Homework 4

Group BUAN636.501-1

04/10/2020

CLASS: "BUAN 6356.501"
GROUP MEMBERS: "Sai Raghavendra Sridhar(sxs180281), Shreya Tippannawar(sst190000), Smruti Viswanath Iyer(sxi180001), Piyush Dangwal(pxd142430),Shanshan Luo(sxl130330)"

a. Load the packages:

```
if(!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(caret, data.table, ISLR, tidyr, devtools, ggplot2, tidyverse,gains, leaps, rpart,
rpart.plot, gbm , randomForest , tinytex, knitr,magrittr,dplyr,tree)
```

```
## Installing package into 'C:/Users/saira/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## package 'tidyr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\saira\AppData\Local\Temp\RtmpQtOnho\downloaded_packages
```

```
##
## tidyr installed
```

```
## Installing package into 'C:/Users/saira/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## also installing the dependencies 'glue', 'rlang'
```

```
##
##   There is a binary version available but the source version is later:
##          binary source needs_compilation
## devtools  2.2.2  2.3.0             FALSE
##
## package 'glue' successfully unpacked and MD5 sums checked
## package 'rlang' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\saira\AppData\Local\Temp\RtmpQtOnho\downloaded_packages
## installing the source package 'devtools'
```

```
##
## devtools installed
```

```
## Installing package into 'C:/Users/saira/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\saira\AppData\Local\Temp\RtmpQtOnho\downloaded_packages
```

```
##
## tidyverse installed
```

```
search()
```

```
##  [1] ".GlobalEnv"           "package:tree"         "package:dplyr"
##  [4] "package:magrittr"     "package:knitr"        "package:tinytex"
##  [7] "package:randomForest" "package:gbm"          "package:rpart.plot"
## [10] "package:rpart"        "package:leaps"        "package:gains"
## [13] "package:ISLR"         "package:data.table"   "package:caret"
## [16] "package:ggplot2"      "package:lattice"      "package:pacman"
## [19] "package:stats"        "package:graphics"     "package:grDevices"
## [22] "package:utils"        "package:datasets"     "package:methods"
## [25] "Autoloads"            "package:base"
```

b. Exploring the dataset:

```
set.seed(42)
library(e1071)
dim(Hitters)
```

```
## [1] 322  20
```

```
str(Hitters)
```

```
## 'data.frame':    322 obs. of  20 variables:
##  $ AtBat    : int  293 315 479 496 321 594 185 298 323 401 ...
##  $ Hits     : int  66 81 130 141 87 169 37 73 81 92 ...
##  $ HmRun    : int  1 7 18 20 10 4 1 0 6 17 ...
##  $ Runs     : int  30 24 66 65 39 74 23 24 26 49 ...
##  $ RBI      : int  29 38 72 78 42 51 8 24 32 66 ...
##  $ Walks    : int  14 39 76 37 30 35 21 7 8 65 ...
##  $ Years    : int  1 14 3 11 2 11 2 3 2 13 ...
##  $ CAtBat   : int  293 3449 1624 5628 396 4408 214 509 341 5206 ...
##  $ CHits    : int  66 835 457 1575 101 1133 42 108 86 1332 ...
##  $ CHmRun   : int  1 69 63 225 12 19 1 0 6 253 ...
##  $ CRuns    : int  30 321 224 828 48 501 30 41 32 784 ...
##  $ CRBI     : int  29 414 266 838 46 336 9 37 34 890 ...
##  $ CWalks   : int  14 375 263 354 33 194 24 12 8 866 ...
##  $ League   : Factor w/ 2 levels "A","N": 1 2 1 2 2 1 2 1 2 1 ...
##  $ Division : Factor w/ 2 levels "E","W": 1 2 2 1 1 2 1 2 2 1 ...
##  $ PutOuts  : int  446 632 880 200 805 282 76 121 143 0 ...
##  $ Assists  : int  33 43 82 11 40 421 127 283 290 0 ...
##  $ Errors   : int  20 10 14 3 4 25 7 9 19 0 ...
##  $ Salary   : num  NA 475 480 500 91.5 750 70 100 75 1100 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 1 2 1 2 2 1 1 1 2 1 ...
```

```
Hitters.df <- data.frame(Hitters)
summary(Hitters.df)
```

```
##      AtBat            Hits         HmRun            Runs
##  Min.   : 16.0   Min.   :  1   Min.   : 0.00   Min.   :  0.00
##  1st Qu.:255.2   1st Qu.: 64   1st Qu.: 4.00   1st Qu.: 30.25
##  Median :379.5   Median : 96   Median : 8.00   Median : 48.00
##  Mean   :380.9   Mean   :101   Mean   :10.77   Mean   : 50.91
##  3rd Qu.:512.0   3rd Qu.:137   3rd Qu.:16.00   3rd Qu.: 69.00
##  Max.   :687.0   Max.   :238   Max.   :40.00   Max.   :130.00
##
##       RBI             Walks            Years            CAtBat
##  Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0
##  1st Qu.: 28.00   1st Qu.: 22.00   1st Qu.: 4.000   1st Qu.:  816.8
##  Median : 44.00   Median : 35.00   Median : 6.000   Median : 1928.0
##  Mean   : 48.03   Mean   : 38.74   Mean   : 7.444   Mean   : 2648.7
##  3rd Qu.: 64.75   3rd Qu.: 53.00   3rd Qu.:11.000   3rd Qu.: 3924.2
##  Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0
##
##      CHits            CHmRun           CRuns             CRBI
##  Min.   :   4.0   Min.   :  0.00   Min.   :   1.0   Min.   :   0.00
##  1st Qu.: 209.0   1st Qu.: 14.00   1st Qu.: 100.2   1st Qu.:  88.75
##  Median : 508.0   Median : 37.50   Median : 247.0   Median : 220.50
##  Mean   : 717.6   Mean   : 69.49   Mean   : 358.8   Mean   : 330.12
##  3rd Qu.:1059.2   3rd Qu.: 90.00   3rd Qu.: 526.2   3rd Qu.: 426.25
##  Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.00
##
##      CWalks        League  Division    PutOuts          Assists
##  Min.   :   0.00   A:175   E:157    Min.   :   0.0   Min.   :  0.0
##  1st Qu.:  67.25   N:147   W:165    1st Qu.: 109.2   1st Qu.:  7.0
##  Median : 170.50                    Median : 212.0   Median : 39.5
##  Mean   : 260.24                    Mean   : 288.9   Mean   :106.9
##  3rd Qu.: 339.25                    3rd Qu.: 325.0   3rd Qu.:166.0
##  Max.   :1566.00                    Max.   :1378.0   Max.   :492.0
##
##      Errors          Salary       NewLeague
##  Min.   : 0.00   Min.   :  67.5   A:176
##  1st Qu.: 3.00   1st Qu.: 190.0   N:146
##  Median : 6.00   Median : 425.0
##  Mean   : 8.04   Mean   : 535.9
##  3rd Qu.:11.00   3rd Qu.: 750.0
##  Max.   :32.00   Max.   :2460.0
##                  NA's   :59
```

####*Question 1: Remove the observations with unknown salary information. How many observations were removed in this process?

```
Hitters.cleant <- Hitters[!(is.na(Hitters$Salary)),]
rows.removed <- nrow(Hitters) - nrow(Hitters.cleant)
rows.removed
```

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

```
# Verification of no 'NA' values in SALARy
sapply(Hitters.cleant$Salary, function(Salary) sum(length(which(is.na(Salary)))))
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [260] 0 0 0 0
```

```
dim(Hitters.cleant)
```

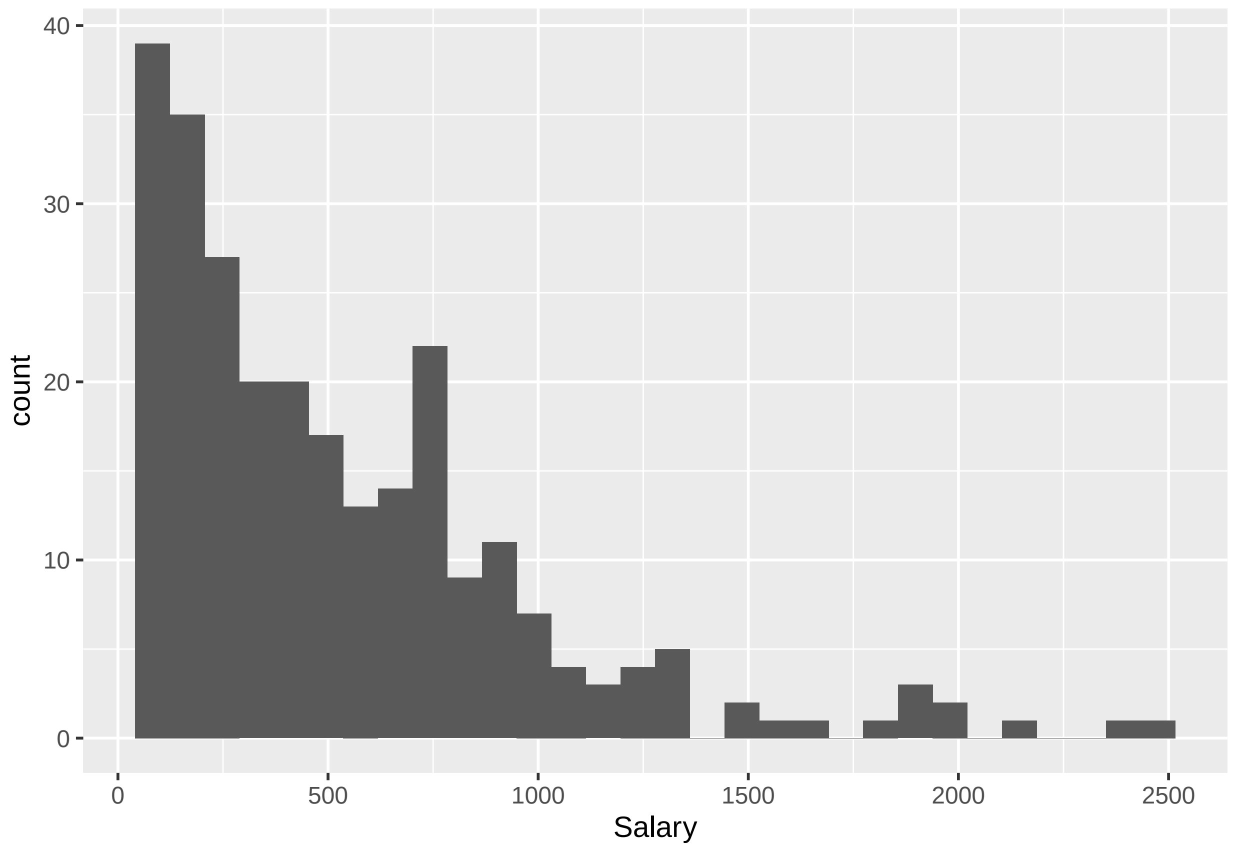
```
## [1] 263  20
```

#*Interpretation 1: Here 59 observations of 322 observations is removed resulting in total of 263 records in the new dataframe. The new dataframe is Hitters.cleant

###*Question 2 :Transform the salaries using a (natural) log transformation. Can you justify this transformation?

```
ggplot(data = Hitters.cleant, aes(Salary)) + geom_histogram()
```

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



```
skewness(Hitters.cleant$Salary)
```

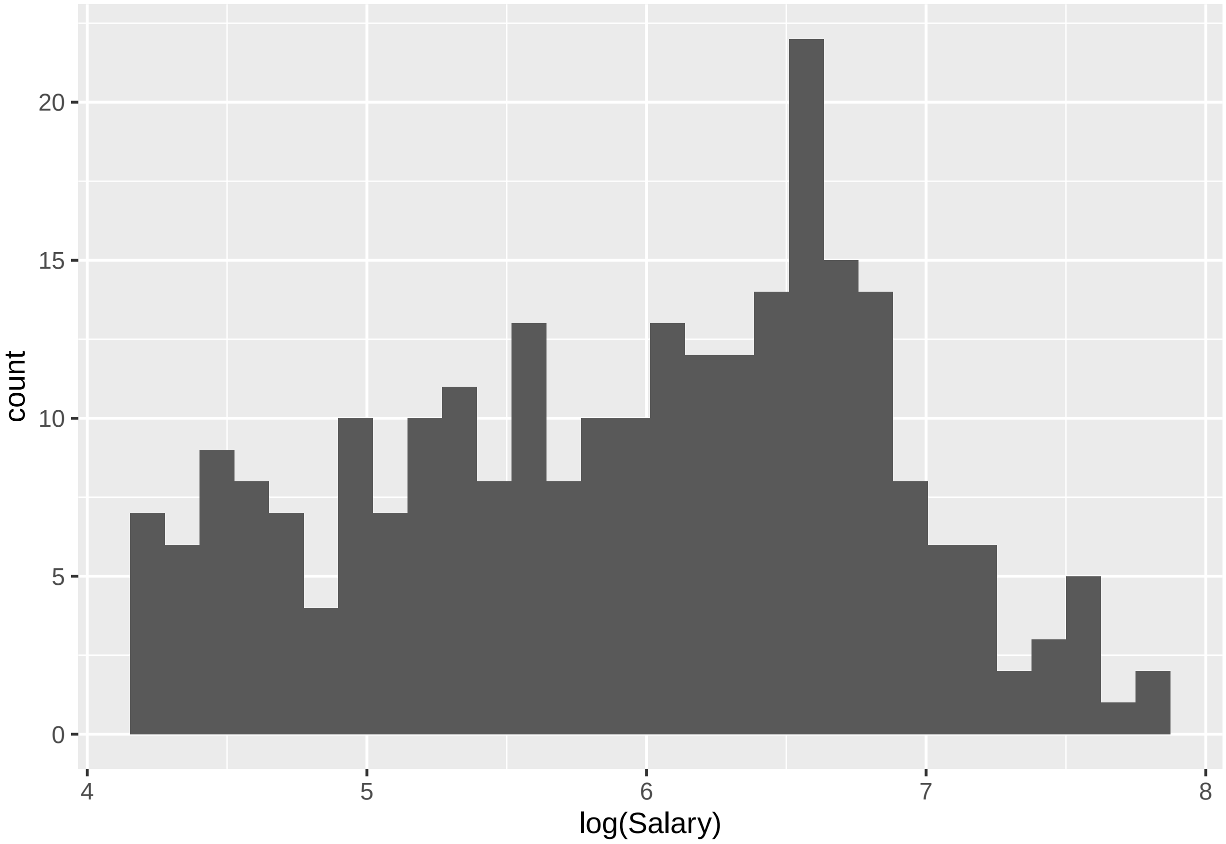
```
## [1] 1.570888
```

```
lm_log.model = lm(log(Salary) ~. , data = Hitters)
summary(lm_log.model)
```

```
##
## Call:
## lm(formula = log(Salary) ~ ., data = Hitters)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22870 -0.45350  0.09424  0.40474  2.77223
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.618e+00  1.765e-01  26.171  < 2e-16 ***
## AtBat       -2.984e-03  1.232e-03  -2.421  0.01620 *
## Hits         1.308e-02  4.622e-03   2.831  0.00503 **
## HmRun        1.179e-02  1.205e-02   0.978  0.32889
## Runs        -1.419e-03  5.794e-03  -0.245  0.80670
## RBI         -1.675e-03  5.056e-03  -0.331  0.74063
## Walks        1.096e-02  3.554e-03   3.082  0.00229 **
## Years        5.696e-02  2.413e-02   2.361  0.01902 *
## CAtBat       1.283e-04  2.629e-04   0.488  0.62596
## CHits       -4.414e-04  1.311e-03  -0.337  0.73670
## CHmRun      -7.809e-05  3.144e-03  -0.025  0.98020
## CRuns        1.513e-03  1.459e-03   1.037  0.30072
## CRBI         1.312e-04  1.346e-03   0.097  0.92246
## CWalks      -1.466e-03  6.377e-04  -2.298  0.02239 *
## LeagueN      2.825e-01  1.541e-01   1.833  0.06797 .
## DivisionW   -1.656e-01  7.847e-02  -2.111  0.03580 *
## PutOuts      3.389e-04  1.505e-04   2.251  0.02526 *
## Assists      6.214e-04  4.300e-04   1.445  0.14970
## Errors      -1.197e-02  8.537e-03  -1.402  0.16225
## NewLeagueN  -1.742e-01  1.536e-01  -1.134  0.25788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6135 on 243 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.5586, Adjusted R-squared:  0.524
## F-statistic: 16.18 on 19 and 243 DF,  p-value: < 2.2e-16
```

```
ggplot(data = Hitters.cleant, aes(log(Salary))) + geom_histogram()
```

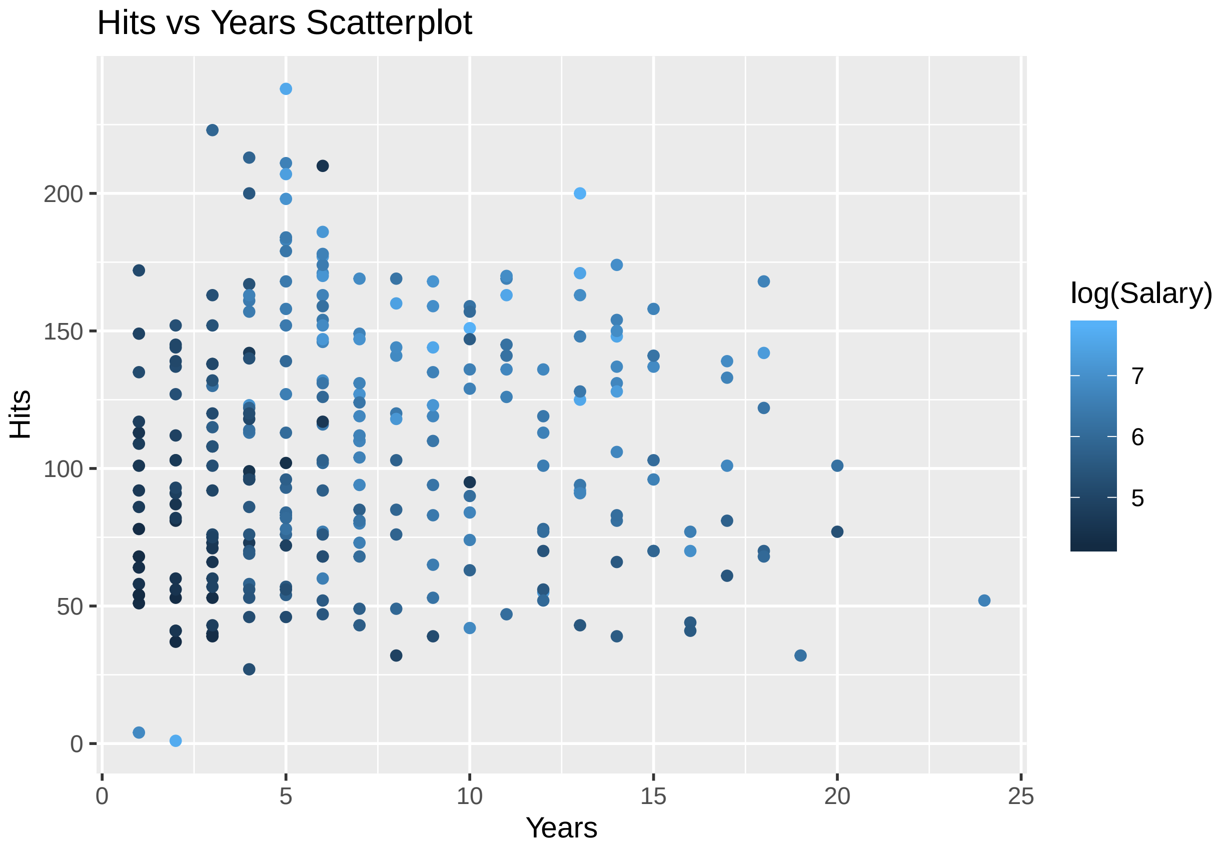
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



#*Interpretation 2: From skewness of the plot we can see that it is right skewed. It doesnt go along with the assumption that the data has to be normally distributed. Hence to overcome this problem we take log of salary. By taking log we can notice the transformation in the salary variable.

####*Question 3:Create a scatterplot with Hits on the y-axis and Years on the x-axis using all the observations. Color code the observations using the log Salary variable. What patterns do you notice on this chart, if any?

```
ggplot(data = Hitters.cleant, aes(x = Years, y = Hits, colour=log(Salary))) +
geom_point() +
     ggtitle("Hits vs Years Scatterplot")
```



#*Interpretation 3: From the scatterplot we can notice that as years increase the value of log salary variable increase. The darker regions are observed more in the intial distribution and as lighter regions in the later years. From the graph we can notice that the hits variable does not have much influence in the distribution across in the log(salary).

####*Question 4: Run a linear regression model of Log Salary on all the predictors using the entire dataset. Use regsubsets() function to perform best subset selection from the regression model. Identify the best model using BIC. Which predictor variables are included in this (best) model?

```
lm_log.model = lm(log(Salary) ~. , data = Hitters.cleant)
reg.search <- regsubsets(log(Salary) ~ ., data = Hitters.cleant, nbest = 1, nvmax = dim(Hitters.cleant)
method = "exhaustive")
sum <- summary(reg.search)
sum
```

```
## Subset selection object
## Call: regsubsets.formula(log(Salary) ~ ., data = Hitters.cleant, nbest = 1,
##     nvmax = dim(Hitters.cleant)[2], method = "exhaustive")
## 19 Variables  (and intercept)
##            Forced in Forced out
## AtBat          FALSE      FALSE
## Hits           FALSE      FALSE
## HmRun          FALSE      FALSE
## Runs           FALSE      FALSE
## RBI            FALSE      FALSE
## Walks          FALSE      FALSE
## Years          FALSE      FALSE
## CAtBat         FALSE      FALSE
## CHits          FALSE      FALSE
## CHmRun         FALSE      FALSE
## CRuns          FALSE      FALSE
## CRBI           FALSE      FALSE
## CWalks         FALSE      FALSE
## LeagueN        FALSE      FALSE
## DivisionW      FALSE      FALSE
## PutOuts        FALSE      FALSE
## Assists        FALSE      FALSE
## Errors         FALSE      FALSE
## NewLeagueN     FALSE      FALSE
## 1 subsets of each size up to 19
## Selection Algorithm: exhaustive
##           AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI
## 1  ( 1 )  " "   " "  " "   " "  " " " "   " "   " "    " "   " "    "*"   " "
## 2  ( 1 )  " "   "*"  " "   " "  " " " "   " "   "*"    " "   " "    " "   " "
## 3  ( 1 )  " "   "*"  " "   " "  " " "*"   "*"   " "    " "   " "    " "   " "
## 4  ( 1 )  "*"   "*"  " "   " "  " " "*"   " "   "*"    " "   " "    " "   " "
## 5  ( 1 )  " "   "*"  " "   " "  " " "*"   "*"   " "    "*"   " "    " "   " "
## 6  ( 1 )  "*"   "*"  " "   " "  " " "*"   "*"   " "    "*"   " "    " "   " "
## 7  ( 1 )  "*"   "*"  " "   " "  " " "*"   "*"   " "    " "   " "    "*"   " "
## 8  ( 1 )  "*"   "*"  " "   " "  " " "*"   "*"   " "    " "   " "    "*"   " "
## 9  ( 1 )  "*"   "*"  " "   " "  " " "*"   "*"   " "    " "   " "    "*"   " "
## 10  ( 1 ) "*"   "*"  " "   " "  " " "*"   "*"   " "    " "   " "    "*"   " "
## 11  ( 1 ) "*"   "*"  "*"   " "  " " "*"   "*"   " "    " "   " "    "*"   " "
## 12  ( 1 ) "*"   "*"  "*"   " "  " " "*"   "*"   " "    " "   " "    "*"   " "
## 13  ( 1 ) "*"   "*"  "*"   " "  " " "*"   "*"   " "    " "   " "    "*"   " "
## 14  ( 1 ) "*"   "*"  "*"   " "  " " "*"   "*"   "*"    " "   " "    "*"   " "
## 15  ( 1 ) "*"   "*"  "*"   " "  " " "*"   "*"   "*"    "*"   " "    "*"   " "
## 16  ( 1 ) "*"   "*"  "*"   " "  "*" "*"   "*"   "*"    "*"   " "    "*"   " "
## 17  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   "*"   "*"    "*"   " "    "*"   " "
## 18  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   "*"   "*"    "*"   " "    "*"   "*"
## 19  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   "*"   "*"    "*"   "*"    "*"   "*"
##           CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1  ( 1 )  " "    " "     " "       " "     " "     " "    " "
## 2  ( 1 )  " "    " "     " "       " "     " "     " "    " "
## 3  ( 1 )  " "    " "     " "       " "     " "     " "    " "
## 4  ( 1 )  " "    " "     " "       " "     " "     " "    " "
## 5  ( 1 )  " "    " "     "*"       " "     " "     " "    " "
## 6  ( 1 )  " "    " "     "*"       " "     " "     " "    " "
## 7  ( 1 )  "*"    " "     " "       "*"     " "     " "    " "
## 8  ( 1 )  "*"    " "     "*"       "*"     " "     " "    " "
## 9  ( 1 )  "*"    "*"     "*"       "*"     " "     " "    " "
## 10  ( 1 ) "*"    "*"     "*"       "*"     " "     " "    "*"
## 11  ( 1 ) "*"    "*"     "*"       "*"     " "     " "    "*"
## 12  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    " "
## 13  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
## 14  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
## 15  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
## 16  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
## 17  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
## 18  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
## 19  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
```

```
which.max(sum$adjr2)
```

```
## [1] 13
```

```
sum$cp
```

```
##  [1] 79.124523 27.981036 21.001033 17.276086 13.740484 11.343922  8.660386
##  [8]  6.185015  6.147901  6.624133  7.344849  7.822749  8.461825 10.352605
## [15] 12.177361 14.121460 16.042709 18.000617 20.000000
```

```
sub_lm_log.model <- lm(log(Salary) ~ AtBat+Hits+Walks+Years+CRuns+CWalks+PutOuts, data = Hitters.cleant
BIC(lm_log.model)
```

```
## [1] 585.5431
```

```
BIC(sub_lm_log.model)
```

```
## [1] 532.0347
```

#*Interpretation 4: The best model from BIC is linear regression obtained through exhaustive search because on comparing the BIC values between subset selection(BIC = 532.03) and linear regression(BIC = 585.54), we can see that the BIC value is low for exhaustive search. At beginning we run linear regression using all variables, then we run the subset selection to find best subset. Through adj R2 we see that subset 13 has highest value. But by using Cp value we find out that for susbset 7 we have reduced the number of predictors and the model seems to best.Also the 7 predictor variables included from the best model are CRuns, Hits, Cwalks, Walks, Putouts, Years, AtBat.(Note : Based on the Cp value we decide to go with 7 predictors from the exhaustive search as 7th Cp = 8.6 which is closer to p+1).

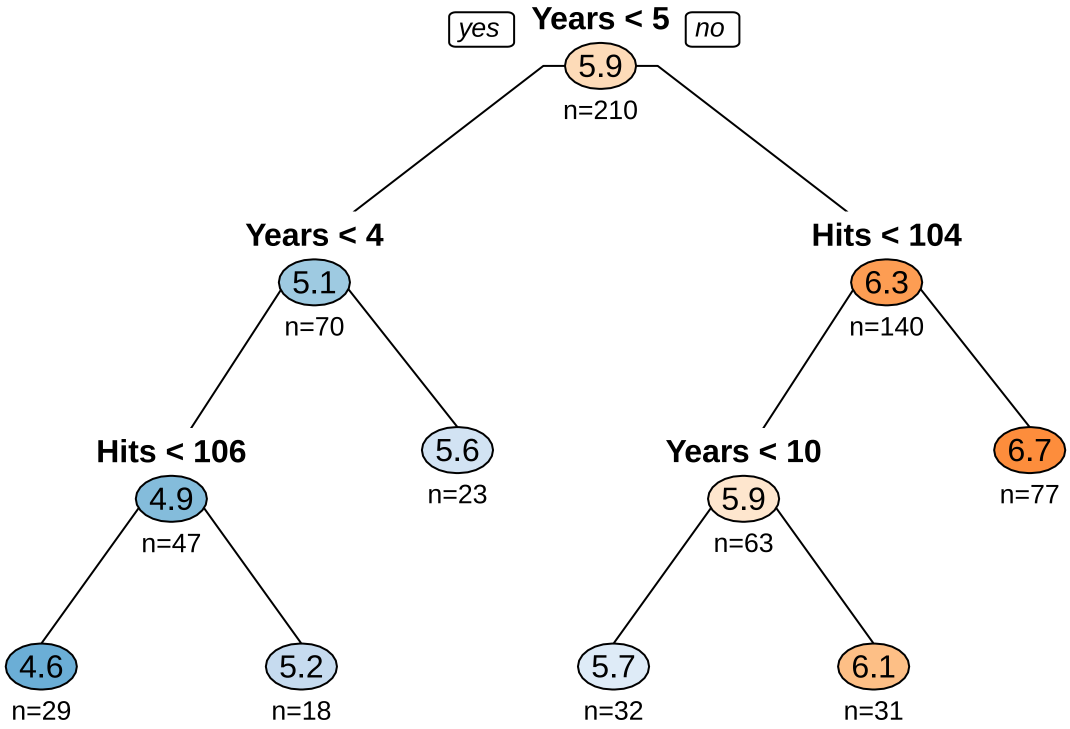
####*Question 5: Now create a training data set consisting of 80 percent of the observations, and a test data set consisting of the remaining observations.

```
train.index <- sample(row.names(Hitters.cleant), 0.8*dim(Hitters.cleant)[1])

valid.index <- setdiff(row.names(Hitters.cleant), train.index)

train.df <- Hitters.cleant[train.index, c("Salary", "Hits", "Walks", "Years", "CRuns", "AtBat", "CWalks
valid.df <- Hitters.cleant[valid.index, c("Salary", "Hits", "Walks", "Years", "CRuns", "AtBat", "CWalks
```

#*Interpretation 5: The training set and testing set has been created as required

####*Question 6:Generate a regression tree of log Salary using only Years and Hits variables from the training data set. Which players are likely to receive highest salaries according to this model? Write down the rule and elaborate on it.

```
tree.model <- rpart(log(Salary) ~ Years+Hits, data = train.df,
method = "anova")
prp(tree.model, type = 1, extra = 1, under = TRUE, split.font = 2,
varlen = -10, box.palette = "BuOr")
```



```
rpart.rules(tree.model, cover = TRUE)
```

```
##  log(Salary)                                            cover
##          4.6 when Years <  4        & Hits <  106     14%
##          5.2 when Years <  4        & Hits >= 106      9%
##          5.6 when Years is 4 to  5                    11%
##          5.7 when Years is 5 to 10 & Hits <  104     15%
##          6.1 when Years >=      10 & Hits <  104     15%
##          6.7 when Years >=       5 & Hits >= 104     37%
```

#*Interpretation 6: We get to know that highest log salary value is 6.7. A player who is playing for for more than 5 years and has more than 104 hits are predicted to recieve the highest salary as per the tree. The salary value is apprpoximated to be around 832.

####*Question 7:Now create a regression tree using all the variables in the training data set. Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter . Produce a plot with different shrinkage values on the xaxis and the corresponding training set MSE on the y-axis.

```
reg.tree <- tree(log(Salary) ~ ., data = train.df)
summary(reg.tree)
```

```
##
## Regression tree:
## tree(formula = log(Salary) ~ ., data = train.df)
## Variables actually used in tree construction:
## [1] "CRuns"   "AtBat"   "Hits"    "CWalks"  "PutOuts"
## Number of terminal nodes:  9
## Residual mean deviance:  0.1676 = 33.69 / 201
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.918000 -0.189000 -0.002313  0.000000  0.189300  1.485000
```

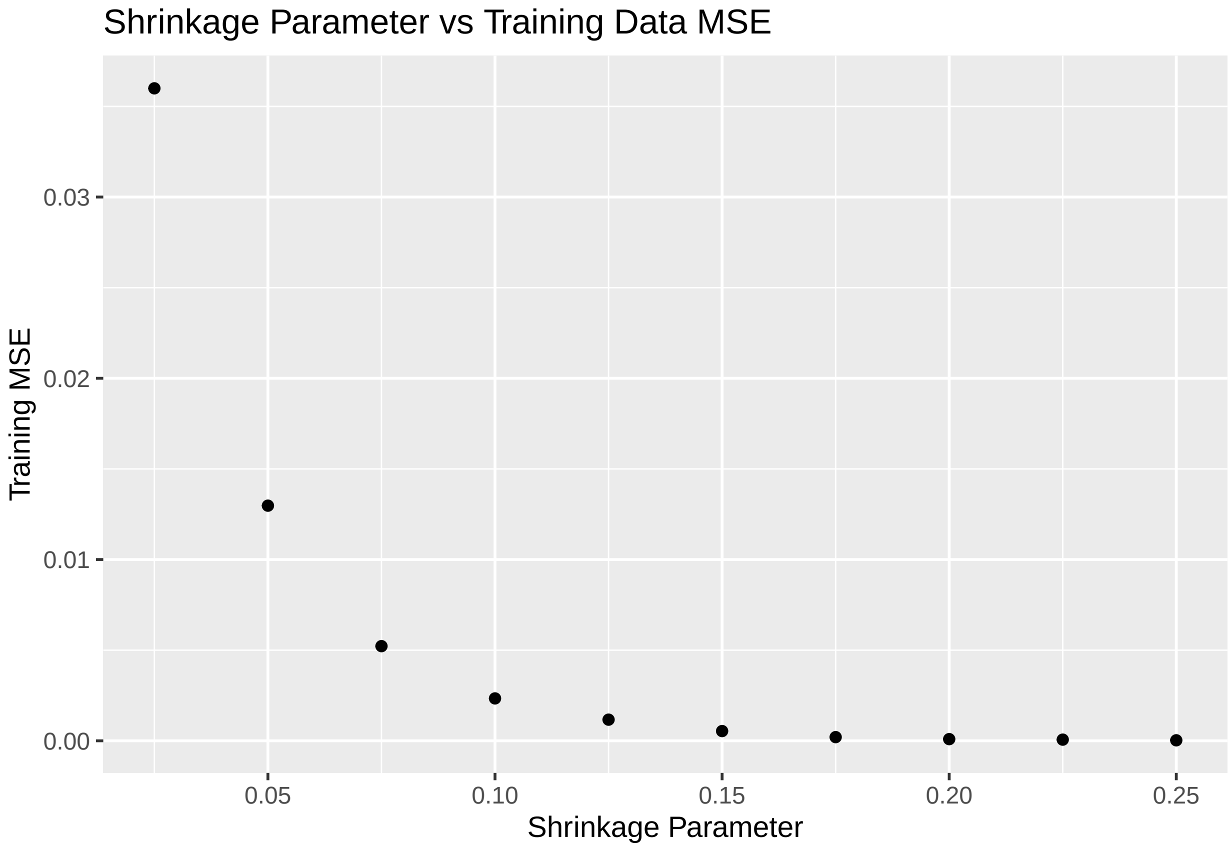
```
sh <- seq(0.025,0.25,0.025)
boost.df <- data.frame(iter= seq(1, 10, 1), shrink = seq(1, 10, 1), val = rep(0, 10), testmse=rep(0, 10
j <- 1
for (i in sh) {
boost <- gbm(log(Salary)~., data=train.df, distribution = "gaussian",
n.trees=1000, interaction.depth = 4, shrinkage = i)
yhat.boost <- predict(boost, newdata=train.df, n.trees=1000)
boost.df[j,2] <- i
boost.df[j,3] <- mean((yhat.boost-log(train.df[,"Salary"]))^2)
yhat.boost <- predict(boost, newdata=train.df, n.trees=1000)
boost.df[j,4] <- mean((yhat.boost- log(valid.df[,"Salary"]))^2)
j <- j+1
}
```

```
## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length

## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length

## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length

## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length

## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length

## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length

## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length
## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length

## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length

## Warning in yhat.boost - log(valid.df[, "Salary"]): longer object length is not a
## multiple of shorter object length
```

```
boost.df
```

```
##    iter shrink          val  testmse
## 1     1  0.025 3.599763e-02 1.592268
## 2     2  0.050 1.297236e-02 1.642595
## 3     3  0.075 5.221669e-03 1.673053
## 4     4  0.100 2.338869e-03 1.690832
## 5     5  0.125 1.166193e-03 1.696098
## 6     6  0.150 5.366310e-04 1.700628
## 7     7  0.175 2.019801e-04 1.706953
## 8     8  0.200 9.258330e-05 1.709014
## 9     9  0.225 6.430904e-05 1.708871
## 10   10  0.250 2.735621e-05 1.710124
```

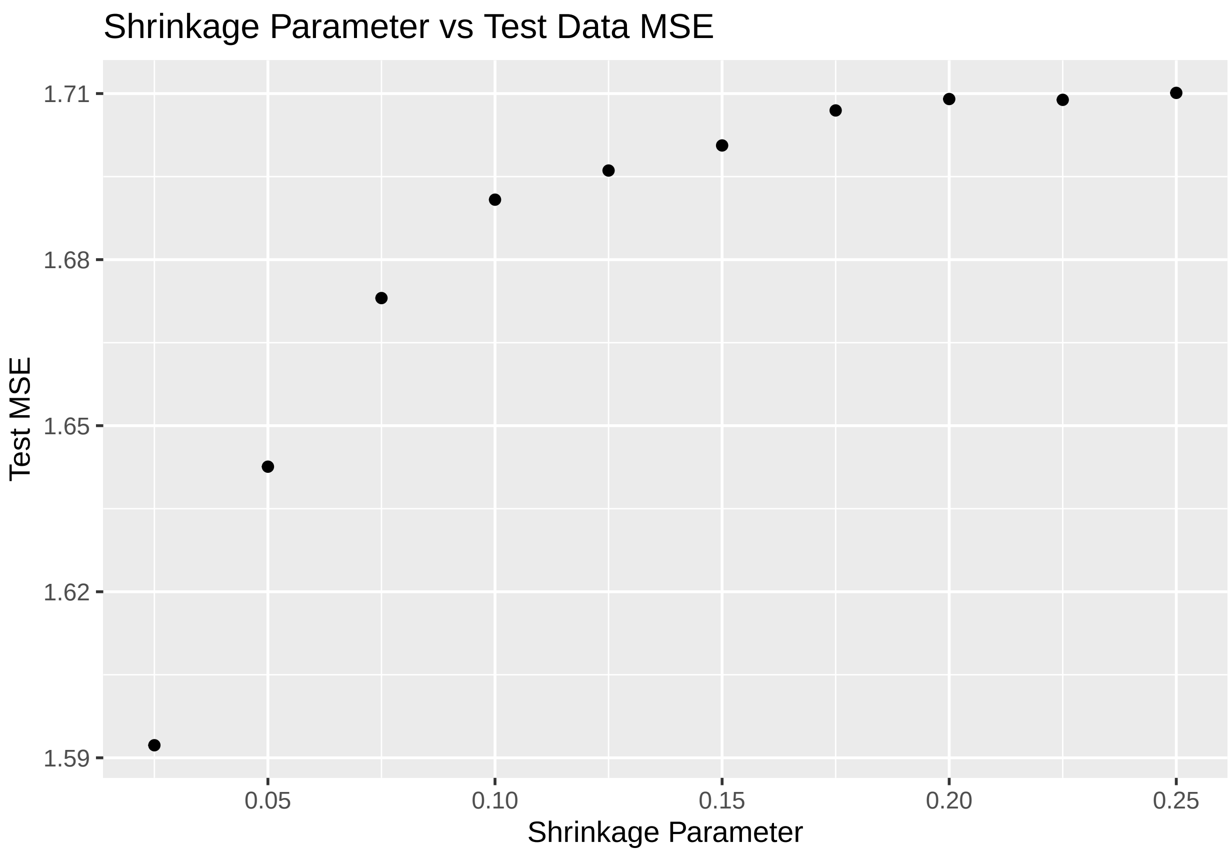
```
ggplot(boost.df)+
geom_point(aes(x=shrink, y=val))+
xlab("Shrinkage Parameter")+
ylab("Training MSE")+
ggtitle("Shrinkage Parameter vs Training Data MSE")
```



# *Interpretation 7: Boosting on training set with 1000 trees has been performed and plot has also been produced.

####*Question 8:Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.

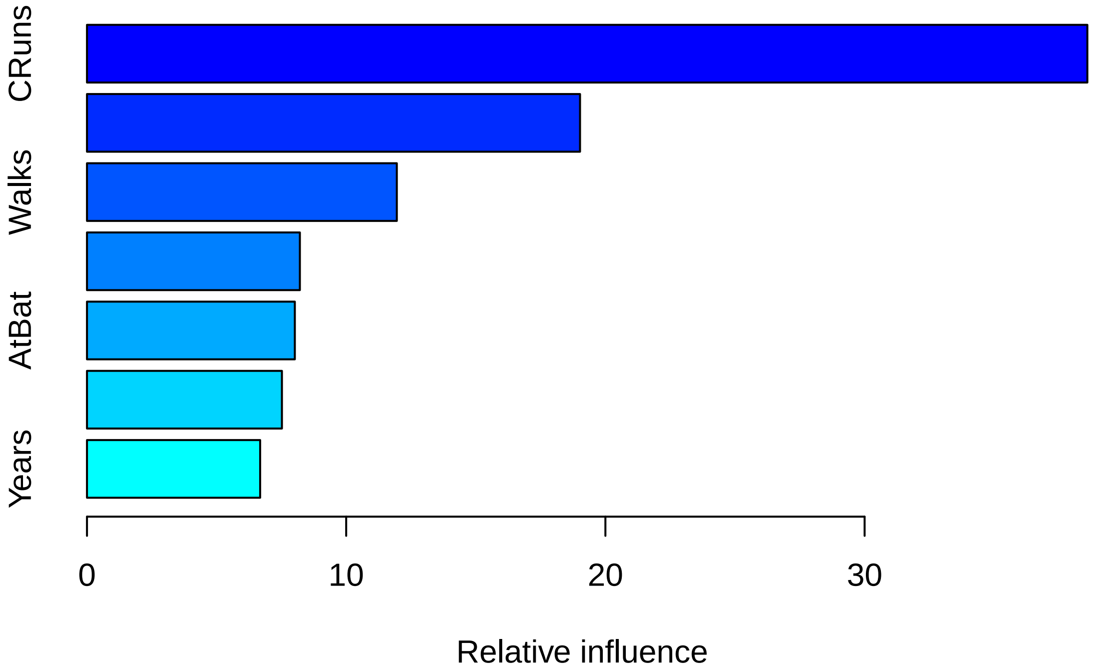
```r
ggplot(boost.df)+
geom_point(aes(x=shrink, y=testmse))+
xlab("Shrinkage Parameter")+
ylab("Test MSE")+
ggtitle("Shrinkage Parameter vs Test Data MSE")
```



***Interpretation 8: We can see that shrinkage parameter increases, MSE increases too. The optimal point at which the training MSE and test MSE are minimal is 0.075 hence we assume a shrinkage of 0.075 and run the model**

####*Question 9:Which variables appear to be the most important predictors in the boosted model?

```
boost <- gbm(log(Salary)~., data=train.df, distribution = "gaussian", n.trees=1000, interaction.depth =


summary(boost)
```

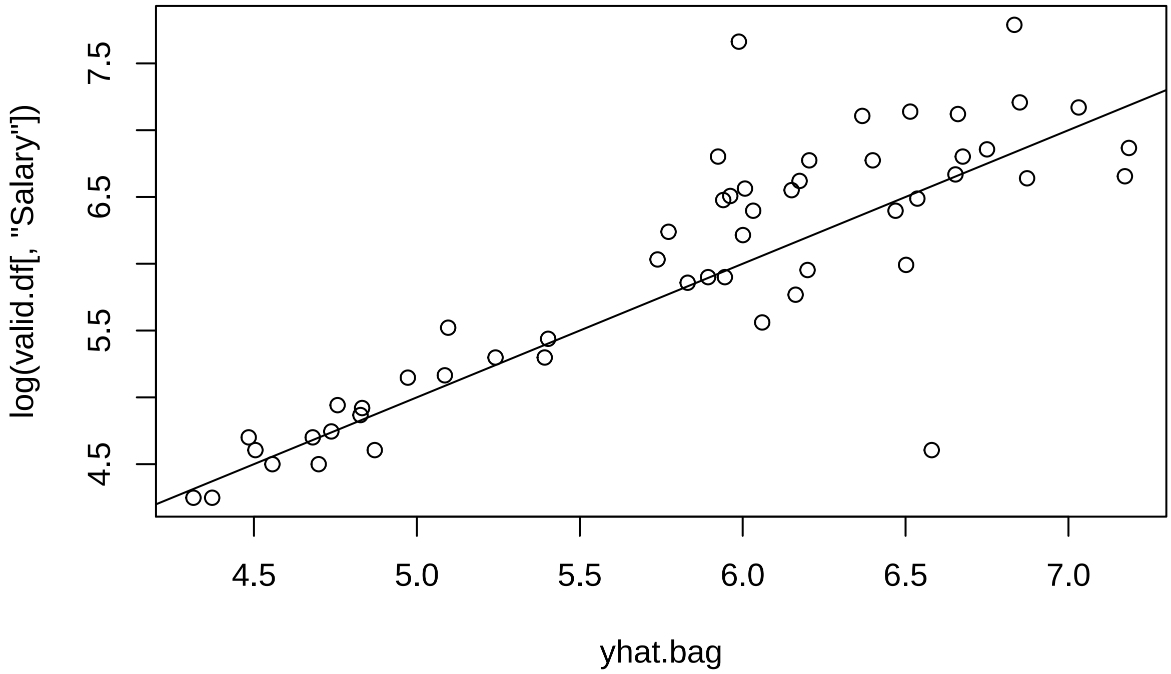


```
##             var   rel.inf
## CRuns     CRuns 38.592587
## PutOuts PutOuts 19.020862
## Walks     Walks 11.952988
## CWalks   CWalks  8.213386
## AtBat     AtBat  8.018606
## Hits       Hits  7.520521
## Years     Years  6.681051
```

#*Interpretation 9 - We are able to see the 7 predictor variables used in the model. CRuns seems to be the most important predictor variable which can be seen from the graph and aslo from the summary.

####*Question 10:Now apply bagging to the training set. What is the test set MSE for this approach?

```
#Bagging
hit.bag <- randomForest(log(Salary)~., data=train.df,
mtry = 3, importance = TRUE)

yhat.bag <- predict(hit.bag, valid.df)
plot(yhat.bag, log(valid.df[,"Salary"]))
abline(0,1)
```



```
mean((yhat.bag- (log(valid.df[,"Salary"])) ) ^2)
```

```
## [1] 0.2554071
```

#*Interpretation 10 - MSE is 0.25. Assumed number of variables to be 3 as we have have 7 variables and assuming root of 7 to be approximately 3