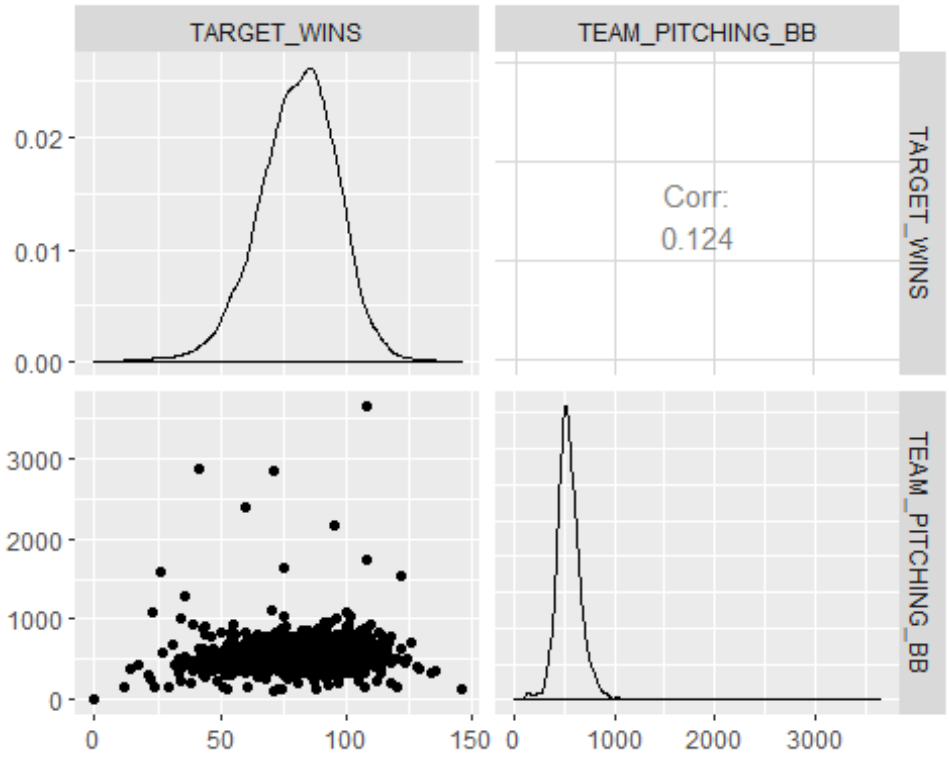


Theoretical Quantiles
lm(mbt$TARGET_WINS ~ mbt$TEAM_FIELDING_E)



Fitted values
lm(mbt$TARGET_WINS ~ mbt$TEAM_FIELDING_E)

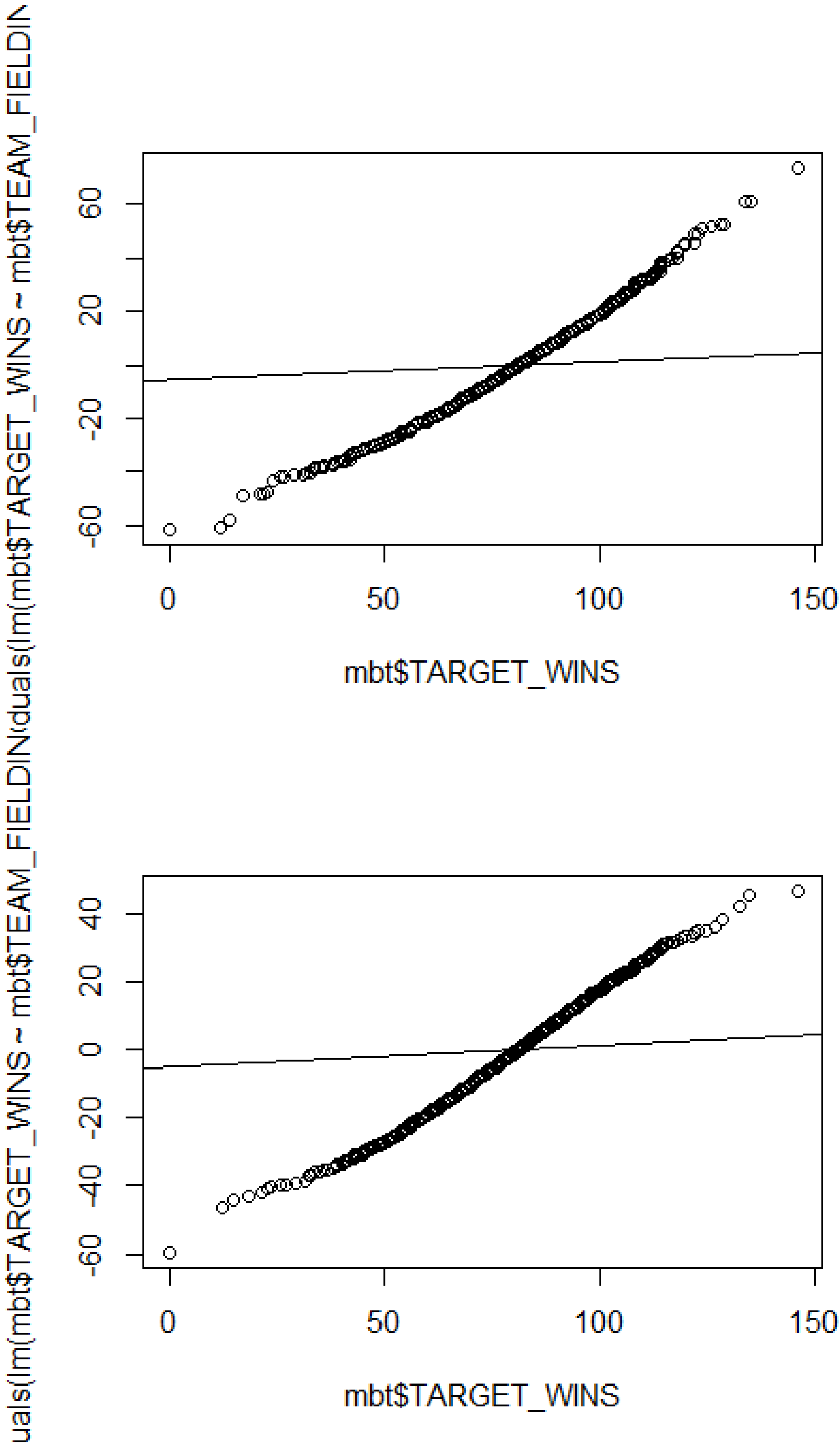




An initial view of the data show that TEAM_FIELDING_E and TEAM_FIELDING_DP have low P value and may have high correlation with team wins.

```
##
## Call:
## lm(formula = TARGET_WINS ~ . - INDEX, data = mbt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8708  -5.6564  -0.0599   5.2545  22.9274
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      60.28826   19.67842   3.064  0.00253 **
## TEAM_BATTING_H    1.91348    2.76139   0.693  0.48927
## TEAM_BATTING_2B   0.02639    0.03029   0.871  0.38484
## TEAM_BATTING_3B  -0.10118    0.07751  -1.305  0.19348
## TEAM_BATTING_HR  -4.84371   10.50851  -0.461  0.64542
## TEAM_BATTING_BB  -4.45969    3.63624  -1.226  0.22167
## TEAM_BATTING_SO   0.34196    2.59876   0.132  0.89546
## TEAM_BASERUN_SB   0.03304    0.02867   1.152  0.25071
## TEAM_BASERUN_CS  -0.01104    0.07143  -0.155  0.87730
## TEAM_BATTING_HBP  0.08247    0.04960   1.663  0.09815 .
## TEAM_PITCHING_H  -1.89096    2.76095  -0.685  0.49432
## TEAM_PITCHING_HR  4.93043   10.50664   0.469  0.63946
## TEAM_PITCHING_BB  4.51089    3.63372   1.241  0.21612
## TEAM_PITCHING_SO -0.37364    2.59705  -0.144  0.88577
## TEAM_FIELDING_E  -0.17204    0.04140  -4.155 5.08e-05 ***
## TEAM_FIELDING_DP -0.10819    0.03654  -2.961  0.00349 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.467 on 175 degrees of freedom
##   (2085 observations deleted due to missingness)
## Multiple R-squared:  0.5501, Adjusted R-squared:  0.5116
## F-statistic: 14.27 on 15 and 175 DF,  p-value: < 2.2e-16
```

The qq dot plots do not follow the residual line and fails the normality test


mbt$TARGET_WINS
mbt$TARGET_WINS
duals(lm(mbt$TARGET_WINS ~ mbt$TEAM_FIELDIN

Data Preparation

As a start, we begin by redjusting the data column headings to shorter column names.

```
## '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 127
1 ...
##  $ 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 127
1 ...
##  $ 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 ...
```

Data Prepare - Replace NA values

In our analysis of the summbary of columns with NA values we noted that median and mean values were close enough to theorize that these column values were fairly normal. As a result, we replace any NAs with the mean value of the column data.

```
##      Index            Wins          TB_Hits         TB_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  
##      TB_3B            TB_HR            TB_BB           TB_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.: 556.8  
##  Median : 47.00   Median :102.00   Median :512.0   Median : 735.6  
##  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.: 925.0  
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0  
##      TBR_SB           TBR_CS           TB_HBP           TP_H       
##  Min.   :  0.0   Min.   :  0.00   Min.   :29.00   Min.   : 1137  
##  1st Qu.: 67.0   1st Qu.: 44.00   1st Qu.:59.36   1st Qu.: 1419  
##  Median :106.0   Median : 52.80   Median :59.36   Median : 1518  
##  Mean   :124.8   Mean   : 52.80   Mean   :59.36   Mean   : 1779
```

```
##  3rd Qu.:151.0   3rd Qu.: 54.25   3rd Qu.:59.36   3rd Qu.: 1682
##  Max.   :697.0   Max.   :201.00   Max.   :95.00   Max.   :30132
##      TP_HR           TP_BB            TP_SO             TP_E
##  Min.   :  0.0   Min.   :   0.0   Min.   :    0.0   Min.   :  65.0
##  1st Qu.: 50.0   1st Qu.: 476.0   1st Qu.:  626.0   1st Qu.: 127.0
##  Median :107.0   Median : 536.5   Median :  817.7   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.:  957.0   3rd Qu.: 249.2
##  Max.   :343.0   Max.   :3645.0   Max.   :19278.0   Max.   :1898.0
##      TP_DP
##  Min.   : 52.0
##  1st Qu.:134.0
##  Median :146.4
##  Mean   :146.4
##  3rd Qu.:161.2
##  Max.   :228.0
```

Data Prepart - Remove NA values

In this next approach we are removing columns with missing data. We run a linear model with reduced columns and look at the corrrelation charts.

```
## 'data.frame':    2276 obs. of  10 variables:
##  $ Wins   : int  39 70 86 70 82 75 80 85 86 76 ...
##  $ TB_Hits: int  1445 1339 1377 1387 1297 1279 1244 1273 1391 1271 ...
##  $ TB_2B  : int  194 219 232 209 186 200 179 171 197 213 ...
##  $ TB_3B  : int  39 22 35 38 27 36 54 37 40 18 ...
##  $ TB_HR  : int  13 190 137 96 102 92 122 115 114 96 ...
##  $ TB_BB  : int  143 685 602 451 472 443 525 456 447 441 ...
##  $ TP_H   : int  9364 1347 1377 1396 1297 1279 1244 1281 1391 1271 ...
##  $ TP_HR  : int  84 191 137 97 102 92 122 116 114 96 ...
##  $ TP_BB  : int  927 689 602 454 472 443 525 459 447 441 ...
##  $ TP_E   : int  1011 193 175 164 138 123 136 112 127 131 ...
```

Now we will run the linear model again with only the columns with complete data.

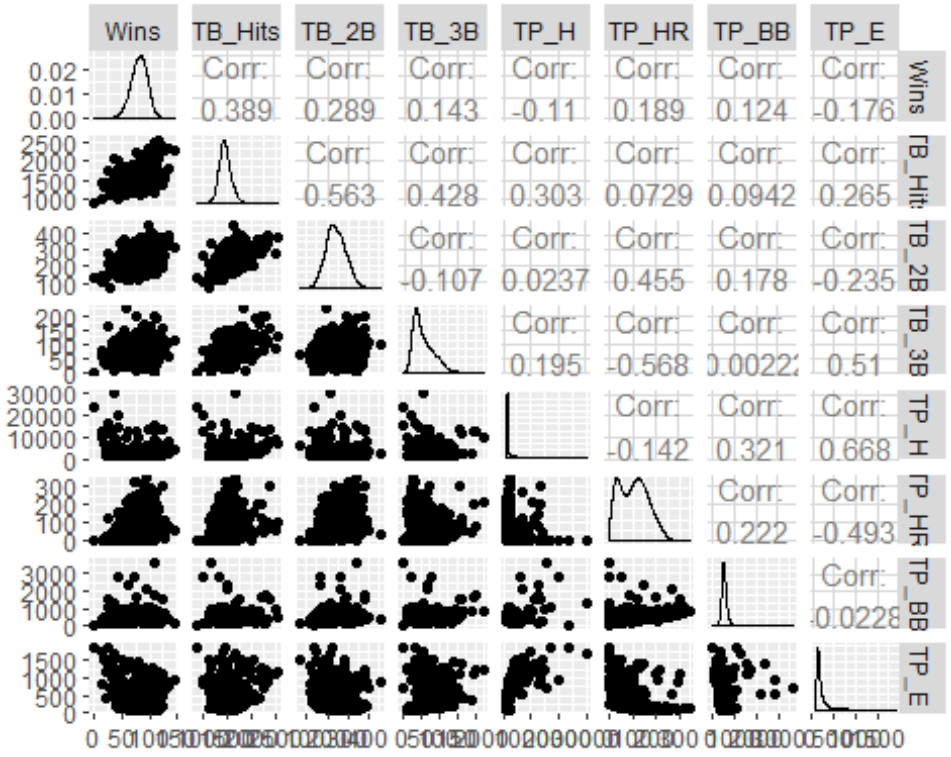
```
##
## Call:
## lm(formula = Wins ~ ., data = mbt2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.423  -8.867   0.115   8.887  55.548
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.738568   3.511940   1.919 0.055140 .
## TB_Hits      0.048908   0.003251  15.045  < 2e-16 ***
## TB_2B       -0.026239   0.009073  -2.892 0.003865 **
## TB_3B        0.102433   0.016734   6.121 1.09e-09 ***
```

```
## TB_HR        0.057039   0.026548   2.149 0.031778 *
## TB_BB       -0.001320   0.004840  -0.273 0.785147
## TP_H        -0.001329   0.000369  -3.602 0.000323 ***
## TP_HR       -0.019072   0.023835  -0.800 0.423689
## TP_BB        0.011387   0.003085   3.691 0.000228 ***
## TP_E        -0.016523   0.002373  -6.963 4.34e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.48 on 2266 degrees of freedom
## Multiple R-squared:  0.2703, Adjusted R-squared:  0.2674
## F-statistic: 93.24 on 9 and 2266 DF,  p-value: < 2.2e-16
```

The linear model shows the P values of TB_BB and TB_HR are greater than .05 so we can remove 2 columns and rerun the model.

```
##
## Call:
## lm(formula = Wins ~ ., data = mbt3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.205  -8.802   0.106   8.991  54.965
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.515149   3.305595   2.576   0.0101 *
## TB_Hits      0.048423   0.003207  15.098  < 2e-16 ***
## TB_2B       -0.024217   0.009019  -2.685   0.0073 **
## TB_3B        0.092961   0.016187   5.743 1.06e-08 ***
## TP_H        -0.001367   0.000325  -4.206 2.70e-05 ***
## TP_HR        0.030342   0.006918   4.386 1.21e-05 ***
## TP_BB        0.009720   0.001983   4.902 1.01e-06 ***
## TP_E        -0.017504   0.002229  -7.854 6.15e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.49 on 2268 degrees of freedom
## Multiple R-squared:  0.2687, Adjusted R-squared:  0.2664
## F-statistic:   119 on 7 and 2268 DF,  p-value: < 2.2e-16
```

Last we use ggally to get an idea of correlation of the data and run a corr test to get raw correlation statistics. TB_Hits and TB_2B show the highest correlation and graphs show a positive correlation.




```
##       Wins    TB_Hits      TB_2B      TB_3B      TB_HR      TB_BB
##  1.0000000  0.3887675  0.2891036  0.1426084  0.1761532  0.2325599
##       TP_H      TP_HR      TP_BB       TP_E
## -0.1099371  0.1890137  0.1241745 -0.1764848
```

Build Models

For all of the linear models we are extracting the coeficients, r squared values, adjusted r squared, sigma and f statistics. Coeficients provide y intercept, slope ,the t value which gives the standard deviations the estimated coefficients are from zero and p value which gives probability the null hypothesis is true. The multiple r-squred and adjusted r squared lets us know how close our data are to the linear regression model. The F-statistic gives us the relationship between dependent and independent variables. A large F-statistics means a strong relationship.

Our first model uses the data set columns with complete data. The P value is very low. The Rsqured and Adjusted RSqaured values are below .5 and F Statistic is low

- Coefficients: 20.9497811, 6.8736524, 3.0478383, 0.0023316
- RSquared: 0.3192196
- Adjusted RSquared: 0.3147011
- Sigma: 13.0400699
- FStatistic: 70.6479608, 15, 2260

The next model uses the data set columns with columns with high pvalues re data. The P value is als very low. The Rsqured and Adjusted RSqaured values are below .5 and F Statistic is higher

- Coefficients: 23.6666983, 5.2220414, 4.5320779, 6.144647610^{-6}
- RSquared: 0.3186326
- Adjusted RSquared: 0.3153221
- Sigma: 13.0341605
- FStatistic: 96.2482041, 11, 2264 ##Data Prepare - Remove NA Values

This next approach removes NA columns. Our first model uses the data set columns wiht complete data. The P value is slightly above .05. The Rsqured and Adjusted RSqaured values are below .5 and F Statistic is low

- Coefficients: 6.7385676, 3.5119403, 1.9187592, 0.0551403
- RSquared: 0.2702515
- Adjusted RSquared: 0.2673532
- Sigma: 13.4830221
- FStatistic: 93.2421767, 9, 2266

Our next model removes TB_BB and TB_HR from previou dataset. The P value is lower at .01. The Rsqured and Adjusted RSqaured values are below .5 and F Statistic is higher than previous dataset. This tells us the new dataset has a lower probability of null hypothesis being true.

- Coefficients: 13.0083989, 3.1370647, 4.1466786, 3.497324210^{-5}
- RSquared: 0.2509174
- Adjusted RSquared: 0.2492675
- Sigma: 13.6484246
- FStatistic: 152.0747037, 5, 2270

Select Models

Select Models - Replace NA Values

In the below models results show comparison of data wih replace NA values. The GGplot below shows a tight cluster with a straight linear line for the NA replacement data.

```
##   Id PredictedWins TargetWins
## 1  1      61.07615         39
## 2  2      76.57526         70
## 3  3      76.21630         86
## 4  4      72.62603         70
## 5  5      68.01394         82
## 6  6      70.35663         75
```

Select Models - Remove NA Values

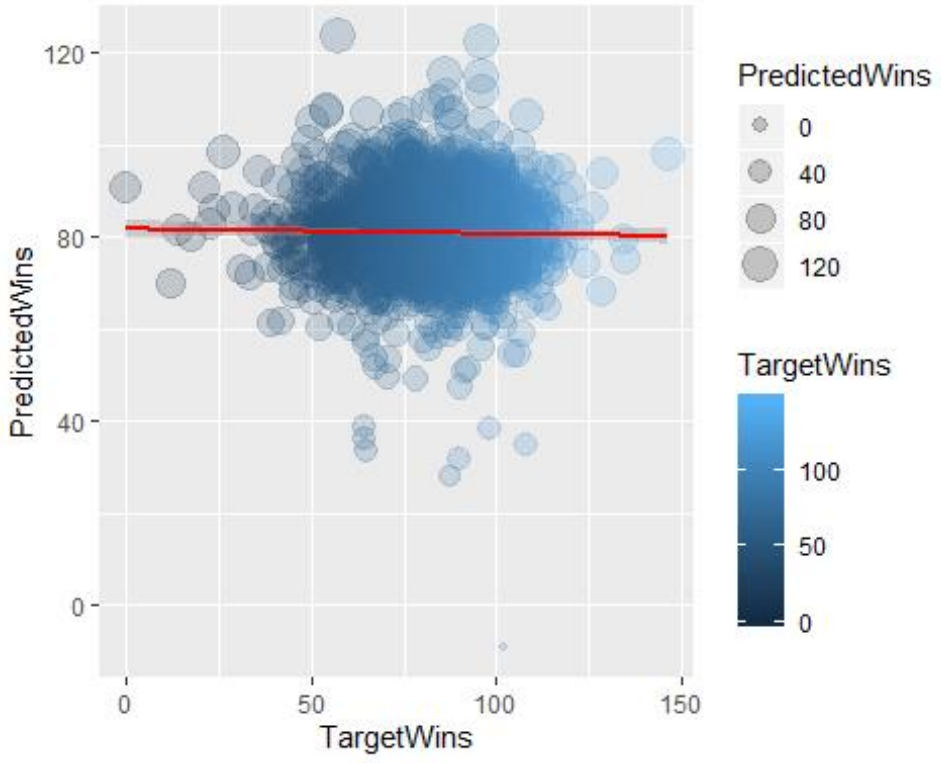
In the below models results show comparison of data wih remove NA values. The GGPlot below shows a scattered cluster with a dispersed linear line for NA Removals

```
##   Id PredictedWins TargetWins
## 1  9      69.83754          86
## 2 10      70.48421          76
## 3 14      78.22330          76
## 4 47      84.64512          92
## 5 60      71.16488         107
## 6 63      70.63571          82
```

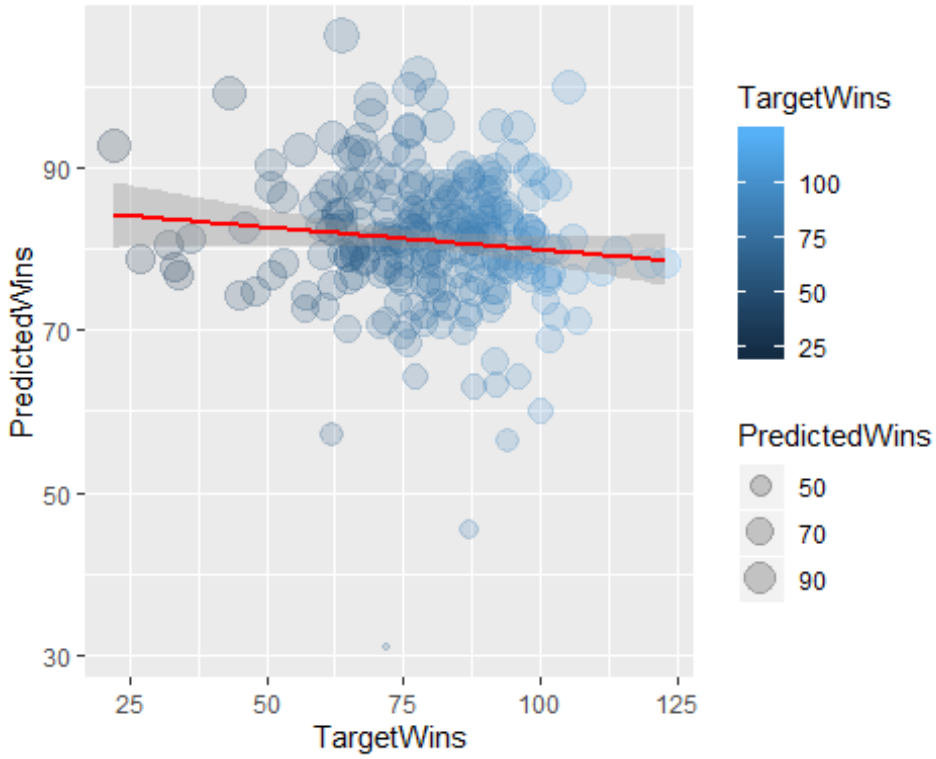
Conclusion

Both predicuation appear to be similar, hower the replacement of NA values appear to be a better approach. The ggplot conifrms tha values are more correlated.

REMOVE NA



REPLACE NA


PredictedWins
90
70
50
30
25
50
75
100
125
TargetWins
TargetWins
100
75
50
25
PredictedWins
50
70
90

Appendix:

R Code:

```
---
title: "Data 608 Homework 1"
author: "Anthony Pagan"
date: "September 15, 2019"
output:
  word_document:
    toc: yes
    toc_depth: '2'
  pdf_document:
    toc: yes
    toc_depth: '2'
  html_document:
    css: style.css
    toc: yes
    toc_depth: 2
    toc_float: yes
---

#Data Exploration

```{r echo=FALSE, message=FALSE, warning=FALSE}
#GEt the data

library(GGally)
library(dplyr)
```
```

mbe <- read.csv("C:\\Users\\apagan\\OneDrive - BizoIT, Inc\\Desktop\\GitHub\\CUNYSPS\\Data621\\HW1\\moneyball-evaluation-data.csv", header= TRUE)

mbt <- read.csv("C:\\Users\\apagan\\OneDrive - BizoIT, Inc\\Desktop\\GitHub\\CUNYSPS\\Data621\\HW1\\moneyball-training-data.csv", header= TRUE)

```

In this exercise we will go with 2 approaches. One approach would be to remove data with NA values and the second approach would be to replace the NA data with a value. We will attempt both approaches and use the one with the best predictions.

##Data Explore  - Replace NA Values

The initial review of the data shows 6 columns with incomplete data for 6 Columns

```{r echo=FALSE, message=FALSE, warning=FALSE}

nrow(mbt)

s<-summary(mbt)

s[7,]

```

THis is a summary of values for each column that has NA data values. For the most part the mean and median values are close enough to theorize that data of the six columns with NA value are fairly normal. We will attempt a replacement as one approachin in the data preparation section.

```{r echo=FALSE, message=FALSE, warning=FALSE}

summary(mbt$TEAM_BATTING_SO)

summary(mbt$TEAM_BASERUN_SB)

summary(mbt$TEAM_BASERUN_CS)

summary(mbt$TEAM_BATTING_HBP)

summary(mbt$TEAM_PITCHING_SO)

summary(mbt$TEAM_FIELDING_DP)

```

##Data Explore - Remove NA Values

If we remove all rows with incomplete rows, there will be a total of 191 rows. We need to decide if using only `r round(191/2085,2)`% of the data suffice

```{r echo=FALSE, message=FALSE, warning=FALSE}

summary(complete.cases(mbt))

```

A look at the mean and median values show a larger spread in TB_SO, TP_H and TP_E. The expectation is there will be more outliers in these groups

```{r echo=FALSE, message=FALSE, warning=FALSE}

s[3:4,]

```

The box plots below confirms the outliers as expected ,but TP_E is not as drastic as the others. However te
```

```{r echo=FALSE, message=FALSE, warning=FALSE}

par(mar=c(9.5,3.5,.5,.5))

boxplot(mbt, las=2)


plot(lm(mbt$TARGET_WINS~mbt$TEAM_FIELDING_E))



ggpairs(data=mbt, columns = c(2,14))
```

An initial view of the data show that TEAM_FIELDING_E and TEAM_FIELDING_DP have low P value and may have high correlation with team wins.

```{r echo=FALSE, message=FALSE, warning=FALSE}

fit<-lm(TARGET_WINS ~.-INDEX, mbt)

summary(fit)
```

The qq dot plots do not follow the residual line and fails the normality test

```{r echo=FALSE, message=FALSE, warning=FALSE}

qqplot(mbt$TARGET_WINS,residuals(lm(mbt$TARGET_WINS~mbt$TEAM_FIELDING_E)))

qqline(mbt$TARGET_WINS,residuals(lm(mbt$TARGET_WINS~mbt$TEAM_FIELDING_E)))

qqplot(mbt$TARGET_WINS,residuals(lm(mbt$TARGET_WINS~mbt$TEAM_FIELDING_DP))
)

qqline(mbt$TARGET_WINS,residuals(lm(mbt$TARGET_WINS~mbt$TEAM_FIELDING_DP))
)
```

```

#Data Preparation

As a start, we begin by redjusting the data column headings to shorter column names.

```{r echo=FALSE, message=FALSE, warning=FALSE}

str(mbt)

colnames(mbt)<-
c('Index','Wins','TB_Hits','TB_2B','TB_3B','TB_HR','TB_BB','TB_SO','TBR_SB','TBR_CS','TB_HBP','TP_H','TP_HR','TP_BB','TP_SO','TP_E','TP_DP')

```

Data Prepare - Replace NA values

In our analysis of the summbary of columns with NA values we noted that median and mean values were close enough to theorize that these column values were fairly normal. As a result, we replace any NAs with the mean value of the column data.

```{r echo=FALSE, message=FALSE, warning=FALSE}

mbta <-mbt

mbta$TB_SO[is.na(mbta$TB_SO)] <- mean(mbta$TB_SO,na.rm=TRUE)

mbta$TBR_SB[is.na(mbta$TBR_SB)] <- mean(mbta$TBR_SB,na.rm=TRUE)

mbta$TBR_CS[is.na(mbta$TBR_CS)] <- mean(mbta$TBR_CS,na.rm=TRUE)

mbta$TB_HBP[is.na(mbta$TB_HBP)] <- mean(mbta$TB_HBP,na.rm=TRUE)

mbta$TP_SO[is.na(mbta$TP_SO)] <- mean(mbta$TP_SO,na.rm=TRUE)

mbta$TP_DP[is.na(mbta$TP_DP)] <- mean(mbta$TP_DP,na.rm=TRUE)
```
```

s<-summary(mbta)

s

```

## Data Prepart - Remove NA values

In this next approach we are removing columns with missing data. We run a linear model with reduced columns and look at the corrrelation charts.

```{r echo=FALSE, message=FALSE, warning=FALSE}

library(dplyr)

mbt2<-mbt%>%

  select(Wins, TB_Hits, TB_2B, TB_3B, TB_HR, TB_BB, TP_H, TP_HR, TP_BB, TP_E)

str(mbt2)

```

Now we will run the linear model again with only the columns with complete data.

```{r echo=FALSE, message=FALSE, warning=FALSE}

fit<-lm(Wins ~., mbt2)

summary(fit)

```

The linear model shows the P values of TB_BB and TB_HR are greater than .05 so we can remove 2 columns and rerun the model.
```

```{r echo=FALSE, message=FALSE, warning=FALSE}

mbt3 <- mbt2%>%

   select(Wins, TB_Hits,TB_2B,TB_3B,TP_H,TP_HR,TP_BB,TP_E)


fit<-lm(Wins ~., mbt3)

summary(fit)


```

Last we use ggally to get an idea of correlation of the data and run a corr test to get raw correlation statistics. TB_Hits and TB_2B show the highest correlation and graphs show a positive correlation.

```{r echo=FALSE, message=FALSE, warning=FALSE}

ggpairs(data=mbt3)


cor(mbt2)[1,]


```

#Build Models

For all of the linear models we are extracting the coeficients, r squared values, adjusted r squared, sigma and f statistics. Coeficients provide y intercept, slope ,the t value which gives the standard deviations the estimated coefficients are from zero and p value which gives probability the null hypothesis is true. The multiple r-squred and adjusted r squared lets us know how close our data are to the linear regression model. The F-statistic gives us the relationship between dependent and independent variables. A large F-statistics means a strong relationship.

```{r echo=FALSE, message=FALSE, warning=FALSE}

fit<-lm(Wins ~.-Index, mbta)

s<-summary(fit)

```

Our first model uses the data set columns with complete data. The P value is very low. The Rsqured and Adjusted RSqaured values are below .5 and F Statistic is low

* Coefficients: `r s$coefficients[1,1:4]`

* RSquared: `r s$r.squared`

* Adjusted RSquared: `r s$adj.r.squared`

* Sigma: `r s$sigma`

* FStatistic: `r s$fstatistic`

```{r echo=FALSE, message=FALSE, warning=FALSE}

fit<-lm(Wins ~.-Index-TBR_CS-TB_HBP-TP_BB-TP_HR, mbta)

s<-summary(fit)

```

The next model uses the data set columns with columns with high pvalues re data. The P value is als very low. The Rsqured and Adjusted RSqaured values are below .5 and F Statistic is higher

* Coefficients: `r s$coefficients[1,1:4]`

* RSquared: `r s$r.squared`

* Adjusted RSquared: `r s$adj.r.squared`

* Sigma: `r s$sigma`

* FStatistic: `r s$fstatistic`

##Data Prepare - Remove NA Values

```{r echo=FALSE, message=FALSE, warning=FALSE}

# Answer Question 1 here

s<-summary(lm(Wins~., mbt2,na.action = na.fail))
```

This next approach removes NA columns. Our first model uses the data set columns wiht complete data. The P value is slightly above .05. The Rsqured and Adjusted RSqaured values are below .5 and F Statistic is low

* Coefficients: `r s$coefficients[1,1:4]`

* RSquared: `r s$r.squared`

* Adjusted RSquared: `r s$adj.r.squared`

* Sigma: `r s$sigma`

* FStatistic: `r s$fstatistic`

```{r echo=FALSE, message=FALSE, warning=FALSE}

l<-lm(Wins~.-TP_BB-TP_HR, mbt3,na.action = na.fail)

s<-summary(l)
```

```

Our next model removes TB_BB and TB_HR from previou dataset. The P value is lower at .01. The Rsqured and Adjusted RSqaured values are below .5 and F Statistic is higher than previous dataset. This tells us the new dataset has a lower probability of null hypothesis being true.

* Coefficients: `r s$coefficients[1,1:4]`

* RSquared: `r s$r.squared`

* Adjusted RSquared: `r s$adj.r.squared`

* Sigma: `r s$sigma`

* FStatistic: `r s$fstatistic`

##Select Models - Replace NA Values

In the below models results show comparison of data wih replace NA values. The GGplot below shows a tight cluster with a straight linear line for the NA replacement data.

```{r echo=FALSE, message=FALSE, warning=FALSE}

# Answer Question 2 here

mbta1 <- mbta%>%
  select(Index,TB_Hits,TB_2B,TB_3B,TP_H,TP_HR,TP_BB,TP_E)

p<-data.frame(mbta$Index,predict(fit, new=mbta),mbta$Wins[mbta$Index])

colnames(p)<-c('Id','PredictedWins','TargetWins')

head(p)
```
```

```

##Select Models - Remove NA Values

In the below models results show comparison of data wih remove NA values. The GGPlot below shows a scattered cluster with a dispersed linear line for NA Removals

```{r echo=FALSE, message=FALSE, warning=FALSE}
colnames(mbe)<-
c('Index','TB_Hits','TB_2B','TB_3B','TB_HR','TB_BB','TB_SO','TBR_SB','TBR_CS','TB_HBP','TP_H','TP_HR','TP_BB','TP_SO','TP_E','TP_DP')

mbe1 <- mbe%>%
  select(Index,TB_Hits,TB_2B,TB_3B,TP_H,TP_HR,TP_BB,TP_E)

p1<-data.frame(mbe$Index,predict(l, new=mbe),mbt$Wins[mbe$Index])

colnames(p1)<-c('Id','PredictedWins','TargetWins')

head(p1)
```

##Conclusion

Both predicuation appear to be similar, hower the replacement of NA values appear to be a better approach. The ggplot conifrms tha values are more correlated.

###REMOVE NA

```{r echo=FALSE, message=FALSE, warning=FALSE}
```
```

```
ggplot(p, aes(TargetWins, PredictedWins,width = 800, height = 300)) +
  geom_point(aes(group=TargetWins,size = PredictedWins, color = TargetWins), alpha = 0.2)+
  stat_smooth(method = "lm", col = "red")
```

###REPLACE NA

```{r echo=FALSE, message=FALSE, warning=FALSE}

ggplot(p1, aes(TargetWins, PredictedWins,width = 800, height = 300)) +
  geom_point(aes(group=TargetWins,size = PredictedWins, color = TargetWins), alpha = 0.2)+
  stat_smooth(method = "lm", col = "red")
```