test

load the data

```
# yankee_df_dum : removed NA observatiosn // add dummy_code
yankee_df = read.csv('yankee_df_dum.csv')
set.seed(1)  ## set the random seed before using the function generating random numbers
ran <- sample(1:nrow(yankee_df),.9*nrow(yankee_df))
nor <-function(x) { (x-min(x))/(max(x)-min(x))}
yankee_df_train<-yankee_df[ran,]
yankee_df_test<-yankee_df[-ran,]
```

EDA

eda

KNN

```
yankee_df_nor = yankee_df
yankee_df_nor[,c(2,6,8)] = apply(yankee_df[,c(2,6,8)],2,nor)
 yankee_df_reg<-yankee_df
Win_outcome <- yankee_df_reg %>% select(Win)
 yankee_df_reg<- yankee_df_reg %>% select(-Win)
 str(yankee_df_reg)
```

```
## 'data.frame':    578 obs. of  37 variables:
##  $ X                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Loss             : int  74 65 58 61 61 67 65 68 73 59 ...
##  $ League           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Athletics        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Blue.Jays        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Braves           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Brewers          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cardinals        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cubs             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Diamondbacks     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Dodgers          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Giants           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Indians          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Mariners         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Marlins          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Mets             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Nationals        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Orioles          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Padres           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Phillies         : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ Pirates          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Rangers          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Rays             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Red.Sox          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Reds             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Rockies          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Royals           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Twins            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ White.Sox        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Yankees          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Angels           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ year             : int  2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ...
##  $ avg_player_salary: int  3190974 3541674 4342365 6109992 6304673 8237537 7786523 7585561 8363263 7
##  $ playoffs         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Run_Scored       : int  871 804 897 877 897 886 930 968 789 915 ...
##  $ place            : int  1 1 1 1 1 1 1 2 3 1 ...
##  $ Div_Champs       : int  0 0 0 0 0 0 0 1 1 1 ...
```

```
knn.reg.fit = knn.reg(yankee_df_train[,-2], yankee_df_test[,-2],y=yankee_df_train[,2],k=4)
#predicted Wins
print(knn.reg.fit)
```

```
## Prediction:
##  [1] 69.00 60.75 85.75 85.25 70.25 72.25 73.50 71.75 92.75 81.75 82.75 72.50
## [13] 91.25 87.25 77.00 74.50 92.75 82.50 80.50 85.00 78.25 70.50 87.00 84.25
## [25] 63.00 91.25 79.50 71.75 77.25 86.50 73.50 83.75 93.50 73.25 80.25 80.25
## [37] 89.00 78.00 79.50 85.75 84.50 79.50 83.25 73.25 81.25 75.50 75.00 82.75
## [49] 80.00 85.25 81.75 86.00 79.50 83.50 88.00 73.00 86.75 85.25
```

```
#Runs Scored
knn.reg.fit = knn.reg(yankee_df_train[,-36], yankee_df_test[,-36],y=yankee_df_train[,36],k=4)
print(knn.reg.fit)
```

```
## Prediction:
##  [1] 816.50 852.00 741.75 748.25 792.00 762.75 761.50 848.25 791.50 751.25
## [11] 720.00 714.25 671.75 761.00 769.00 744.25 791.50 661.00 753.75 710.25
## [21] 729.75 803.75 713.75 701.75 840.25 671.75 742.00 760.25 718.50 765.00
## [31] 743.75 788.75 707.50 674.25 704.25 723.00 759.75 733.50 786.25 718.00
## [41] 717.50 797.00 712.50 707.75 654.00 757.00 684.50 769.00 649.00 703.50
## [51] 745.00 743.00 742.00 705.50 763.75 822.75 682.75 748.25
```

Fit KNN

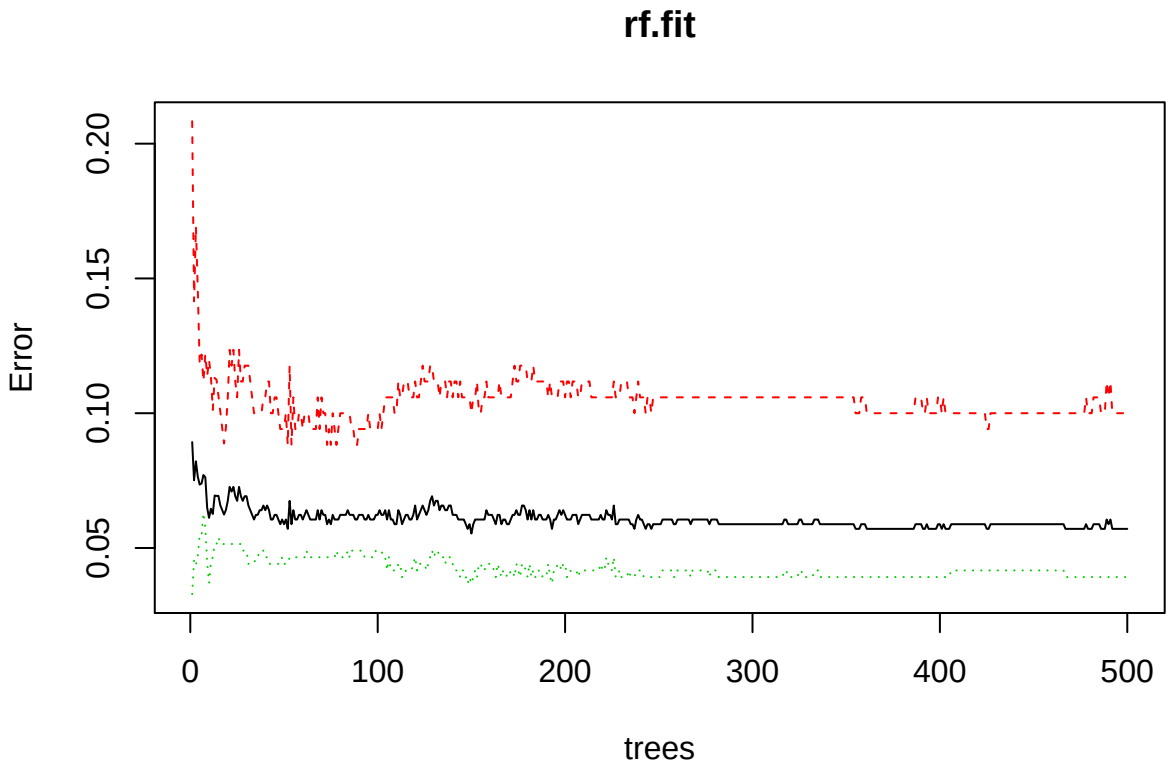
```
knn.fit = knn.reg(yankee_df_train[,-2], yankee_df_test[,-2],y=yankee_df_train[,2],k=4)
```

#random forest and decision tree

```
yankee_df = read.csv('yankee_df_dum.csv')
library(rpart)
library(class)
library(randomForest)
above_500= ifelse(yankee_df$Win>=83, "No","Yes")
yankee_df$playoffs = as.factor(yankee_df$playoffs)  # change the variable as a factor -
winning_season=data.frame(yankee_df, above_500)
randomForest(playoffs~., data=yankee_df)
```

```
##
## Call:
##  randomForest(formula = playoffs ~ ., data = yankee_df)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##         OOB estimate of  error rate: 4.5%
## Confusion matrix:
##     0   1 class.error
## 0 156  14  0.08235294
## 1  12 396  0.02941176
```

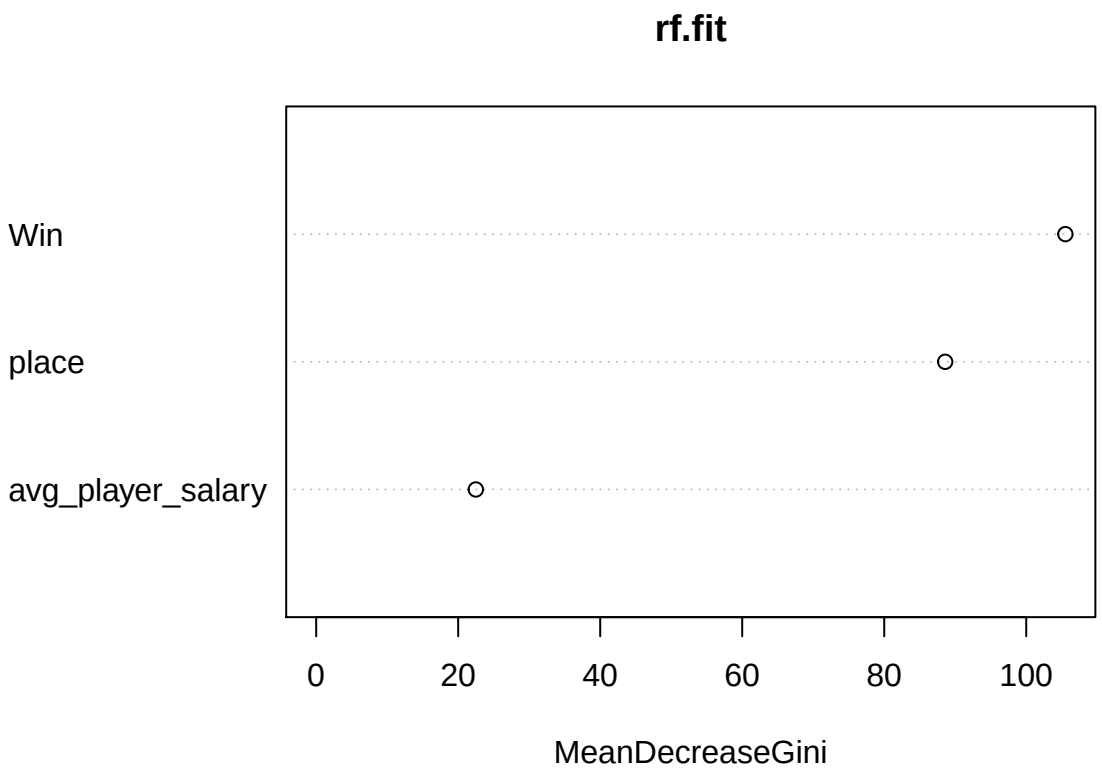
```
output.tree<-ctree(playoffs~Win+avg_player_salary+place, data=yankee_df)
rf.fit<-randomForest(playoffs~Win+avg_player_salary+place, data=yankee_df)
plot(rf.fit)
```



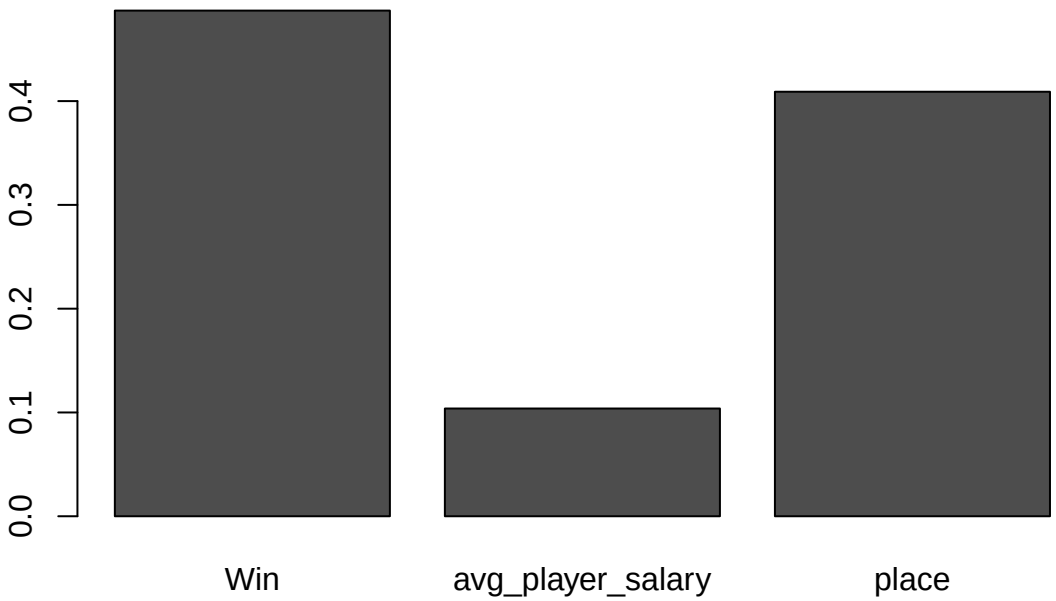

```
(VI_F=importance(rf.fit))
```

```
##                   MeanDecreaseGini
## Win                      105.51286
## avg_player_salary         22.48170
## place                     88.58102
```

```
varImpPlot(rf.fit,type=2)
```




```
barplot(t(VI_F/sum(VI_F)))
```



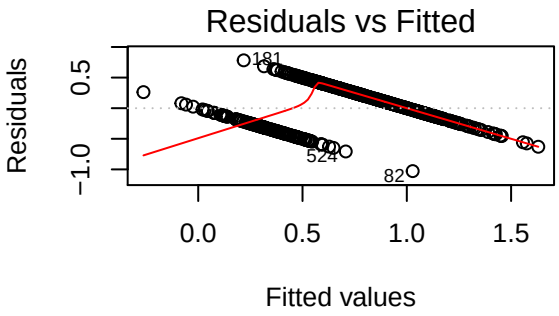

```
getTree(rf.fit, 1, labelVar=TRUE)
```

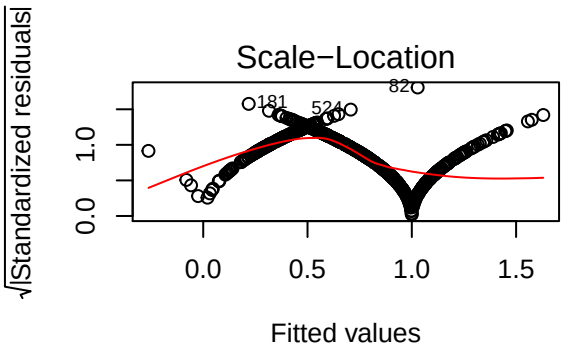
```
##    left daughter right daughter         split var split point status prediction
## 1              2              3             place         2.5      1       <NA>
## 2              4              5 avg_player_salary   6055072.0      1       <NA>
## 3              6              7             place         3.5      1       <NA>
## 4              8              9 avg_player_salary   5919339.0      1       <NA>
## 5              0              0              <NA>         0.0     -1          0
## 6             10             11               Win        86.5      1       <NA>
```

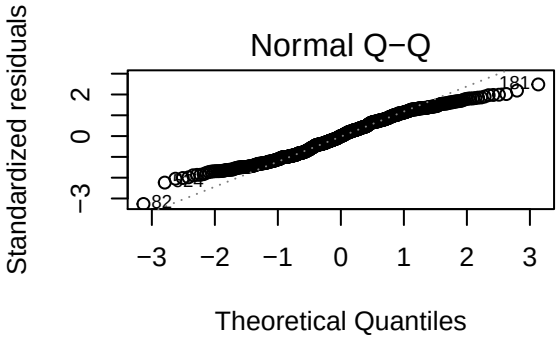
```
## 7              0              0              <NA>        0.0     -1          1
## 8             12             13 avg_player_salary   2663675.5      1       <NA>
## 9              0              0              <NA>        0.0     -1          1
## 10             0              0              <NA>        0.0     -1          1
## 11            14             15               Win       87.5      1       <NA>
## 12            16             17               Win       89.0      1       <NA>
## 13            18             19             place        1.5      1       <NA>
## 14            20             21 avg_player_salary   3168929.5      1       <NA>
## 15             0              0              <NA>        0.0     -1          1
## 16            22             23             place        1.5      1       <NA>
## 17            24             25             place        1.5      1       <NA>
## 18             0              0              <NA>        0.0     -1          0
## 19            26             27               Win       89.5      1       <NA>
## 20             0              0              <NA>        0.0     -1          1
## 21             0              0              <NA>        0.0     -1          0
## 22             0              0              <NA>        0.0     -1          0
## 23             0              0              <NA>        0.0     -1          1
## 24             0              0              <NA>        0.0     -1          0
## 25            28             29 avg_player_salary   2163345.0      1       <NA>
## 26             0              0              <NA>        0.0     -1          1
## 27             0              0              <NA>        0.0     -1          0
## 28            30             31               Win       90.5      1       <NA>
## 29            32             33               Win       90.5      1       <NA>
## 30             0              0              <NA>        0.0     -1          0
## 31             0              0              <NA>        0.0     -1          1
## 32             0              0              <NA>        0.0     -1          1
## 33             0              0              <NA>        0.0     -1          0
```

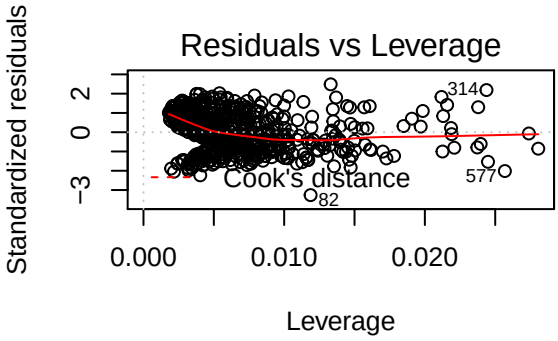
#linear regression

```
yankee_df$playoffs = as.character(yankee_df$playoffs)
yankee_df$playoffs = as.numeric(yankee_df$playoffs)
fit1 <- lm(playoffs ~ Win + avg_player_salary + Run_Scored, data=yankee_df)
layout(matrix(c(1,2,3,4),2,2))
plot(fit1)
```

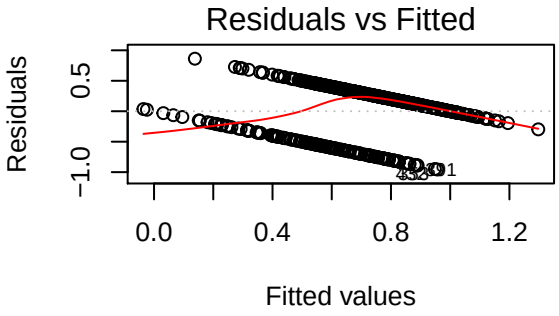


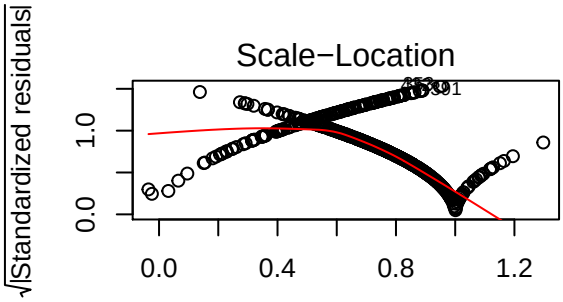


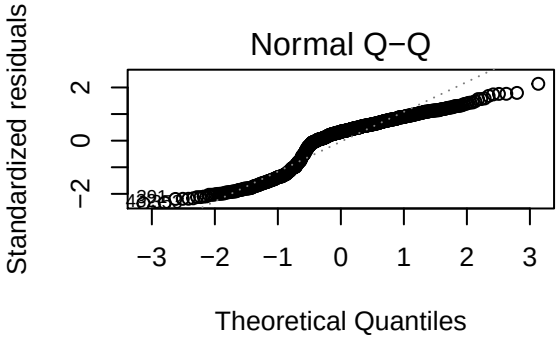


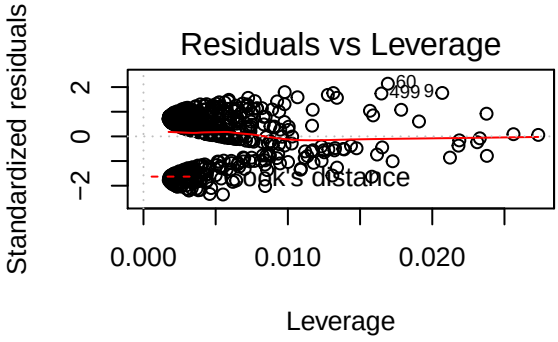



```
fit2 <- lm(playoffs ~ Run_Scored + avg_player_salary,data=yankee_df)
plot(fit2)
```









```
anova(fit1,fit2)
```

```
## Analysis of Variance Table
##
## Model 1: playoffs ~ Win + avg_player_salary + Run_Scored
## Model 2: playoffs ~ Run_Scored + avg_player_salary
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    574 57.663
## 2    575 95.226 -1   -37.562 373.91 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```