Data 621 - HW 1

Group 4 Layla Quinones, Ian Castello, Dmitriy Burtsev & Esteban Aramayo

Sept. 26, 2021

```
#libraries
library(kableExtra)
library(tidyverse)
library(tidymodels)
library(VIM)
library(naniar)
library(GGally)
library(caret)
library(psych)
```

Data Exploration

```
#import the data
urlTraining = "https://raw.githubusercontent.com/MsQCompSci/Data621Group4/main/HW1/moneyball-training-da

#get the data
rawData <- read.csv(urlTraining)

#Display what we imported
str(rawData)
```

```
## '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 ...
```

```
#From this we can see that there are 2276 observations and 17 variables in total which means we have 1
```

```
#display summary statistics
summary(rawData)
```

```
##      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
```

```
#From this we can gain some more insight into each variable specifically. We can see that most variable
```

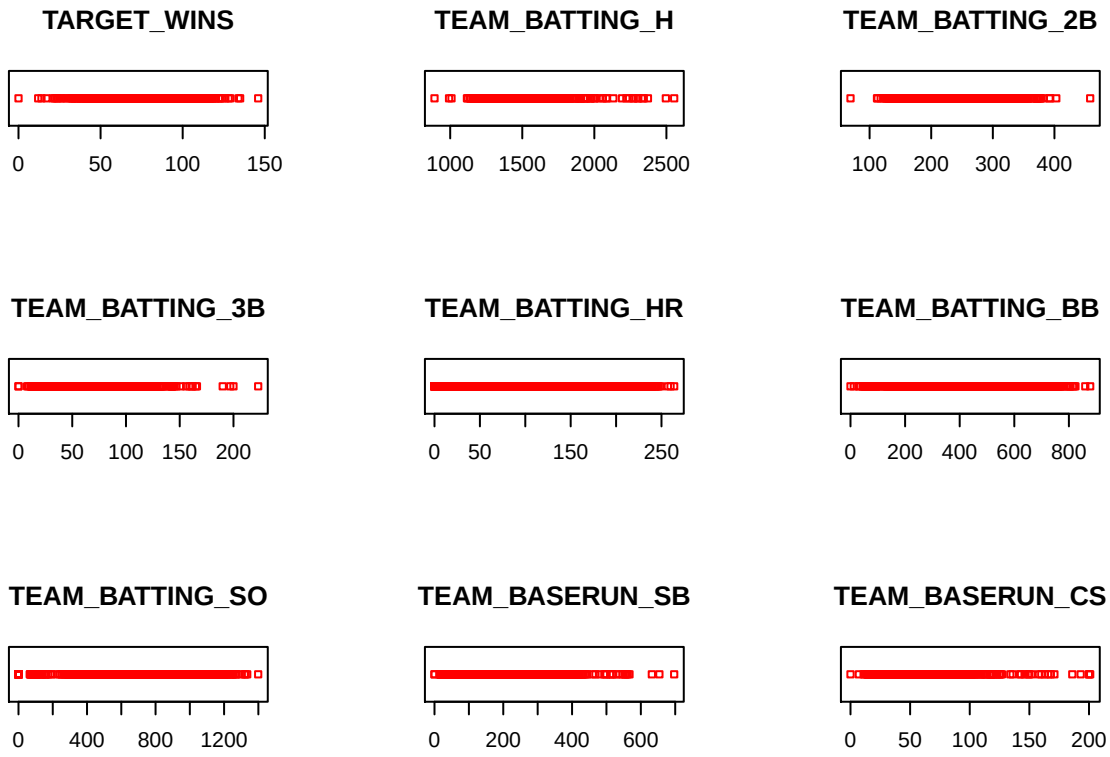
```
#Using describe we can get even more insight into the shape of each variable
describe(rawData)
```

```
##                  vars    n    mean      sd median trimmed    mad  min   max
```

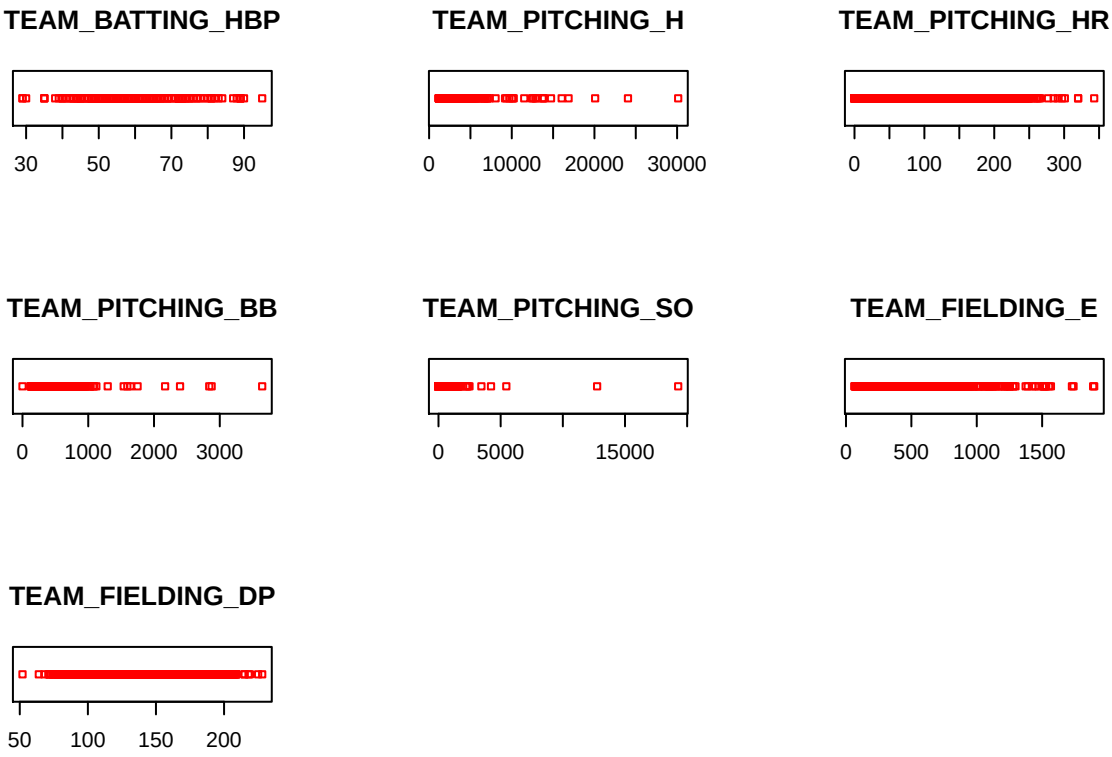
```
## INDEX              1 2276 1268.46  736.35 1270.5 1268.57 952.57    1  2535
## TARGET_WINS        2 2276   80.79   15.75   82.0   81.31  14.83    0   146
## TEAM_BATTING_H     3 2276 1469.27  144.59 1454.0 1459.04 114.16  891  2554
## TEAM_BATTING_2B    4 2276  241.25   46.80  238.0  240.40  47.44   69   458
## TEAM_BATTING_3B    5 2276   55.25   27.94   47.0   52.18  23.72    0   223
## TEAM_BATTING_HR    6 2276   99.61   60.55  102.0   97.39  78.58    0   264
## TEAM_BATTING_BB    7 2276  501.56  122.67  512.0  512.18  94.89    0   878
## TEAM_BATTING_SO    8 2174  735.61  248.53  750.0  742.31 284.66    0  1399
## TEAM_BASERUN_SB    9 2145  124.76   87.79  101.0  110.81  60.79    0   697
## TEAM_BASERUN_CS   10 1504   52.80   22.96   49.0   50.36  17.79    0   201
## TEAM_BATTING_HBP  11  191   59.36   12.97   58.0   58.86  11.86   29    95
## TEAM_PITCHING_H   12 2276 1779.21 1406.84 1518.0 1555.90 174.95 1137 30132
## TEAM_PITCHING_HR  13 2276  105.70   61.30  107.0  103.16  74.13    0   343
## TEAM_PITCHING_BB  14 2276  553.01  166.36  536.5  542.62  98.59    0  3645
## TEAM_PITCHING_SO  15 2174  817.73  553.09  813.5  796.93 257.23    0 19278
## TEAM_FIELDING_E   16 2276  246.48  227.77  159.0  193.44  62.27   65  1898
## TEAM_FIELDING_DP  17 1990  146.39   26.23  149.0  147.58  23.72   52   228
##                   range  skew kurtosis    se
## INDEX              2534  0.00    -1.22 15.43
## TARGET_WINS         146 -0.40     1.03  0.33
## TEAM_BATTING_H     1663  1.57     7.28  3.03
## TEAM_BATTING_2B     389  0.22     0.01  0.98
## TEAM_BATTING_3B     223  1.11     1.50  0.59
## TEAM_BATTING_HR     264  0.19    -0.96  1.27
## TEAM_BATTING_BB     878 -1.03     2.18  2.57
## TEAM_BATTING_SO    1399 -0.30    -0.32  5.33
## TEAM_BASERUN_SB     697  1.97     5.49  1.90
## TEAM_BASERUN_CS     201  1.98     7.62  0.59
## TEAM_BATTING_HBP     66  0.32    -0.11  0.94
## TEAM_PITCHING_H   28995 10.33   141.84 29.49
## TEAM_PITCHING_HR    343  0.29    -0.60  1.28
## TEAM_PITCHING_BB   3645  6.74    96.97  3.49
## TEAM_PITCHING_SO  19278 22.17   671.19 11.86
## TEAM_FIELDING_E    1833  2.99    10.97  4.77
## TEAM_FIELDING_DP    176 -0.39     0.18  0.59
```

```
#pitching looks highly skewed
```

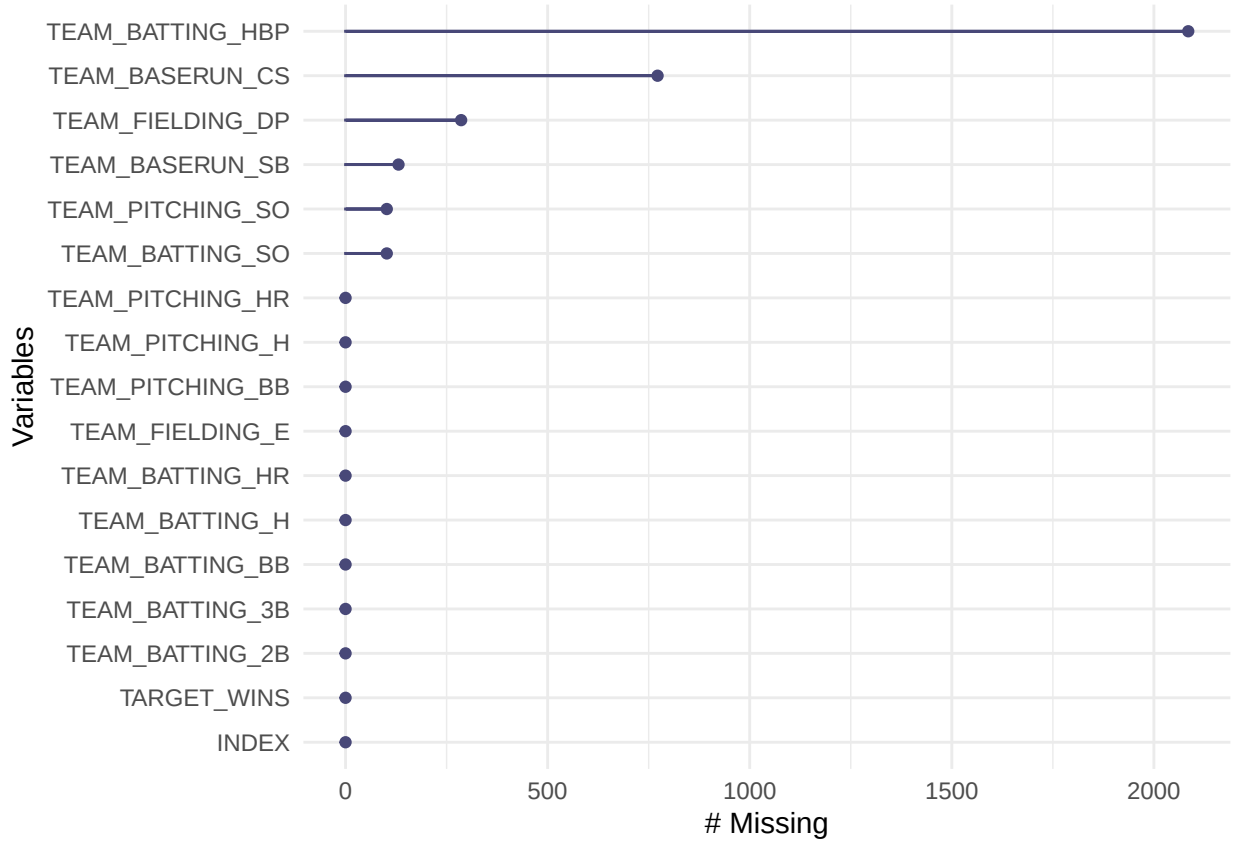
```
#distribution of each variable
par(mfrow = c(3,3))
for(i in 2:ncol(rawData)) {
  plot(rawData[i], main = colnames(rawData[i]), col = "red")
}
```




```
#shows that pitching variables are skewed and can affect training the model
```



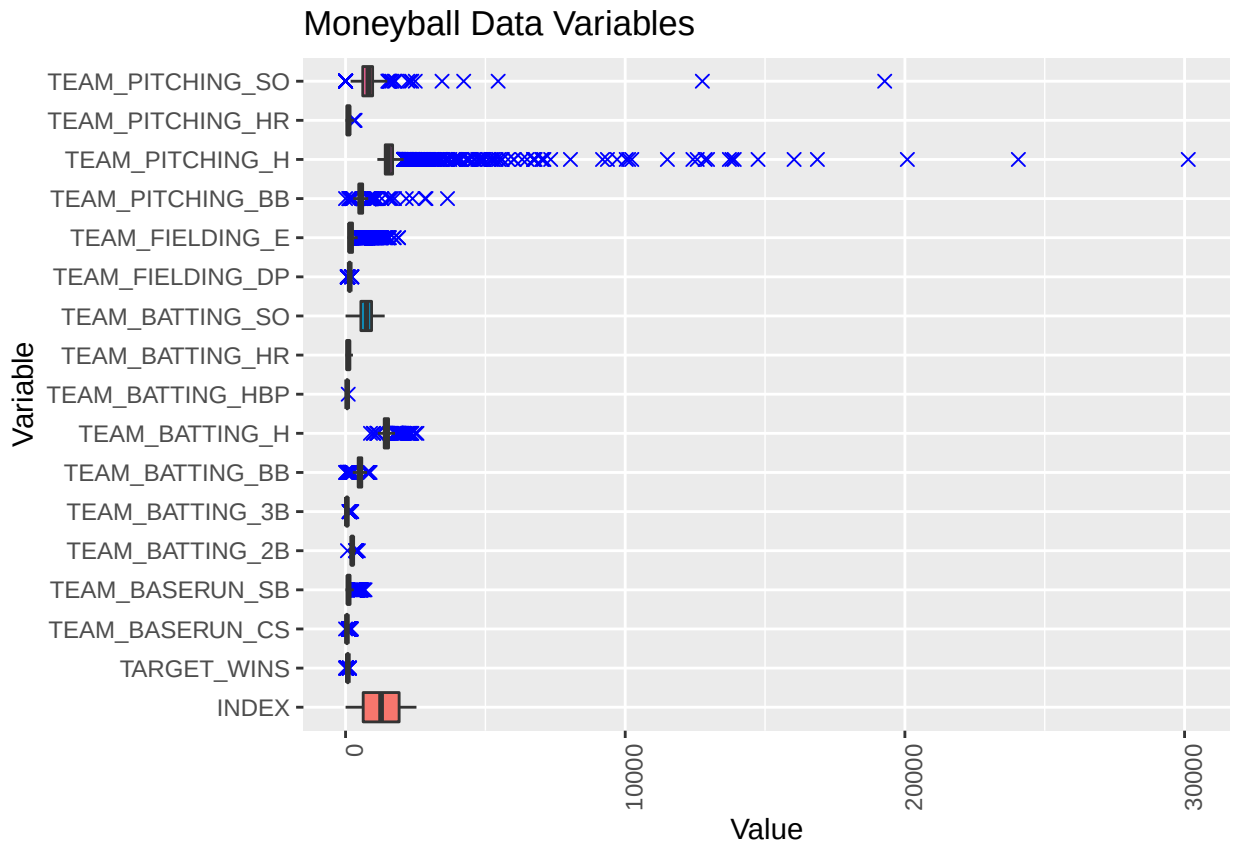

```
#Lets take a look at missing values (using naniar)
gg_miss_var(rawData)
```




```
#This graph gives us a visual of how many missing values there are n each variable which will determine
```

```
#make long
longData <- rawData %>%
  gather(key = Variable, value = Value)

#Initial boxplot with outliers and NAs omitted
ggplot(longData, aes(Variable, Value, fill = Variable)) +
  geom_boxplot(outlier.colour="blue",
               outlier.shape=4,
               outlier.size=2,
               show.legend=FALSE) +
   theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
   coord_flip()+
  labs(title="Moneyball Data Variables", y="Value")
```

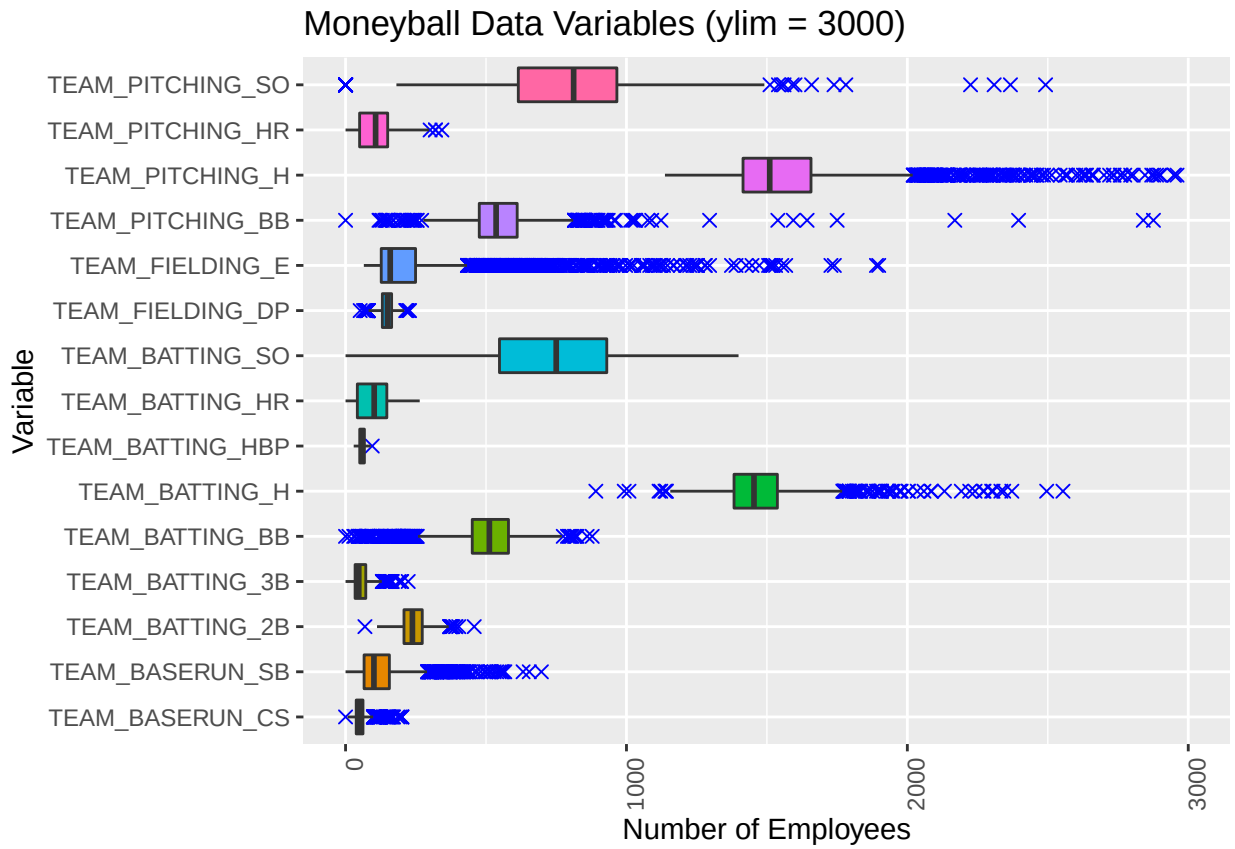
```
## Warning: Removed 3478 rows containing non-finite values (stat_boxplot).
```




```
#This visualization gives us an idea of all the outliers we have in each variable but does not give us
```

```
#Boxplot without outliers (ignoring all points greeater than 1500)
#filtering out target and index
rawData2 <- longData %>%
  filter(Variable != "INDEX", Variable != "TARGET_WINS")


ggplot(rawData2, aes(Variable, Value, fill = Variable)) +
  geom_boxplot(outlier.colour="blue",
               outlier.shape=4,
               outlier.size=2,
               show.legend=FALSE) +
  ylim(0,3000) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  coord_flip()+
  labs(title="Moneyball Data Variables (ylim = 3000)", y="Number of Employees")
```

```
## Warning: Removed 3570 rows containing non-finite values (stat_boxplot).
```

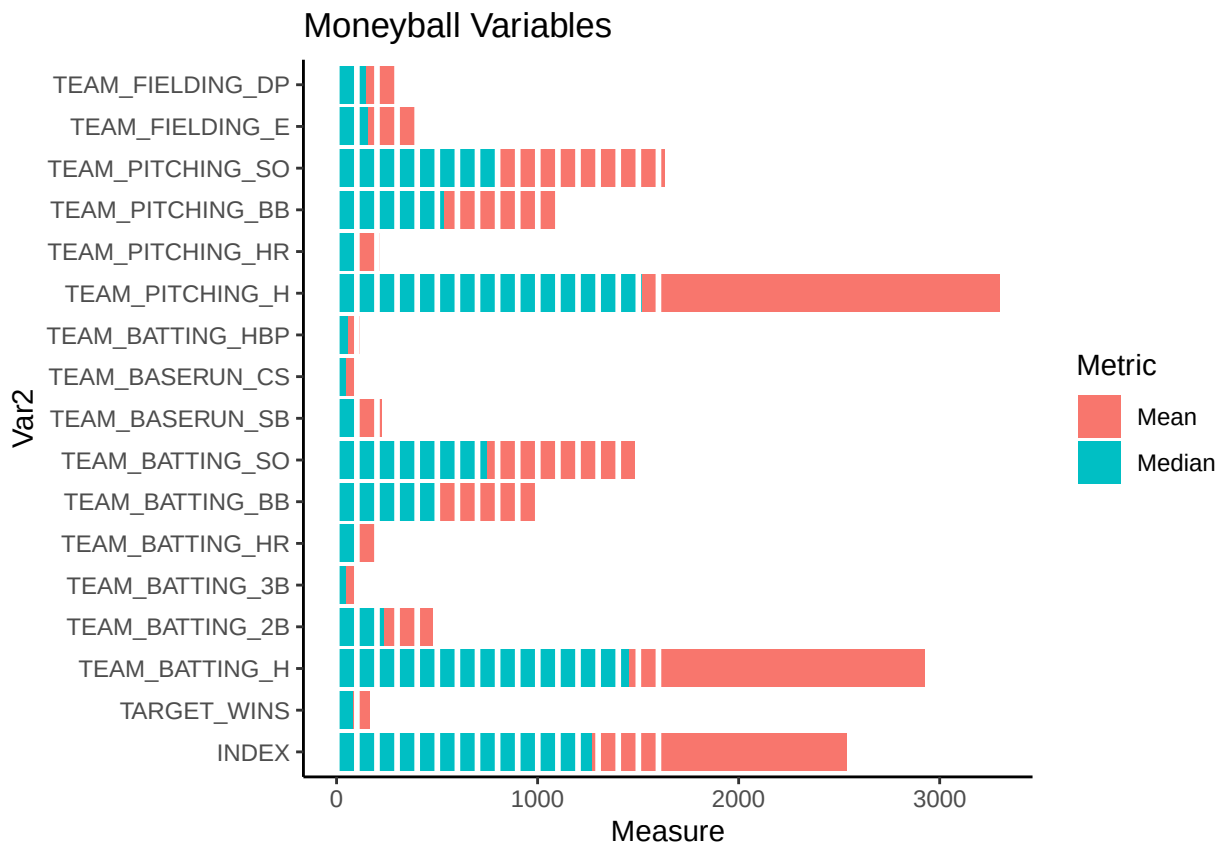



```
#removed rows are NA values - This visualization gives a better sense of how many outliers there are in
```

```
#Lets compare means and medians to get a better sense of skew
p <-summary(rawData) %>% #gather sumary stats
  as.data.frame(.) %>% #trurn into DF
  filter(grepl('Mean|Median', Freq))%>% #grab means and medians
  separate(Freq, c('Measure', 'Value'))%>% # seperate column into numeric and measure
  transform(Value = as.numeric(Value))%>% #make sure the value column is numeric
  select(Var2:Value)
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 34 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
#plot overlapping
ggplot(p, aes(x=Var2, y=Value))+
  geom_bar(stat = "identity", aes(fill=Measure))+
  geom_hline(yintercept=seq(1,1700,100), col="white", lwd=1)+
  theme_classic()+
  coord_flip()+
  scale_fill_discrete(name = "Metric", labels = c("Mean", "Median")) +
  labs(title="Moneyball Variables", y="Measure")
```




```
#This graph gives an even clearer sense of possible skew in data. We see again TEAM_PITCHING_H has a hu
```

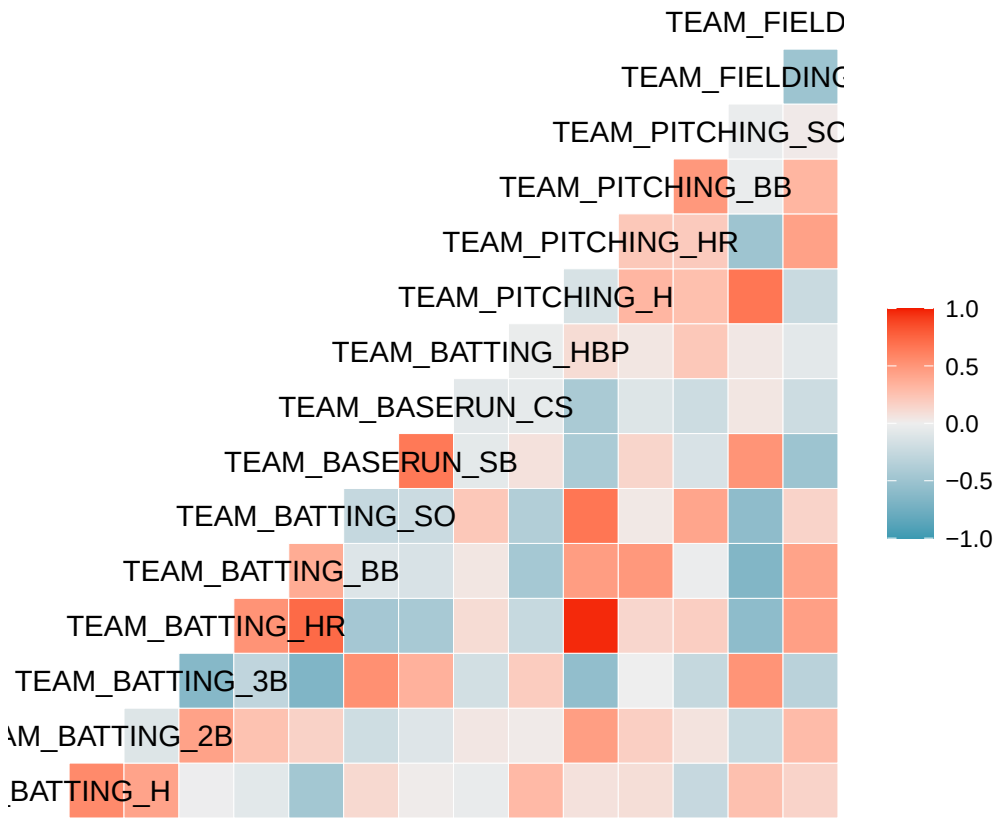
```
#correlation between variables
rawTrainX <- rawData %>%
  select(TEAM_BATTING_H: TEAM_FIELDING_DP)

rawTrainY <- rawData$TARGET_WINS

#correlation matrix (use only complete observations)
cor(rawTrainX, use = "complete.obs")
```

```
##                  TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## TEAM_BATTING_H       1.00000000      0.56177286      0.21391883      0.39627593
## TEAM_BATTING_2B      0.56177286      1.00000000      0.04203441      0.25099045
## TEAM_BATTING_3B      0.21391883      0.04203441      1.00000000     -0.21879927
## TEAM_BATTING_HR      0.39627593      0.25099045     -0.21879927      1.00000000
## TEAM_BATTING_BB      0.19735234      0.19749256     -0.20584392      0.45638161
## TEAM_BATTING_SO     -0.34174328     -0.06415123     -0.19291841      0.21045444
## TEAM_BASERUN_SB      0.07167495     -0.18768279      0.16946086     -0.19021893
## TEAM_BASERUN_CS     -0.09377545     -0.20413884      0.23213978     -0.27579838
## TEAM_BATTING_HBP    -0.02911218      0.04608475     -0.17424715      0.10618116
## TEAM_PITCHING_H      0.99919269      0.56045355      0.21250322      0.39549390
## TEAM_PITCHING_HR     0.39495630      0.24999875     -0.21973263      0.99993259
## TEAM_PITCHING_BB     0.19529071      0.19592157     -0.20675383      0.45542468
## TEAM_PITCHING_SO    -0.34445001     -0.06616615     -0.19386654      0.20829574
```

```
## TEAM_FIELDING_E     -0.25381638     -0.19427027     -0.06513145      0.01567397
## TEAM_FIELDING_DP     0.01776946     -0.02488808      0.13314758     -0.06182222
##                  TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB
## TEAM_BATTING_H        0.19735234     -0.34174328      0.07167495
## TEAM_BATTING_2B       0.19749256     -0.06415123     -0.18768279
## TEAM_BATTING_3B      -0.20584392     -0.19291841      0.16946086
## TEAM_BATTING_HR       0.45638161      0.21045444     -0.19021893
## TEAM_BATTING_BB       1.00000000      0.21833871     -0.08806123
## TEAM_BATTING_SO       0.21833871      1.00000000     -0.07475974
## TEAM_BASERUN_SB      -0.08806123     -0.07475974      1.00000000
## TEAM_BASERUN_CS      -0.20878051     -0.05613035      0.62473781
## TEAM_BATTING_HBP      0.04746007      0.22094219     -0.06400498
## TEAM_PITCHING_H       0.19848687     -0.34145321      0.07395373
## TEAM_PITCHING_HR      0.45659283      0.21111617     -0.18948057
## TEAM_PITCHING_BB      0.99988140      0.21895783     -0.08741902
## TEAM_PITCHING_SO      0.21793253      0.99976835     -0.07351325
## TEAM_FIELDING_E      -0.07847126      0.30814540      0.04292341
## TEAM_FIELDING_DP     -0.07929078     -0.12319072     -0.13023054
##                  TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H
## TEAM_BATTING_H      -0.093775445      -0.02911218      0.99919269
## TEAM_BATTING_2B     -0.204138837       0.04608475      0.56045355
## TEAM_BATTING_3B      0.232139777      -0.17424715      0.21250322
## TEAM_BATTING_HR     -0.275798375       0.10618116      0.39549390
## TEAM_BATTING_BB     -0.208780510       0.04746007      0.19848687
## TEAM_BATTING_SO     -0.056130355       0.22094219     -0.34145321
## TEAM_BASERUN_SB      0.624737808      -0.06400498      0.07395373
## TEAM_BASERUN_CS      1.000000000      -0.07051390     -0.09297789
## TEAM_BATTING_HBP    -0.070513896       1.00000000     -0.02769699
## TEAM_PITCHING_H     -0.092977893      -0.02769699      1.00000000
## TEAM_PITCHING_HR    -0.275471495       0.10675878      0.39463199
## TEAM_PITCHING_BB    -0.208470154       0.04785137      0.19703302
## TEAM_PITCHING_SO    -0.055308336       0.22157375     -0.34330646
## TEAM_FIELDING_E      0.207701189       0.04178971     -0.25073028
## TEAM_FIELDING_DP    -0.006764233      -0.07120824      0.01416807
##                  TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
## TEAM_BATTING_H         0.39495630       0.19529071      -0.34445001
## TEAM_BATTING_2B        0.24999875       0.19592157      -0.06616615
## TEAM_BATTING_3B       -0.21973263      -0.20675383      -0.19386654
## TEAM_BATTING_HR        0.99993259       0.45542468       0.20829574
## TEAM_BATTING_BB        0.45659283       0.99988140       0.21793253
## TEAM_BATTING_SO        0.21111617       0.21895783       0.99976835
## TEAM_BASERUN_SB       -0.18948057      -0.08741902      -0.07351325
## TEAM_BASERUN_CS       -0.27547150      -0.20847015      -0.05530834
## TEAM_BATTING_HBP       0.10675878       0.04785137       0.22157375
## TEAM_PITCHING_H        0.39463199       0.19703302      -0.34330646
## TEAM_PITCHING_HR       1.00000000       0.45580983       0.20920115
## TEAM_PITCHING_BB       0.45580983       1.00000000       0.21887700
## TEAM_PITCHING_SO       0.20920115       0.21887700       1.00000000
## TEAM_FIELDING_E        0.01689330      -0.07692315       0.31008407
## TEAM_FIELDING_DP      -0.06292475      -0.08040645      -0.12492321
##                  TEAM_FIELDING_E TEAM_FIELDING_DP
## TEAM_BATTING_H       -0.25381638      0.017769456
## TEAM_BATTING_2B      -0.19427027     -0.024888081
## TEAM_BATTING_3B      -0.06513145      0.133147578
```

```
## TEAM_BATTING_HR       0.01567397     -0.061822219
## TEAM_BATTING_BB      -0.07847126     -0.079290775
## TEAM_BATTING_SO       0.30814540     -0.123190715
## TEAM_BASERUN_SB       0.04292341     -0.130230537
## TEAM_BASERUN_CS       0.20770119     -0.006764233
## TEAM_BATTING_HBP      0.04178971     -0.071208241
## TEAM_PITCHING_H      -0.25073028      0.014168073
## TEAM_PITCHING_HR      0.01689330     -0.062924751
## TEAM_PITCHING_BB     -0.07692315     -0.080406452
## TEAM_PITCHING_SO      0.31008407     -0.124923213
## TEAM_FIELDING_E       1.00000000      0.040205814
## TEAM_FIELDING_DP      0.04020581      1.000000000
```

```
#correlation matrix visualization
ggcorr(rawTrainX)
```

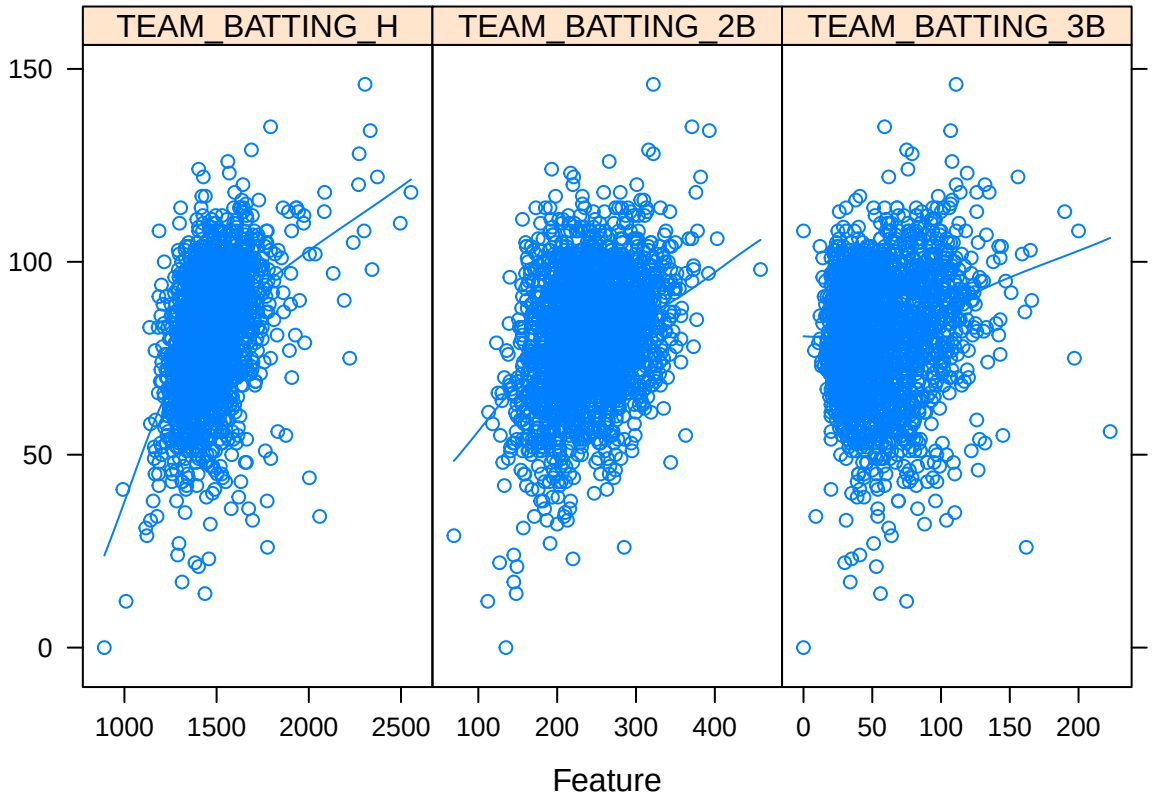



```
#check variance (there are none) which is good - we would throw this out if there was one iwth zero var
nearZeroVar(rawTrainX)
```

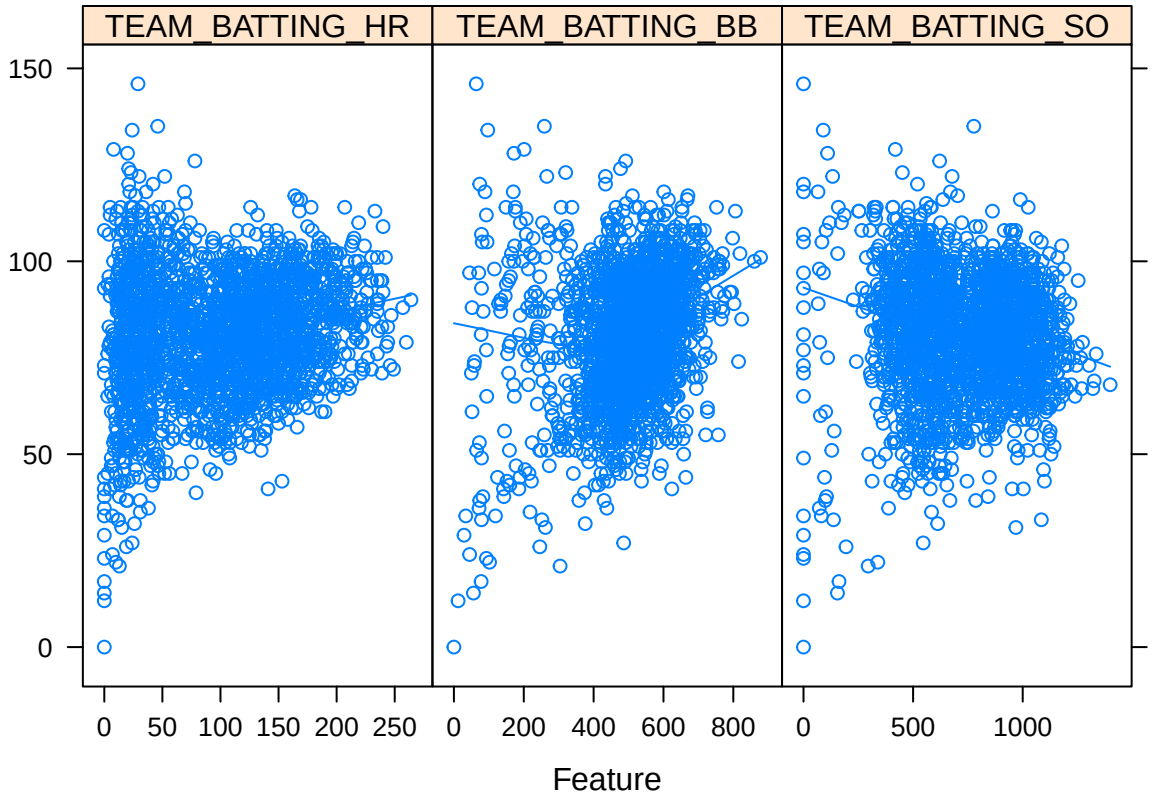
```
## integer(0)
```

```
#There are a number of correlated variables which may affect the model - we want to make sure that coli
```

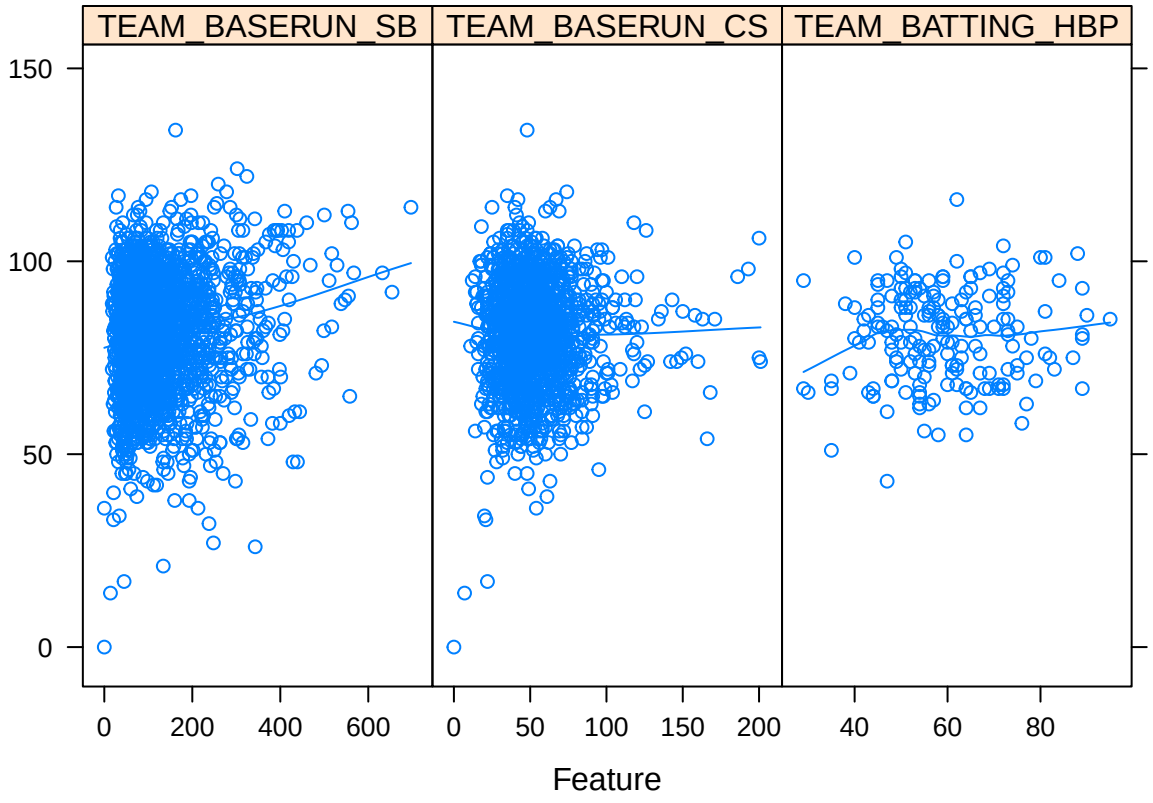
```r
#Feature Plot against target variable
  featurePlot(y = rawTrainY,
              x = rawTrainX[1:3],
              plot = "scatter",
            type = c("p", "smooth"),
            span = .5,
            layout = c(3, 1))
```



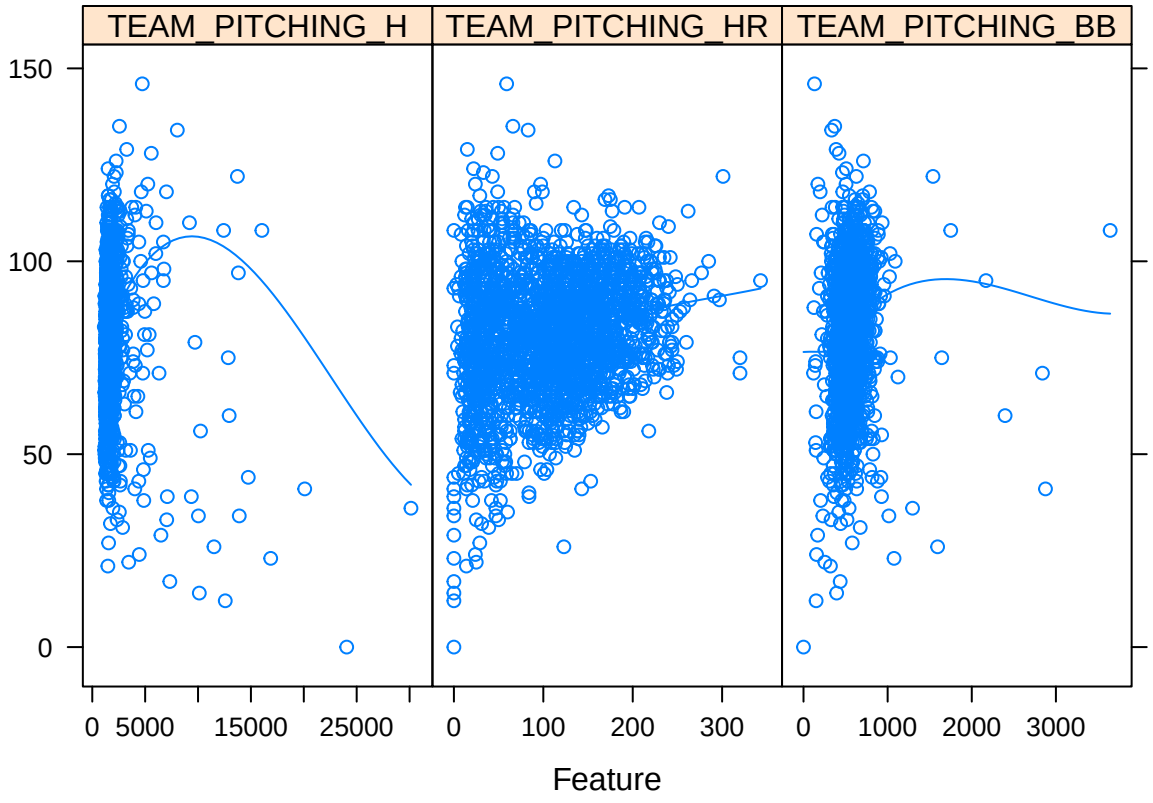

```r
  featurePlot(y = rawTrainY,
              x = rawTrainX[4:6],
              plot = "scatter",
            type = c("p", "smooth"),
            span = .5,
            layout = c(3, 1))
```



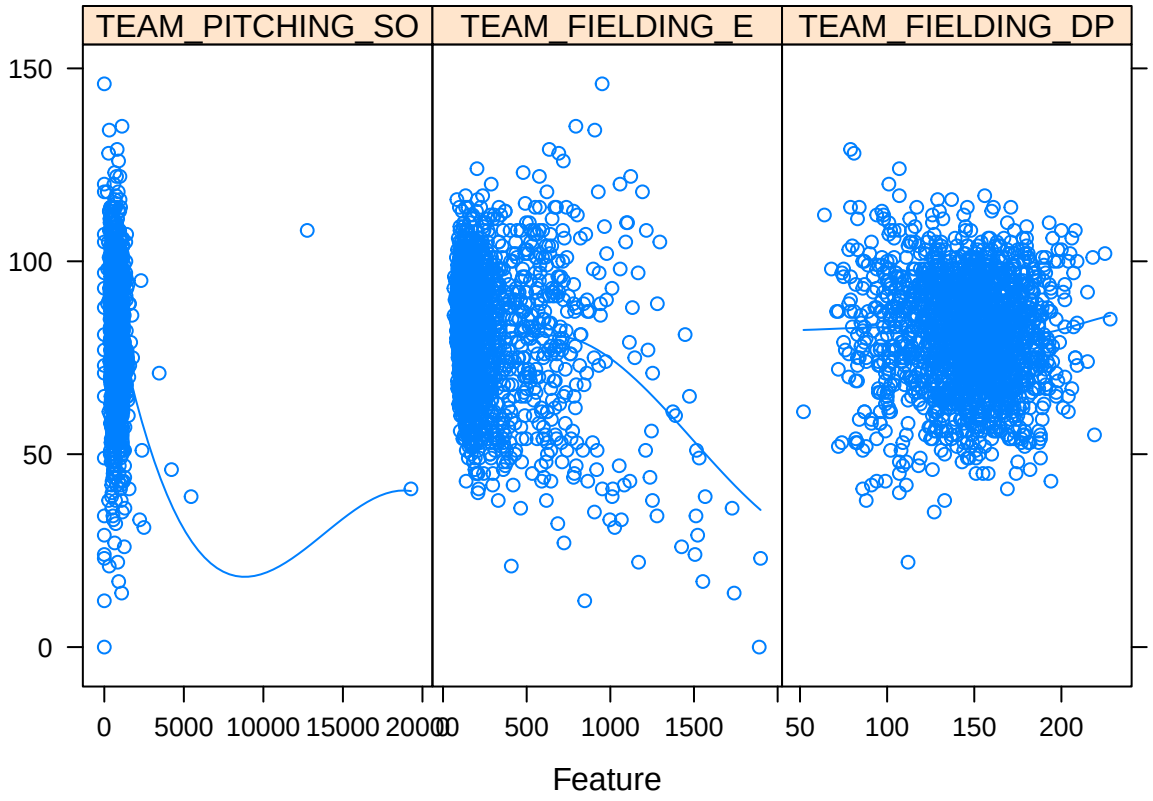

```
featurePlot(y = rawTrainY,
            x = rawTrainX[7:9],
            plot = "scatter",
          type = c("p", "smooth"),
          span = .5,
          layout = c(3, 1))
```




```
featurePlot(y = rawTrainY,
            x = rawTrainX[10:12],
            plot = "scatter",
          type = c("p", "smooth"),
          span = .5,
          layout = c(3, 1))
```




```
featurePlot(y = rawTrainY,
      x = rawTrainX[13:15],
      plot = "scatter",
    type = c("p", "smooth"),
    span = .5,
    layout = c(3, 1))
```




```
#from these plots we can see that we should be mindful of TEAM_PITCHING_ (not so much HR), TEAM_FIELDIN
```

Data Preperation

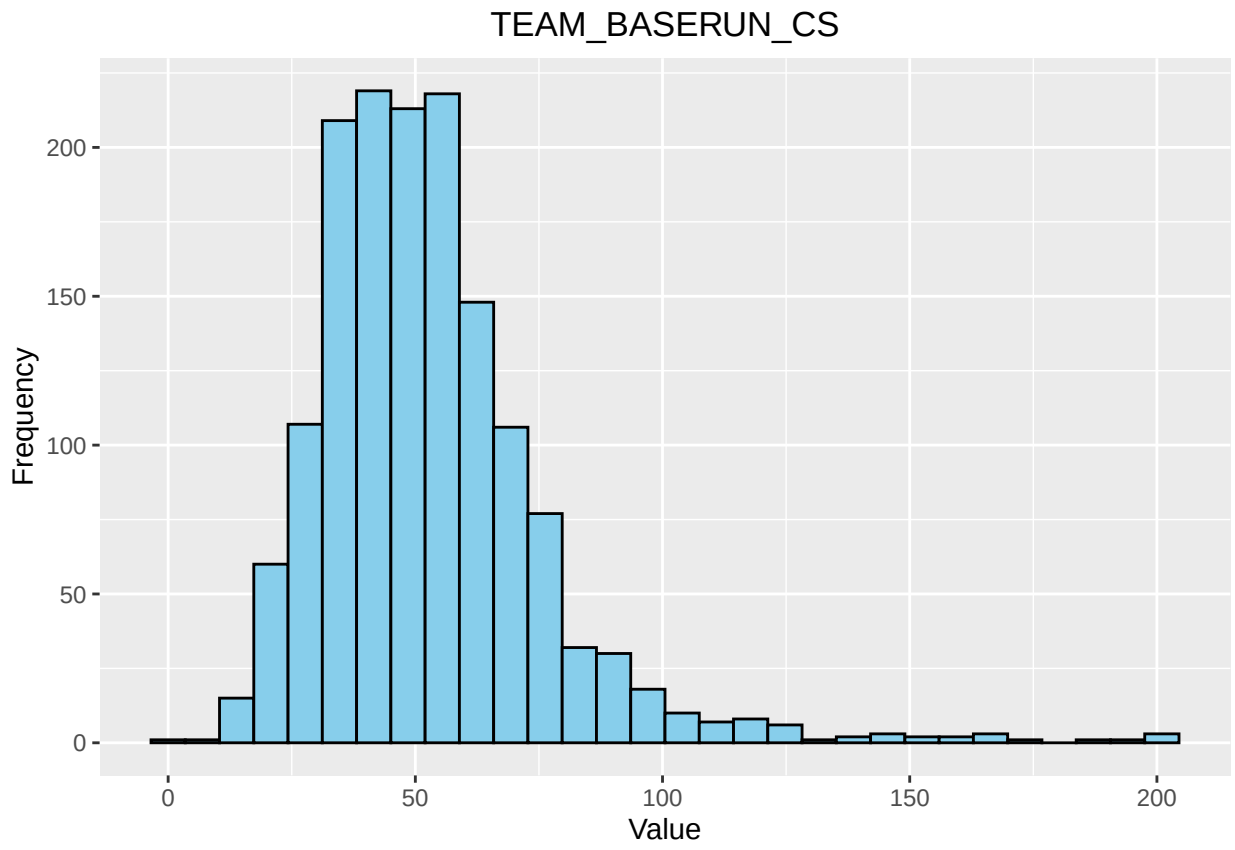
```
#Take a look at the 4 variables we identified with missing data
summary(rawTrainX%>% select(TEAM_BATTING_HBP, TEAM_BASERUN_CS, TEAM_BASERUN_SB,TEAM_FIELDING_DP, TEAM_P
```

```
##  TEAM_BATTING_HBP TEAM_BASERUN_CS TEAM_BASERUN_SB TEAM_FIELDING_DP
##  Min.   :29.00    Min.   :  0.0   Min.   :  0.0   Min.   : 52.0
##  1st Qu.:50.50    1st Qu.: 38.0   1st Qu.: 66.0   1st Qu.:131.0
##  Median :58.00    Median : 49.0   Median :101.0   Median :149.0
##  Mean   :59.36    Mean   : 52.8   Mean   :124.8   Mean   :146.4
##  3rd Qu.:67.00    3rd Qu.: 62.0   3rd Qu.:156.0   3rd Qu.:164.0
##  Max.   :95.00    Max.   :201.0   Max.   :697.0   Max.   :228.0
##  NA's   :2085     NA's   :772     NA's   :131     NA's   :286
##  TEAM_PITCHING_SO  TEAM_BATTING_SO
##  Min.   :    0.0   Min.   :   0.0
##  1st Qu.:  615.0   1st Qu.: 548.0
##  Median :  813.5   Median : 750.0
##  Mean   :  817.7   Mean   : 735.6
##  3rd Qu.:  968.0   3rd Qu.: 930.0
##  Max.   :19278.0   Max.   :1399.0
##  NA's   :102       NA's   :102
```

```
#Out of the 2276 observations, TEAM_BATTING_HBP has 2085 missing data points
#since thats most of the observations we will omit this column*****

ggplot(rawTrainX, aes(x=TEAM_BASERUN_CS)) +
  geom_histogram(color="black", fill="skyblue") +
  labs(title = "TEAM_BASERUN_CS", y = "Frequency", x = "Value")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

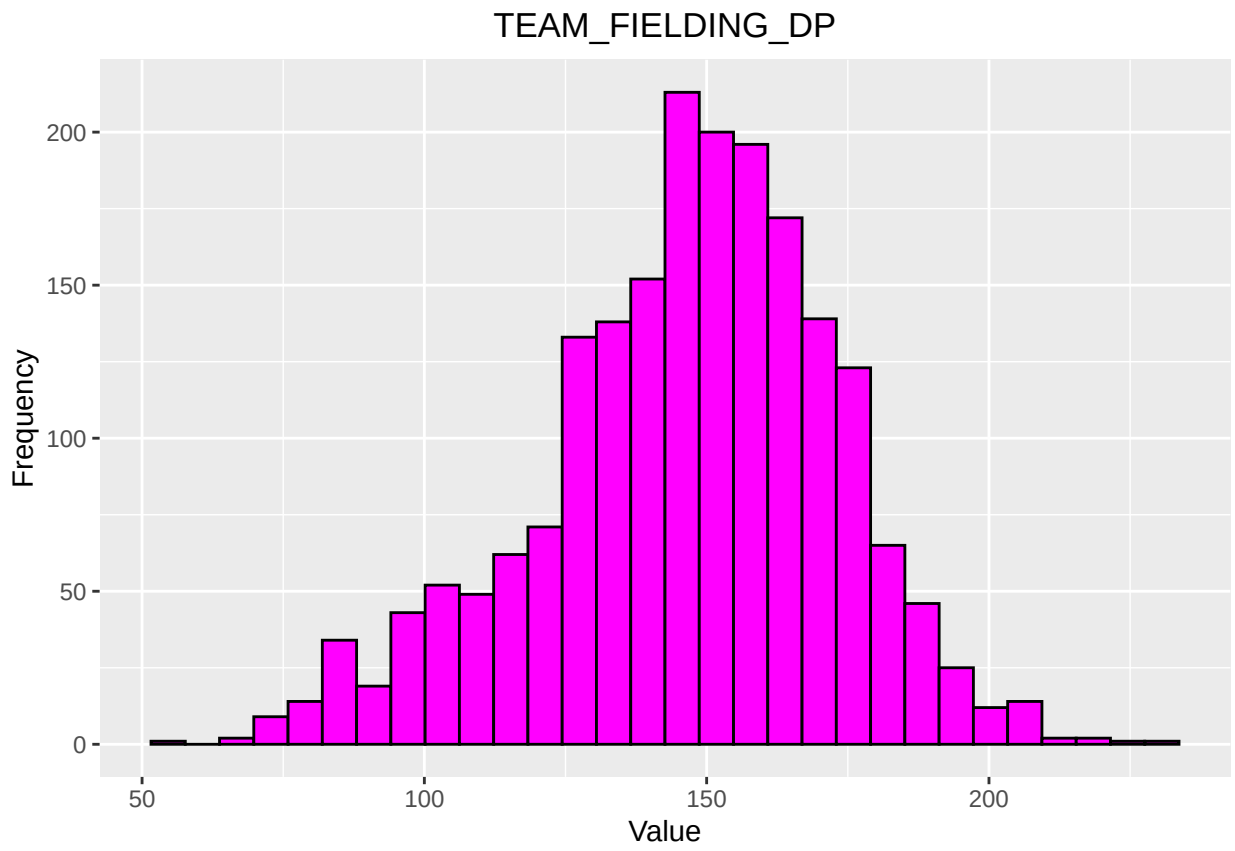
```
## Warning: Removed 772 rows containing non-finite values (stat_bin).
```




```
#since this variable has a slight skew, I will impute using the kNN (I can also do median)

ggplot(rawTrainX, aes(x=TEAM_FIELDING_DP)) +
  geom_histogram(color="black", fill="magenta") +
  labs(title = "TEAM_FIELDING_DP", y = "Frequency", x = "Value")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

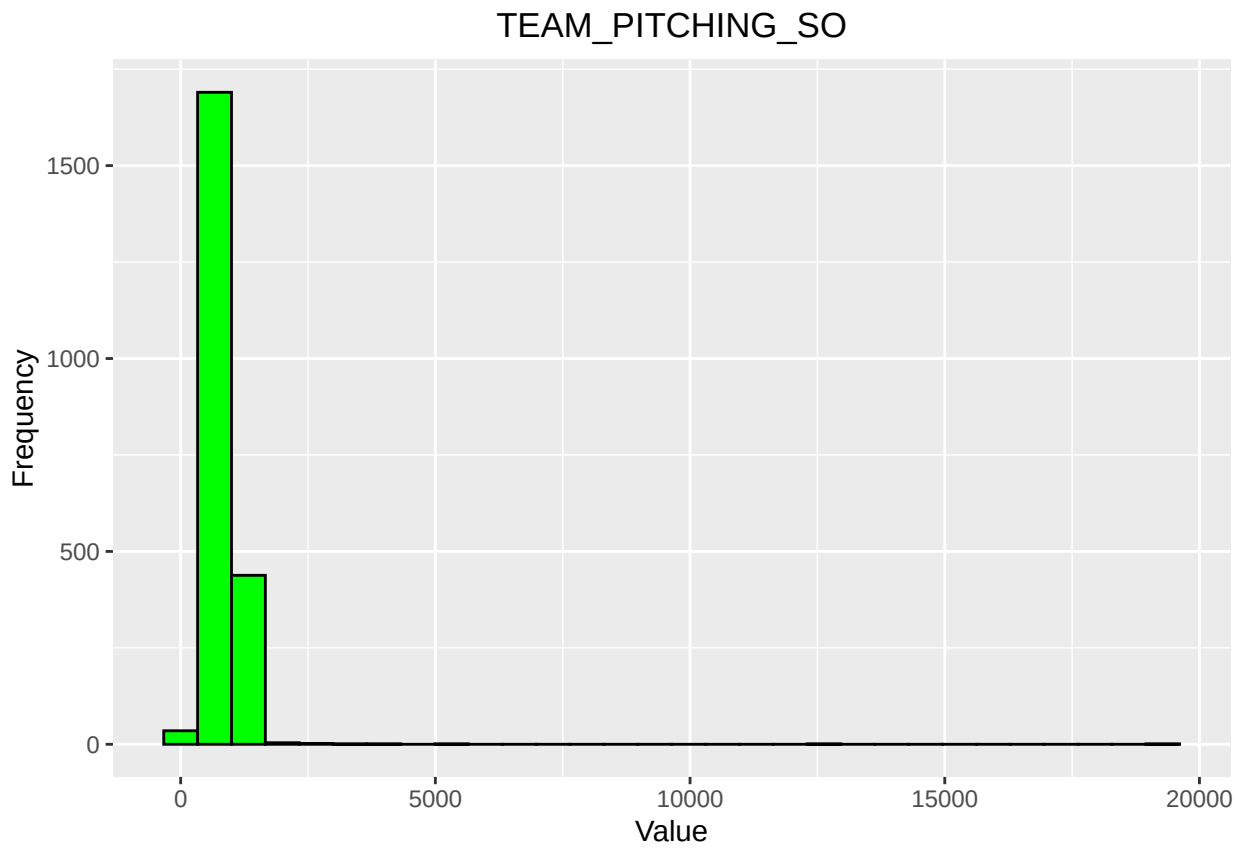
```
## Warning: Removed 286 rows containing non-finite values (stat_bin).
```




```
#since this variable is normally distributed, I will impute using the kNN (I can also do mean)

ggplot(rawTrainX, aes(x=TEAM_PITCHING_SO)) +
  geom_histogram(color="black", fill="green") +
  labs(title = "TEAM_PITCHING_SO", y = "Frequency", x = "Value")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

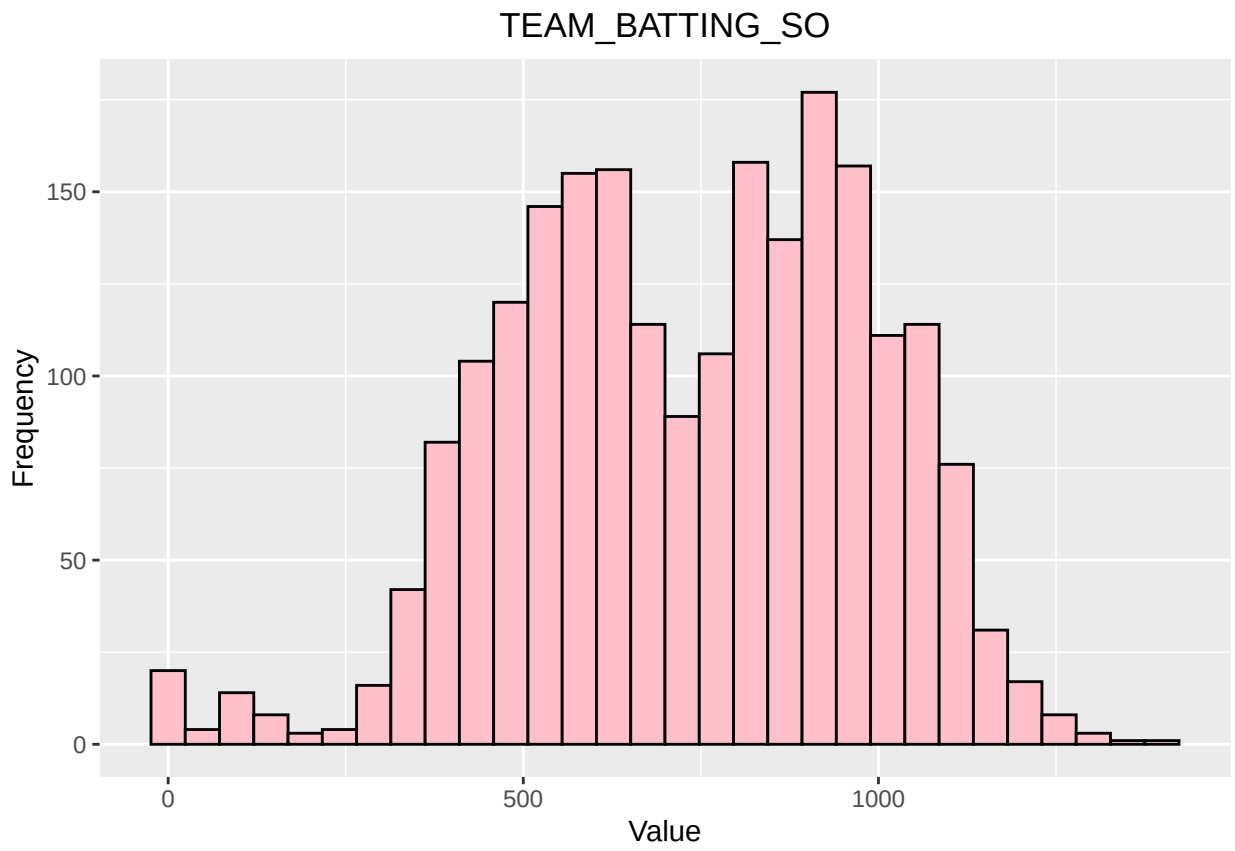
```
## Warning: Removed 102 rows containing non-finite values (stat_bin).
```




```
#since this variable has a noticable skew, I will impute using the kNN (I can also do median)


ggplot(rawTrainX, aes(x=TEAM_BATTING_SO)) +
  geom_histogram(color="black", fill="pink") +
  labs(title = "TEAM_BATTING_SO", y = "Frequency", x = "Value")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

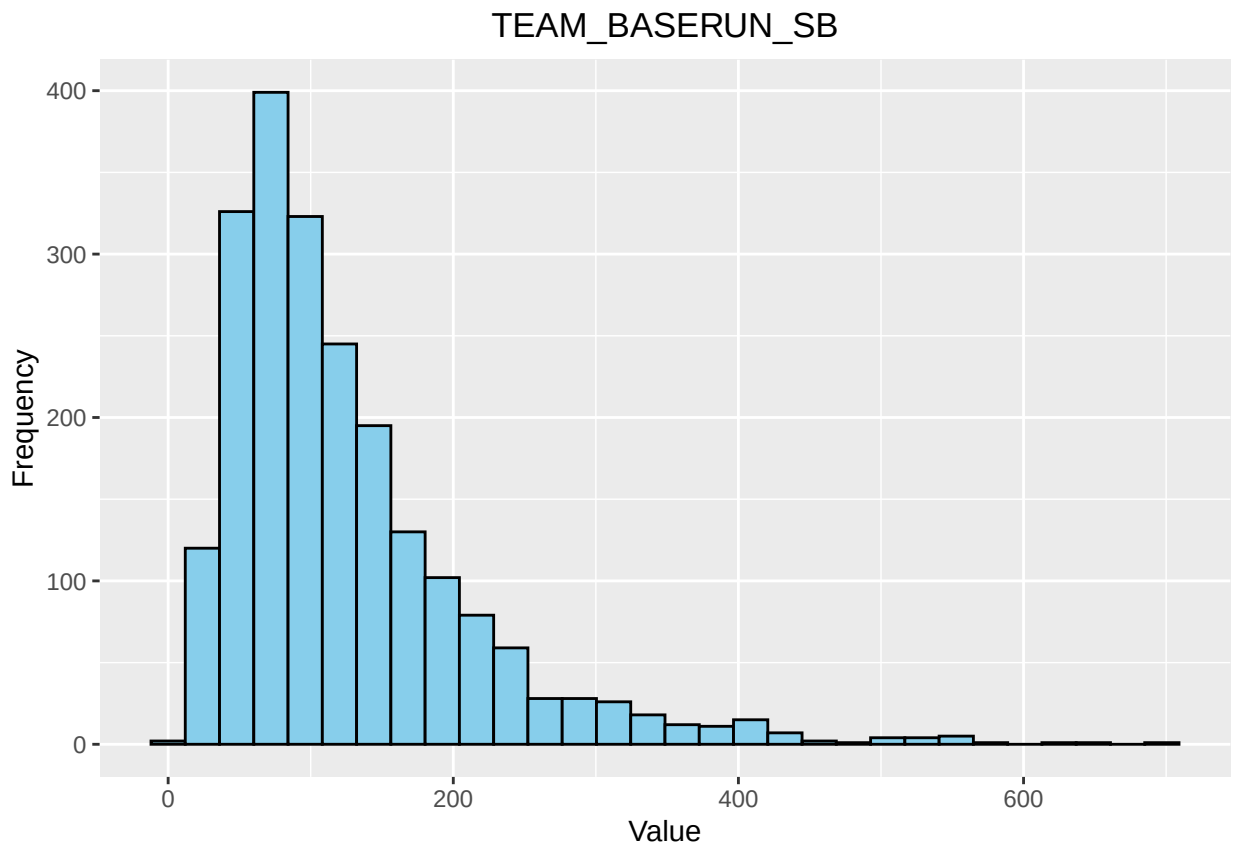
```
## Warning: Removed 102 rows containing non-finite values (stat_bin).
```




```
#since this variable is normally distributed, I will impute using the kNN (I can also do mean)

ggplot(rawTrainX, aes(x=TEAM_BASERUN_SB)) +
  geom_histogram(color="black", fill="skyblue") +
  labs(title = "TEAM_BASERUN_SB", y = "Frequency", x = "Value")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

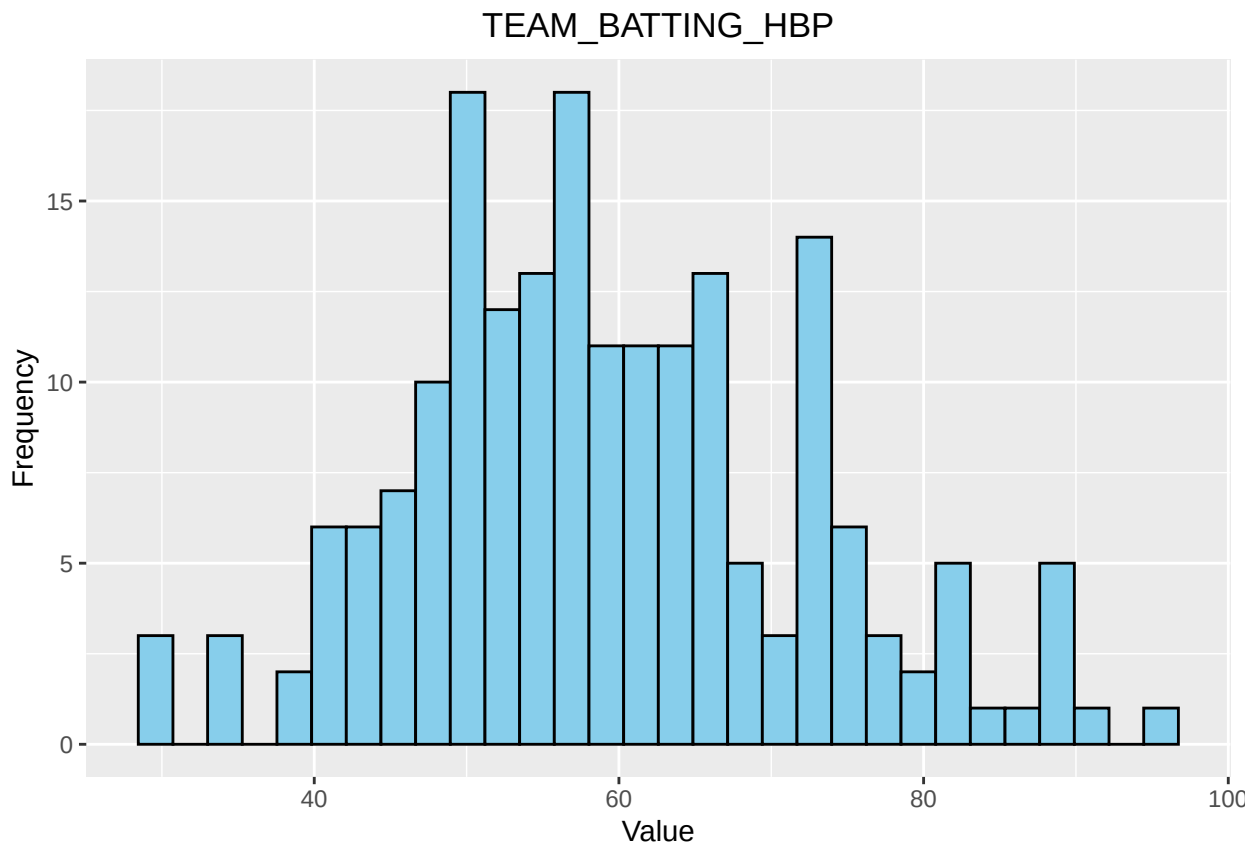
```
## Warning: Removed 131 rows containing non-finite values (stat_bin).
```




```
ggplot(rawTrainX, aes(x=TEAM_BATTING_HBP)) +
  geom_histogram(color="black", fill="skyblue") +
  labs(title = "TEAM_BATTING_HBP", y = "Frequency", x = "Value")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2085 rows containing non-finite values (stat_bin).
```




```
#KNN is chosen because it will fix all our missing data issues. As part of this preprocessing step I al
#and center each variable
```

Set A

- dropped TEAM_BATTING_HBP due to mostly NAs
- Knn Imputing
- dropped TEAM_BATTING_HR due to highly correlated

```
#drop team batting HBPbecause too many missing values
trainXA <- rawTrainX %>%
  select(-TEAM_BATTING_HBP)

#IMPUTE USING kNN
imputeProcessA <- preProcess(trainXA, method = c("knnImpute", "center", "scale"))
trainXA <- predict(imputeProcessA, trainXA)

#we can see there are no missing values, and all variables are scaled and centered
summary(trainXA)
```

```
##  TEAM_BATTING_H    TEAM_BATTING_2B    TEAM_BATTING_3B    TEAM_BATTING_HR
##  Min.   :-3.9993   Min.   :-3.68038   Min.   :-1.9776   Min.   :-1.64521
##  1st Qu.:-0.5966   1st Qu.:-0.71038   1st Qu.:-0.7606   1st Qu.:-0.95153
##  Median :-0.1056   Median :-0.06938   Median :-0.2953   Median : 0.03944
```

```
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.4702   3rd Qu.: 0.67846   3rd Qu.: 0.5995   3rd Qu.: 0.78267
##  Max.   : 7.5020   Max.   : 4.63134   Max.   : 6.0042   Max.   : 2.71505
##  TEAM_BATTING_BB    TEAM_BATTING_SO    TEAM_BASERUN_SB     TEAM_BASERUN_CS
##  Min.   :-4.08866   Min.   :-2.95987   Min.   :-1.421120   Min.   :-2.3002
##  1st Qu.:-0.41215   1st Qu.:-0.73878   1st Qu.:-0.657945   1st Qu.:-0.4271
##  Median : 0.08511   Median :-0.02255   Median :-0.236490   Median : 0.2263
##  Mean   : 0.00000   Mean   :-0.01895   Mean   :-0.001065   Mean   : 0.8365
##  3rd Qu.: 0.63944   3rd Qu.: 0.76207   3rd Qu.: 0.333043   3rd Qu.: 1.7510
##  Max.   : 3.06871   Max.   : 2.66931   Max.   : 6.518176   Max.   : 6.4556
##  TEAM_PITCHING_H    TEAM_PITCHING_HR   TEAM_PITCHING_BB   TEAM_PITCHING_SO
##  Min.   :-0.45649   Min.   :-1.72432   Min.   :-3.32422   Min.   :-1.47849
##  1st Qu.:-0.25604   1st Qu.:-0.90864   1st Qu.:-0.46291   1st Qu.:-0.36392
##  Median :-0.18567   Median : 0.02123   Median :-0.09923   Median :-0.03748
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   :-0.01230
##  3rd Qu.:-0.06874   3rd Qu.: 0.72271   3rd Qu.: 0.34860   3rd Qu.: 0.25180
##  Max.   :20.15349   Max.   : 3.87123   Max.   :18.58645   Max.   :33.37691
##  TEAM_FIELDING_E    TEAM_FIELDING_DP
##  Min.   :-0.79677   Min.   :-3.59897
##  1st Qu.:-0.52456   1st Qu.:-0.66299
##  Median :-0.38407   Median : 0.02334
##  Mean   : 0.00000   Mean   :-0.05286
##  3rd Qu.: 0.01216   3rd Qu.: 0.59528
##  Max.   : 7.25079   Max.   : 3.11183
```

```
#Lets look at some highly correlated variables and drop them
findCorrelation(cor(trainXA),
                cutoff = 0.75,
                verbose = TRUE,
                names = TRUE)
```

```
## Compare row 4  and column  10 with corr  0.969
##   Means:  0.464 vs 0.314 so flagging column 4
## All correlations <= 0.75
```

```
## [1] "TEAM_BATTING_HR"
```

```
#Lets drop the highly correlated variable
trainXA <- trainXA %>%
  select(-TEAM_BATTING_HR)
```

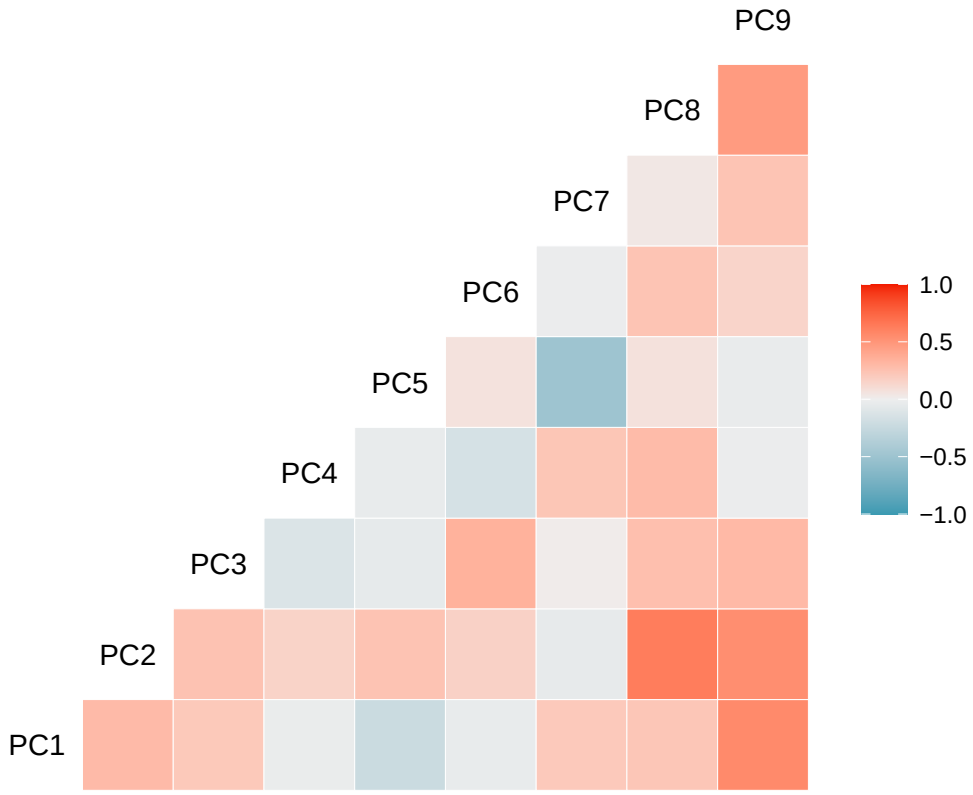
```
#Now we are ready for splitting the data
set.seed(222)
#create the train index using a column
#Here that is 80/20 split
trainIndex <- createDataPartition(trainXA$TEAM_BATTING_H, list=FALSE, p = 0.8, times = 1)
trainXa <- trainXA[trainIndex,]
testXa <- trainXA[-trainIndex,]
trainYa <- rawTrainY[trainIndex]
testYa <- rawTrainY[-trainIndex]
```

Set B

```
#With PCA & Knn

#Feature Engineer using PCA
imputeProcessB <- preProcess(rawTrainX, method = c("knnImpute","pca", "center", "scale"))
trainXb <- predict(imputeProcessB, rawTrainX)

#we can see there are no missing values, and all variables are scaled and centered
summary(trainXb)
```

```
##       PC1                PC2                PC3                PC4
##  Min.   :-9.667360   Min.   :-15.04623   Min.   :-2.02105   Min.   :-6.82943
##  1st Qu.:-1.089240   1st Qu.: -0.95054   1st Qu.:-0.82700   1st Qu.:-0.70144
##  Median :-0.075899   Median : -0.24111   Median :-0.21190   Median :-0.15193
##  Mean   :-0.000394   Mean   :  0.01521   Mean   :-0.04563   Mean   :-0.06643
##  3rd Qu.: 1.193678   3rd Qu.:  0.79230   3rd Qu.: 0.49006   3rd Qu.: 0.45498
##  Max.   : 5.202078   Max.   :  7.48870   Max.   : 7.92710   Max.   : 6.14405
##       PC5                PC6                PC7                PC8
##  Min.   :-2.6120   Min.   :-3.745328   Min.   :-2.0642   Min.   :-1.8105
##  1st Qu.:-0.3636   1st Qu.:-0.462664   1st Qu.:-0.1690   1st Qu.:-0.5024
##  Median : 0.2040   Median : 0.006512   Median : 0.1978   Median :-0.1379
##  Mean   : 0.1817   Mean   : 0.023438   Mean   : 0.2400   Mean   : 0.0222
##  3rd Qu.: 0.7498   3rd Qu.: 0.451934   3rd Qu.: 0.6050   3rd Qu.: 0.3170
##  Max.   : 3.6529   Max.   :13.347294   Max.   : 2.5043   Max.   : 7.0597
##       PC9
##  Min.   :-4.1565
##  1st Qu.:-0.8584
##  Median :-0.2350
##  Mean   :-0.1075
##  3rd Qu.: 0.5575
##  Max.   : 5.2765
```

```
#Lets look at the correlation (none higher than 75)
ggcorr(trainXb)
```




```
#Now we are ready for splitting the data
set.seed(211)
#create the train index using a column
#Here that is 80/20 split
trainIndexB <- createDataPartition(trainXb$PC1, p = 0.8, list=FALSE)
trainXB <- trainXb[trainIndexB,]
testXb <- trainXb[-trainIndexB,]
trainYb <- rawTrainY[trainIndexB]
testYb <- rawTrainY[-trainIndexB]
```

Set C

- imuted using mean and median for specific variables

```
#Impute using mean and median depending on shape
trainXc = rawTrainX  %>%
  mutate(TEAM_BASERUN_CS = ifelse(is.na(rawTrainX$TEAM_BASERUN_CS), median(rawTrainX$TEAM_BASERUN_CS,na
  mutate(TEAM_BATTING_SO = ifelse(is.na(rawTrainX$TEAM_BATTING_SO), mean(rawTrainX$TEAM_BATTING_SO,na.rm
  mutate(TEAM_FIELDING_DP = ifelse(is.na(rawTrainX$TEAM_FIELDING_DP), mean(rawTrainX$TEAM_FIELDING_DP,na
  mutate(TEAM_PITCHING_SO = ifelse(is.na(rawTrainX$TEAM_PITCHING_SO), median(rawTrainX$TEAM_PITCHING_SO
  mutate(TEAM_BASERUN_SB = ifelse(is.na(rawTrainX$TEAM_BASERUN_SB), median(rawTrainX$TEAM_BASERUN_SB,na
  mutate(TEAM_BATTING_HBP = ifelse(is.na(rawTrainX$TEAM_BATTING_HBP), mean(rawTrainX$TEAM_BATTING_HBP,na

#highly correlated variables TEAM_BATTING_HR and TEAM_PITCHN_HR
findCorrelation(
```

```
  cor(trainXc),
  cutoff = 0.75,
  verbose = FALSE,
  names = TRUE)
```

```
## [1] "TEAM_BATTING_HR"
```

```
#Linear combination of correlated variables
trainXc = trainXc %>%
  mutate(Batting_Pitching_HR = TEAM_BATTING_HR + TEAM_PITCHING_HR) %>%
  select(-TEAM_BATTING_HR,-TEAM_PITCHING_HR)
```

```
#Now we are ready for splitting the data
set.seed(221)
#create the train index using a column
#Here that is 80/20 split
trainIndexC <- createDataPartition(trainXc$TEAM_BATTING_H, p = 0.8, list=FALSE)
trainXC <- trainXc[trainIndexC,]
testXc <- trainXc[-trainIndexC,]
trainYc <- rawTrainY[trainIndexC]
testYc <- rawTrainY[-trainIndexC]
```

Build Models

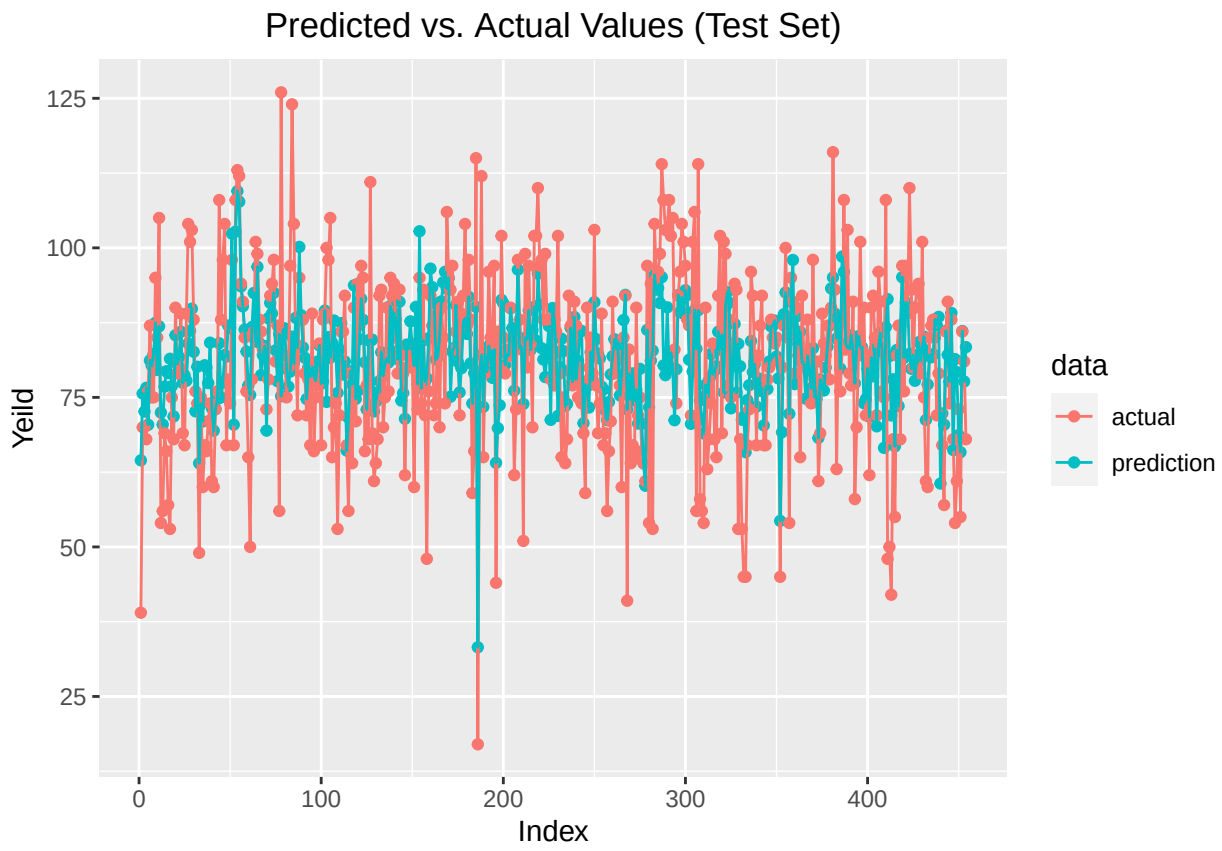
```
#This model using train/test data A
#Set up control to be 10-fold-crossvalidation
ctrl <- trainControl(method = "cv", number = 10)

#Mulitple linear regression A
#train the model
lrOne <- train(trainXa, trainYa, method = "lm", trControl = ctrl)

# something is wrong here
#make predictions
predOne <- predict(lrOne$finalModel, testXa)

#accuracy measures
acc1 <- postResample(pred = predOne, obs = testYa)

#change to df
predOne <- as.data.frame(predOne)
#Df for plotting
plotDataOne <- data.frame(index = seq(1,length(testYa),1), prediction = predOne$predOne, actual = testYa

#Long Format
plotDataOne <- gather(plotDataOne, "data", "value", -index)

#Print variables that contribute
varImp(lrOne)
```

```
## lm variable importance
##
##                   Overall
## TEAM_BATTING_H   100.000
## TEAM_FIELDING_E   70.416
## TEAM_FIELDING_DP  54.928
## TEAM_PITCHING_HR  48.018
## TEAM_BATTING_SO   27.034
## TEAM_PITCHING_SO  25.747
## TEAM_BASERUN_SB   18.417
## TEAM_BATTING_3B   13.748
## TEAM_BATTING_BB   13.629
## TEAM_BATTING_2B   10.149
## TEAM_PITCHING_H    9.904
## TEAM_BASERUN_CS    8.879
## TEAM_PITCHING_BB   0.000
```

```r
# we can remove TEAM_PITCHING_BB based on variable importance
#display
kable(acc1)
```

	x
RMSE	12.3734569
Rsquared	0.3678179
MAE	9.5460039

```r
#Plot predicted and actual data
ggplot(plotDataOne) +
  geom_point(aes(x = index, y = value, color = data)) +
  geom_line(aes(x = index, y = value, color = data)) +
  labs(x='Index', y="Yeild", title='Predicted vs. Actual Values (Test Set)') +
  theme( plot.title = element_text(hjust = 0.5))
```




```
#Mulitple linear regression B
#train the model
lrTwo <- train(trainXB,trainYb, method = "lm", trControl = ctrl)

#make predictions
predTwo <- predict(lrTwo$finalModel, testXb)

#accuracy measures
acc2 <- postResample(pred = predTwo, obs = testYb)
#change to df
predTwo <- as.data.frame(predTwo)
#Df for plotting
plotDataTwo <- data.frame(index = seq(1,length(testYb),1), prediction = predTwo$predTwo, actual = testYb

#Long Format
plotDataTwo <- gather(plotDataTwo, "data", "value", -index)

#Print variables that contribute
varImp(lrTwo)
```

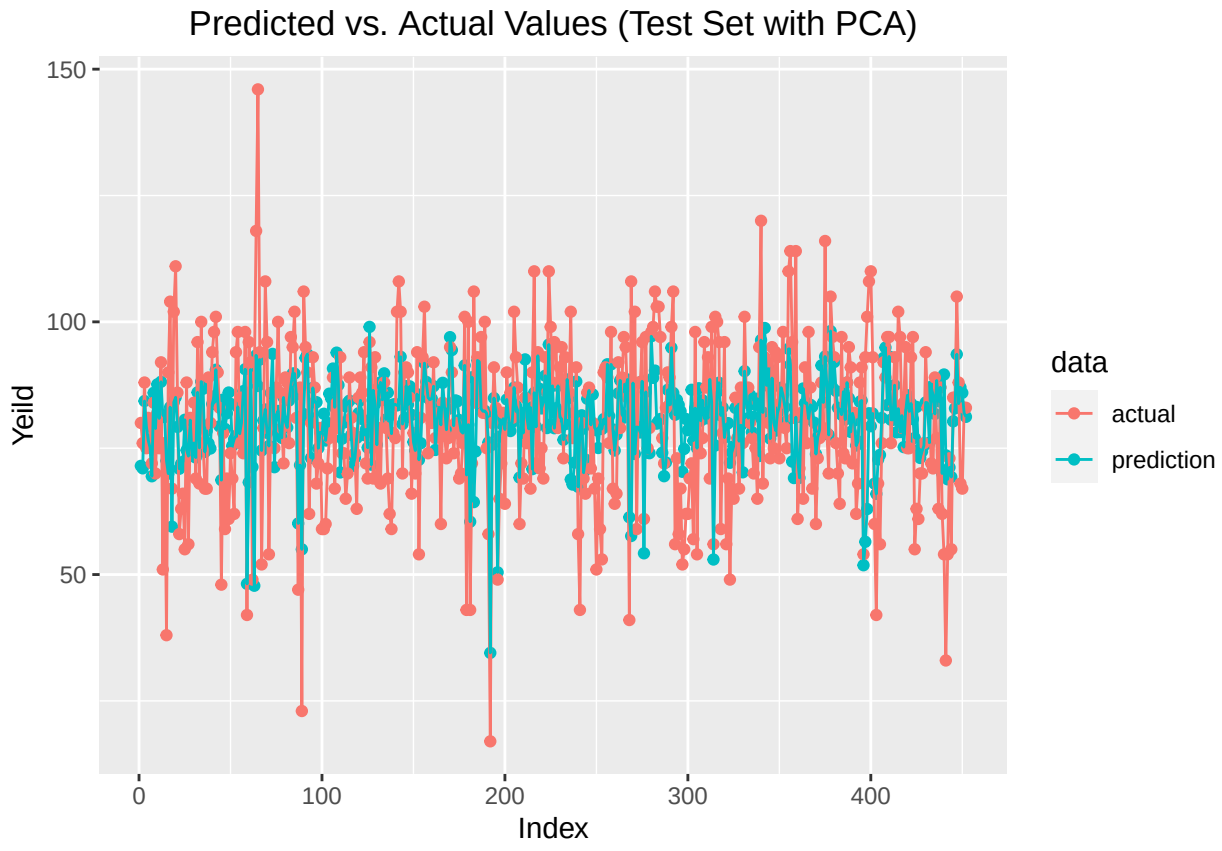
```
## lm variable importance
##
##     Overall
## PC2  100.00
## PC1   93.54
## PC8   68.57
```

```
## PC5   37.07
## PC3   33.89
## PC7   30.62
## PC4   19.77
## PC6    8.76
## PC9    0.00
```

```
#display
kable(acc2)
```

	x
RMSE	13.8556343
Rsquared	0.2585644
MAE	10.8950167

```
#Plot predicted and actual data
ggplot(plotDataTwo) +
  geom_point(aes(x = index, y = value, color = data)) +
  geom_line(aes(x = index, y = value, color = data)) +
  labs(x='Index', y="Yeild", title='Predicted vs. Actual Values (Test Set with PCA)') +
  theme( plot.title = element_text(hjust = 0.5))
```




```
#May Want to remove PC9 because it does not contribute
```

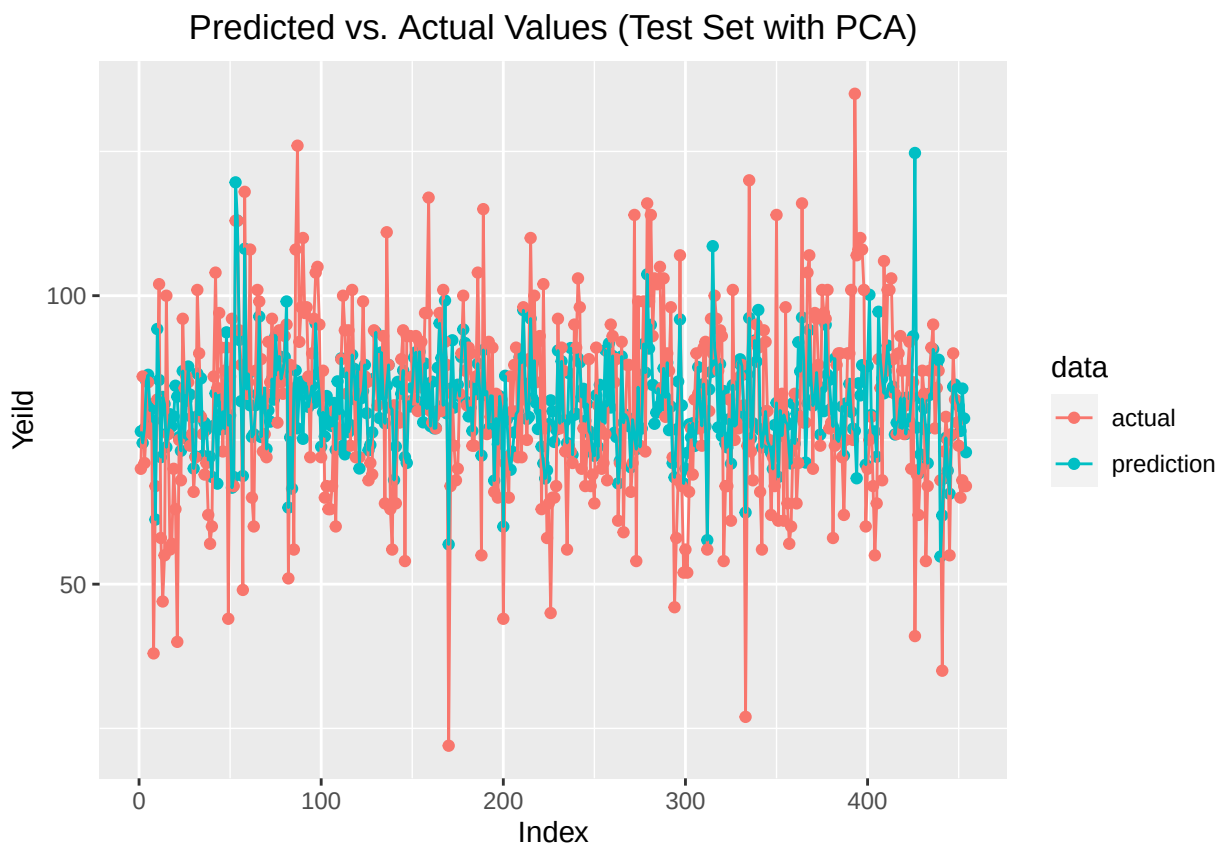
```r
#Mulitple linear regression C
#train the model
lrThree <- train(trainXC,trainYc, method = "lm", trControl = ctrl)

#make predictions
predThree <- predict(lrThree$finalModel, testXc)

#accuracy measures
acc3 <- postResample(pred = predThree, obs = testYc)
#change to df
predThree <- as.data.frame(predThree)
#Df for plotting
plotDataThree <- data.frame(index = seq(1,length(testYc),1), prediction = predThree$predThree, actual =

#Long Format
plotDataThree <- gather(plotDataThree, "data", "value", -index)

#Print variables that contribute
varImp(lrThree)
```

```
## lm variable importance
##
##                     Overall
## TEAM_BATTING_H      100.000
## TEAM_FIELDING_DP     72.197
## TEAM_FIELDING_E      63.678
## Batting_Pitching_HR  56.104
## TEAM_BASERUN_SB      42.771
## TEAM_PITCHING_SO     38.009
## TEAM_BATTING_SO      37.739
## TEAM_BATTING_3B      22.529
## TEAM_BATTING_BB      20.924
## TEAM_BATTING_2B      16.946
## TEAM_PITCHING_BB     13.439
## TEAM_PITCHING_H      11.435
## TEAM_BATTING_HBP      6.661
## TEAM_BASERUN_CS       0.000
```

```r
#team baserun cs should be deleted not important

#display
kable(acc3)
```

	x
RMSE	14.0302491
Rsquared	0.2407181
MAE	10.7604881

```r
#Plot predicted and actual data
ggplot(plotDataThree) +
  geom_point(aes(x = index, y = value, color = data)) +
  geom_line(aes(x = index, y = value, color = data)) +
  labs(x='Index', y="Yeild", title='Predicted vs. Actual Values (Test Set with PCA)') +
  theme( plot.title = element_text(hjust = 0.5))
```



I want to add pmm imputation model # can we use ridge regression or random forest?

Select Models

- display accuracy