Baseball hitters salary prediction

***___Bharani ***

Do we have all the packages ?

```
pacman::p_load(ISLR,gbm, tree,rpart, rpart.plot, caret,
               data.table, MASS, ggplot2,gains,data.table, forecast, leaps, tidyverse,randomForest)
options(digits = 3)
knitr::opts_chunk$set(echo = TRUE, fig.width=12, fig.height=6, fig.path = 'Figs/')
theme_set(theme_classic())
```

1. Data Inset

```
wseries.df <- Hitters

wseries_NA.df <- wseries.df
wseries.df <- na.omit(wseries.df)
##wseries.df$League <- ifelse( wseries.df$League == 'N',0,1)
#wseries.df$Division <- ifelse( wseries.df$Division == 'W',0,1)
#wseries.df$NewLeague <- ifelse( wseries.df$NewLeague == 'N',0,1)

sum(is.na(Hitters))
```

```
## [1] 59
```

59 'NA' records were removed in this process.

2 Log transform : Salary

```
wseries.df$Salary <- log(wseries.df$Salary)
```

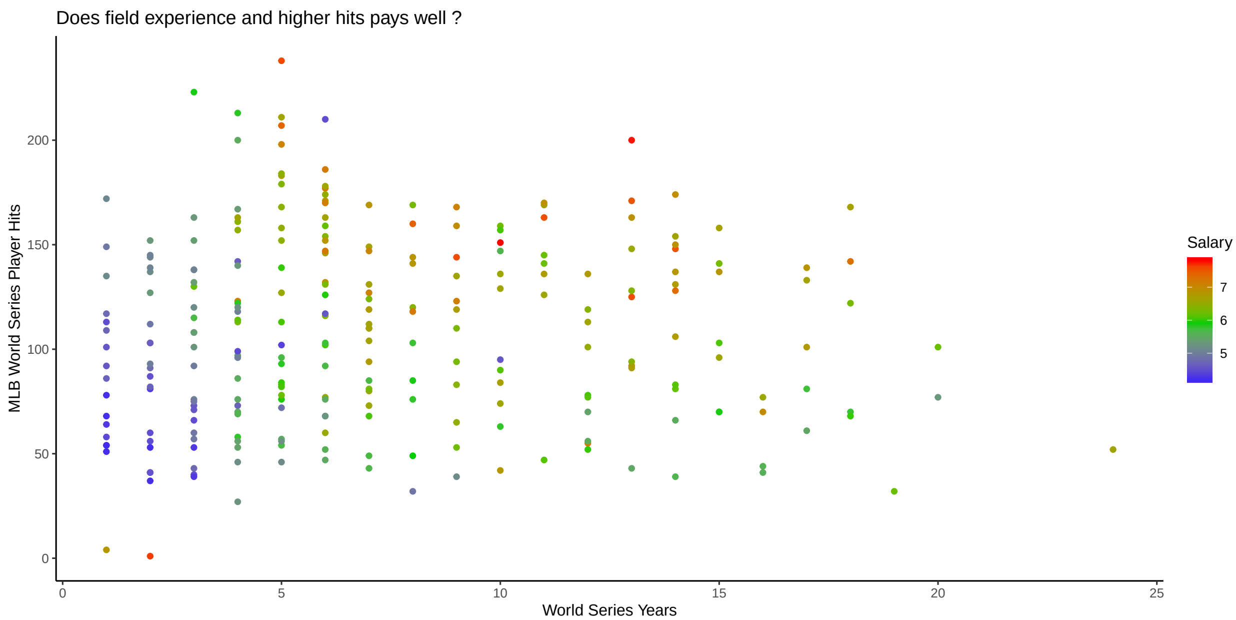
It's difficult to analyze data with high variance such as Salary. Log transformation makes the analysis relatively easy as it scales the data and closely couples the datapoints. Also it helps minimizing the effects of outliers.

3 Years ~ Hits - ggplot

```
mid <- mean(wseries.df$Salary)

ggplot(wseries.df, aes(x=wseries.df$Years,y=wseries.df$Hits,color=Salary)) +  geom_point() +
```

```
  scale_color_gradient2(midpoint=mid, low="blue", mid="green3",high="red", space ="lab" ) +
    xlab("World Series Years") +
     ylab("MLB World Series Player Hits") +
       ggtitle("Does field experience and higher hits pays well ?")
```

```
## Warning: Non Lab interpolation is deprecated
```



The interesting patters that we found are: a) In general, sportspersons with higher number of hits (more than 150s) are amongst the highest paid. b) In initial years of their careers (1-5 years) sportspersons are paid way less compared to later on in their career.

4. Linear regression model

```
wseries.lm <- regsubsets(log(Salary) ~ ., data = wseries.df,method='exhaustive')
names((summary(wseries.lm)))
```

```
## [1] "which"  "rsq"     "rss"     "adjr2"  "cp"      "bic"     "outmat" "obj"
```

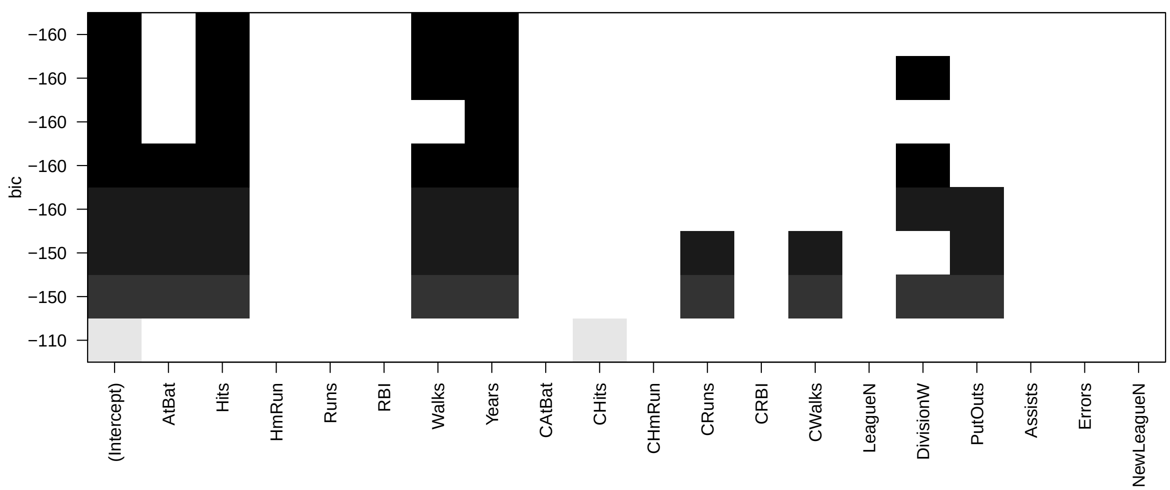
```
sum <- summary(wseries.lm)

sum$bic
```

```
## [1] -115 -159 -161 -160 -158 -156 -153 -151
```

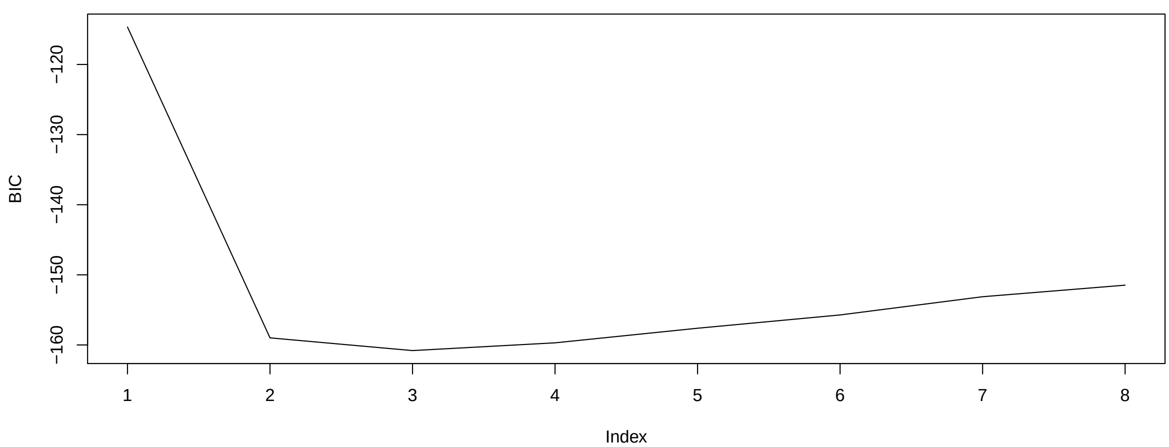
```
coef(wseries.lm,7)
```

```
## (Intercept)       AtBat        Hits       Walks       Years       CRuns
##    1.52e+00   -4.32e-04    2.06e-03    1.72e-03    1.33e-02    1.95e-04
##      CWalks     PutOuts
##   -1.85e-04    4.96e-05
```

```r
plot(wseries.lm,scale="bic")
```




```r
plot(sum$bic,xlab='Index',ylab='BIC',type='l')
```



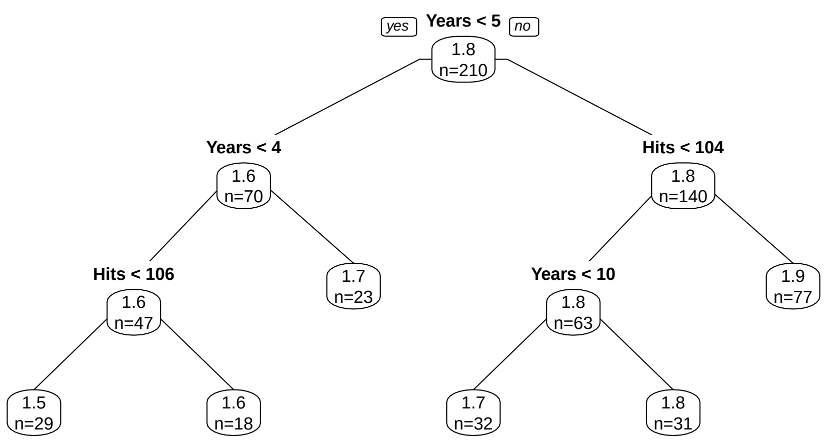
5. Data Partition

```r
set.seed(42)
training.index <- sample(1:nrow(wseries.df), 0.8 *(nrow(wseries.df)))
mlb.train <- wseries.df[training.index, ]
```

```r
mlb.test <- wseries.df[-training.index, ]
mlb.test.salary <- wseries.df[-training.index, "Salary"]
```

6. Regression Tree ~ Years + Hits

```r
set.seed(42)
regtree <- rpart(log(Salary) ~ Years + Hits, data = mlb.train)
prp(regtree, type = 1, extra = 1, split.font = 2)
```




```r
rpart.rules(regtree, cover = TRUE)
```

```
##  log(Salary)                                       cover
##          1.5 when Years <  4       & Hits <  106     14%
##          1.6 when Years <  4       & Hits >= 106      9%
##          1.7 when Years is 4 to  5                   11%
##          1.7 when Years is 5 to 10 & Hits <  104     15%
##          1.8 when Years >=      10 & Hits <  104     15%
##          1.9 when Years >=       5 & Hits >= 104     37%
```

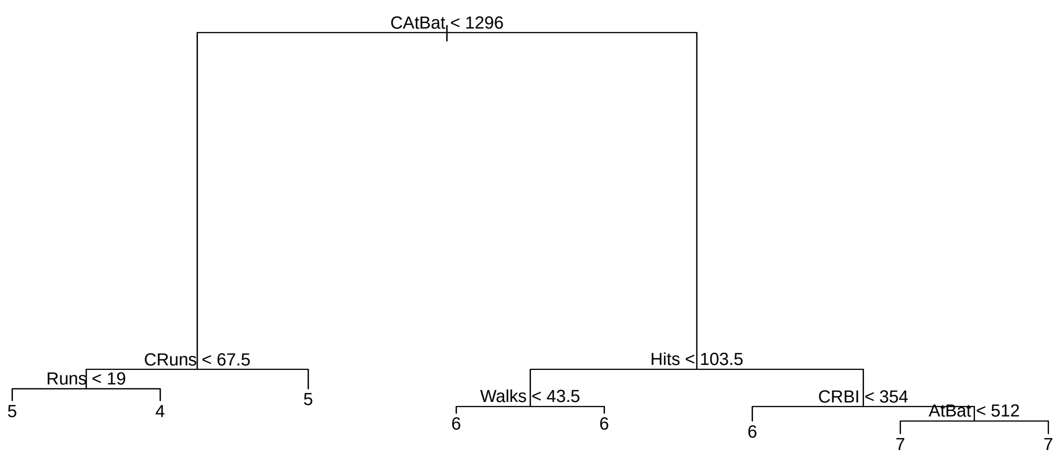
3 rules that give highest salary are: a) Years > 4.5 + Hits > 88.5 –> So when the number of hits are more than 88.5 and the years of experience is greater than 4.5 years than those sportspersons have highest salaries. b) Years > 8.5 + Hits < 88.5 –> Second highest paid sportsperson will be those that have years of experience greater than 8.5 and number of hits is less than 88.5. c) Years < 4.5 + Hits > 150.5 –> If the years of experience is less than 4.5 and number of hits is greater than 150.5 then these sporsperson will be third highest paid.

7. Regression Tree ~ All variables

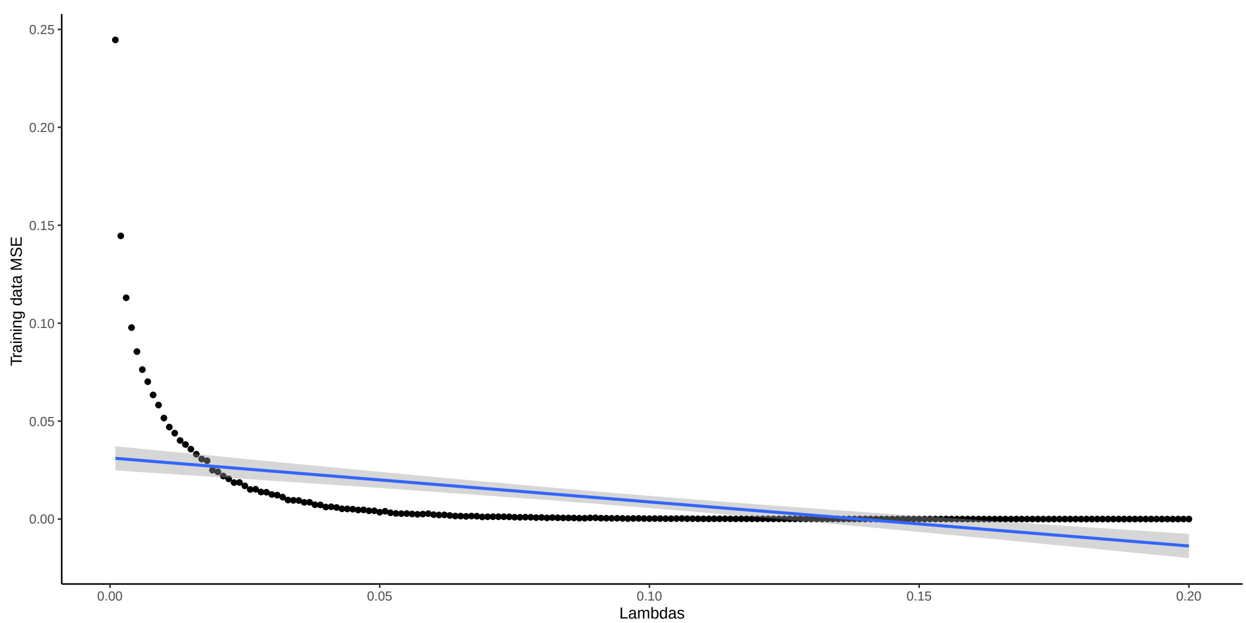
```
train <- sample(1:nrow(mlb.train), nrow(mlb.train)/2)
set.seed(42)
mlb.tree <- tree(Salary~ ., mlb.train, subset =train)
summary(mlb.tree)
```

```
##
## Regression tree:
## tree(formula = Salary ~ ., data = mlb.train, subset = train)
## Variables actually used in tree construction:
## [1] "CAtBat" "CRuns"  "Runs"   "Hits"   "Walks"  "CRBI"   "AtBat"
## Number of terminal nodes:  8
## Residual mean deviance:  0.125 = 12.1 / 97
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -0.862  -0.230   0.010   0.000   0.186   1.640
```

```
plot(mlb.tree)
text(mlb.tree,pretty=0)
```

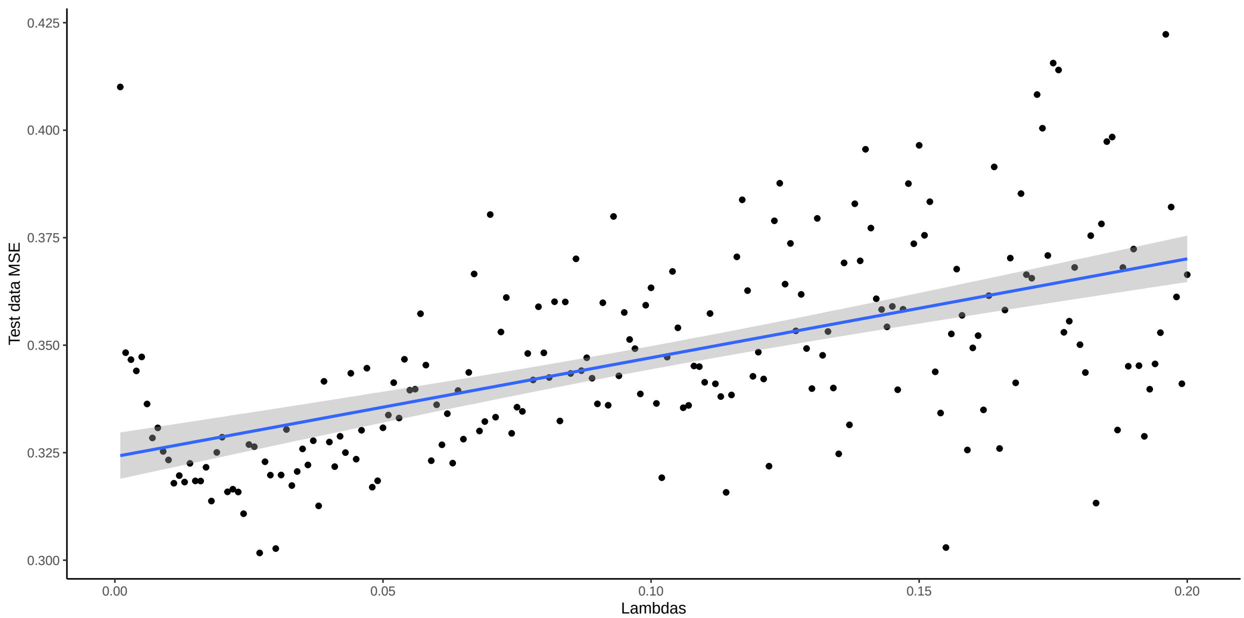


```
# Boosting for different lambdas
lambdas <- c(c(), seq(0.001, 0.2, by= 0.001)) #Starting lambda with default value 0.01

len_lambdas <- length(lambdas)
MSE.train <- rep(NA,len_lambdas)
MSE.test <- rep(NA,len_lambdas)
for( i in 1:len_lambdas)
{ boost.mlb<-gbm(Salary~., data=mlb.train, distribution = "gaussian",
                  n.trees = 1000, interaction.depth = 4,
                  shrinkage = lambdas[i], verbose = F)
  mlb.boost.pred.train <-predict(boost.mlb, mlb.train,n.trees = 1000)
  mlb.boost.pred.test <-predict(boost.mlb, mlb.test,n.trees = 1000)
  MSE.train[i] <- mean((mlb.boost.pred.train - mlb.train$Salary)^2)
  MSE.test[i] <- mean ((mlb.boost.pred.test - mlb.test$Salary)^2)
}


# Plotting of different lambdas ~ MSE for training data
ggplot(data.frame(x=lambdas,y= MSE.train), aes(x=x,y=y)) + geom_point() + geom_smooth(method=glm)+ xlab
```



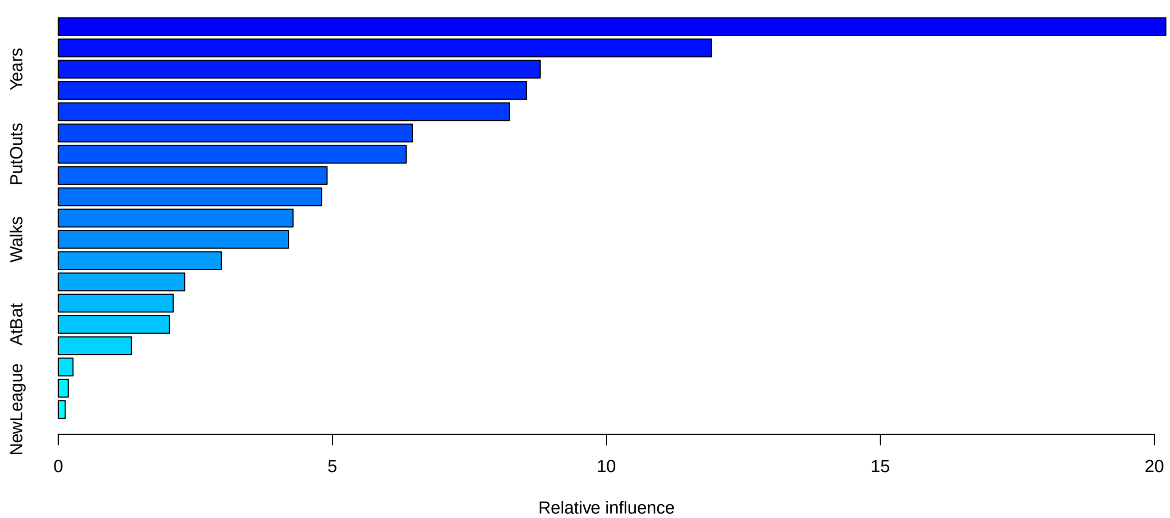
8. plot(Lambdas ~ MSE)

```
# Plotting of different lambdas ~ MSE for test data
set.seed(42)
ggplot(data.frame(x=lambdas,y= MSE.test), aes(x=x,y=y)) + geom_point() + geom_smooth(method=glm)+ xlab(
```



9. What did the boosted model say about the predictors ?

```
# Since we ran boosting for various shrinkage parameters,
# Let's take the lambda with least error from the test data.

mlb.boost.train <- gbm(Salary ~ ., data = mlb.train, distribution = "gaussian",
    n.trees = 1000, shrinkage = lambdas[which(MSE.test == min(MSE.test))])

summary(mlb.boost.train)
```



```
##                 var rel.inf
## CAtBat       CAtBat  20.208
## CRuns         CRuns  11.918
## Years         Years   8.791
## CWalks       CWalks   8.546
## CRBI           CRBI   8.230
## CHits         CHits   6.460
## PutOuts     PutOuts   6.347
## CHmRun       CHmRun   4.904
## Hits           Hits   4.804
## RBI             RBI   4.283
## Walks         Walks   4.199
## HmRun         HmRun   2.974
## Errors       Errors   2.305
## Runs           Runs   2.097
## AtBat         AtBat   2.027
## Assists     Assists   1.332
## Division   Division   0.268
## League       League   0.182
## NewLeague NewLeague   0.126
```

From the boosted model's relative influence plot : CAtBat

10. Bagging

```
set.seed(42)
bagging <- randomForest(mlb.train$Salary~., data=mlb.train, SSmtry = 19, importance = TRUE,ntree=1000)
bagging.prediction <- predict(bagging, mlb.test)

mean((bagging.prediction-mlb.test$Salary)^2)
```

```
## [1] 0.236
```