BUAN6356_Homewrok4_Group11

Jeffrey Chang,Naishal Kanubhai Thakkar,Bhavana Chowdary Kandimalla,Navya Bulusu,Rucha Nichkawade

11/14/2019

```
if(!require("pacman")) install.packages("pacman")
pacman::p_load(ISLR, MASS, rpart, rpart.plot, caret,
               randomForest, gbm, tree, ggplot2, dplyr, caTools, leaps, stats, moments )
```

Hitters data set contains information on Major League Baseball from the 1986 and 1987 seasons. Among other information, it contains 1987 annual salary of baseball players (in thousands of dollars) on opening day of the season. Our goal is the predict salaries of the players. This data set is available from the ISLR package.

Question 1:

Remove the observations with unknown salary information. How many observations were removed in this process?

```
#head(Hitters)
hitters.na.out <- Hitters[!is.na(Hitters$Salary),]
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 ...
```

```
str(hitters.na.out)
```

```
## 'data.frame':    263 obs. of  20 variables:
##  $ AtBat    : int  315 479 496 321 594 185 298 323 401 574 ...
##  $ Hits     : int  81 130 141 87 169 37 73 81 92 159 ...
##  $ HmRun    : int  7 18 20 10 4 1 0 6 17 21 ...
##  $ Runs     : int  24 66 65 39 74 23 24 26 49 107 ...
##  $ RBI      : int  38 72 78 42 51 8 24 32 66 75 ...
##  $ Walks    : int  39 76 37 30 35 21 7 8 65 59 ...
```

```
##  $ Years    : int  14 3 11 2 11 2 3 2 13 10 ...
##  $ CAtBat   : int  3449 1624 5628 396 4408 214 509 341 5206 4631 ...
##  $ CHits    : int  835 457 1575 101 1133 42 108 86 1332 1300 ...
##  $ CHmRun   : int  69 63 225 12 19 1 0 6 253 90 ...
##  $ CRuns    : int  321 224 828 48 501 30 41 32 784 702 ...
##  $ CRBI     : int  414 266 838 46 336 9 37 34 890 504 ...
##  $ CWalks   : int  375 263 354 33 194 24 12 8 866 488 ...
##  $ League   : Factor w/ 2 levels "A","N": 2 1 2 2 1 2 1 2 1 1 ...
##  $ Division : Factor w/ 2 levels "E","W": 2 2 1 1 2 1 2 2 1 1 ...
##  $ PutOuts  : int  632 880 200 805 282 76 121 143 0 238 ...
##  $ Assists  : int  43 82 11 40 421 127 283 290 0 445 ...
##  $ Errors   : int  10 14 3 4 25 7 9 19 0 22 ...
##  $ Salary   : num  475 480 500 91.5 750 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 2 1 2 2 1 1 1 2 1 1 ...
```

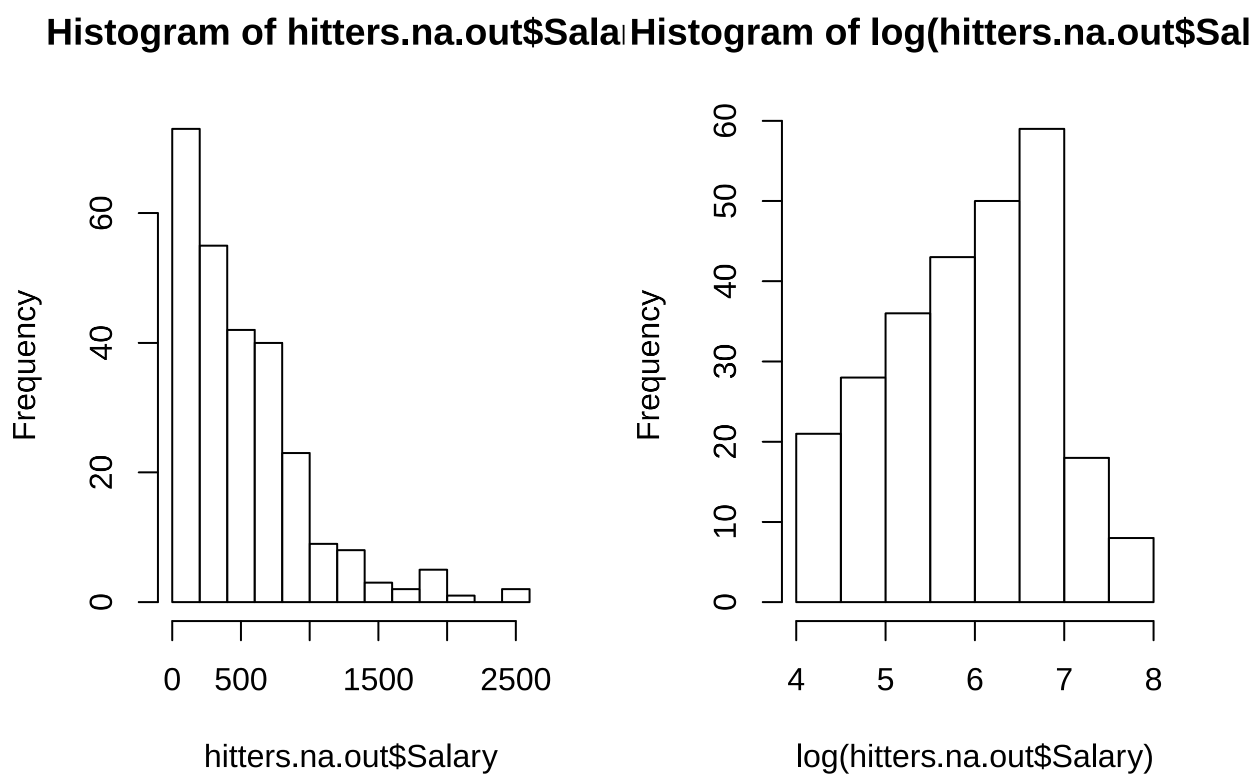
Explanation:

The original data have 322 observations. After we took off the row which have NA value, the observations are 263. 322-263=59, so 59 observations were removed in this process.

Question 2:

Generate log-transform the salaries. Can you justify this transformation?

```r
#library('dplyr'), library('moments')
par(mfrow = c(1,2))
hist(hitters.na.out$Salary)
hitters_new<-hitters.na.out%>%mutate(log_salary=log(Salary))
log_transformation<-hist(log(hitters.na.out$Salary))
```




```r
skewness(hitters_new$log_salary)
```

```
## [1] -0.1809668
```

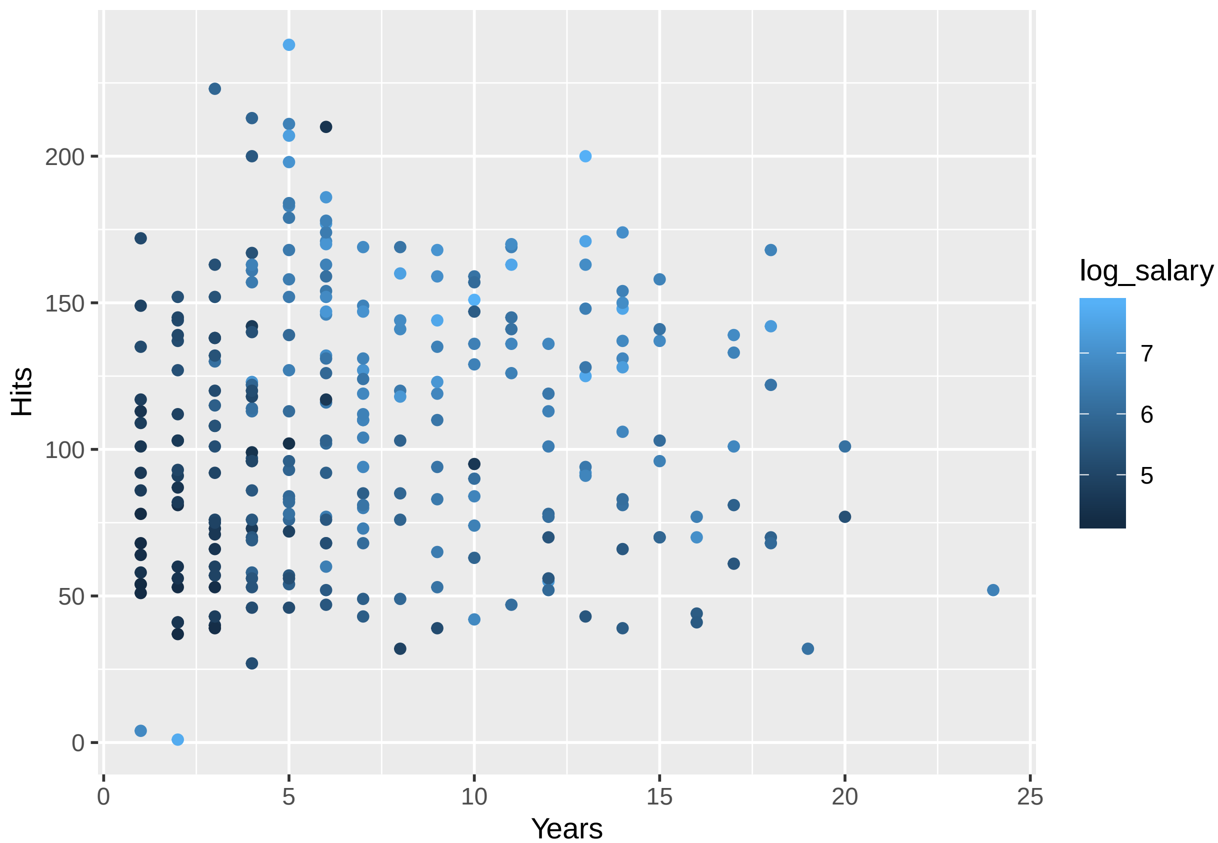
Explanation:

The log transformation can be used to make highly skewed distributions less skewed. This can be valuable both for making patterns in the data more interpretable and for helping to meet the assumptions of inferential statistics. Above plots shows an example of how a log transformation can make patterns more visible. The log transformation makes the distribution approximately normal.

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?

```
# library('ggplot2'), library('dplyr')
hitters.na.out<-hitters.na.out%>%mutate(log_salary=log(Salary))
hitters.na.out<-hitters.na.out[,-c(19)]
ggplot(hitters.na.out,aes(y=Hits,x=Years,col=log_salary))+geom_point()
```



Explanation:

The low salary is distributed between 0-5 years, and as the years increases the salary also tend to increase. Hence, the years and the number of hits have positive influence on the salary. There are some exceptions where salary is higher at less number of years and hits.

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?

```
#library(leaps), library(stats)
models <- regsubsets(log_salary~., data = hitters.na.out, nvmax = 19)
res.sum <- summary(models)
res.sum$bic
```

```
##  [1] -117.0304 -156.4291 -159.2777 -159.2182 -159.0885 -157.9207 -157.1229
##  [8] -156.1954 -152.7649 -148.8061 -144.5962 -140.6541 -136.5480 -131.0939
## [15] -125.7112 -120.1995 -114.7125 -109.1859 -103.6145
```

```
num<-data.frame(BIC = which.min(res.sum$bic))
num #results is 3 -159.2777
```

```
##   BIC
## 1   3
```

```
res.sum$which[1:5,]
```

```
##   (Intercept) AtBat  Hits HmRun  Runs   RBI Walks Years CAtBat CHits
## 1        TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  FALSE FALSE
## 2        TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE   TRUE FALSE
## 3        TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  FALSE FALSE
## 4        TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE   TRUE FALSE
## 5        TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  FALSE  TRUE
##   CHmRun CRuns  CRBI CWalks LeagueN DivisionW PutOuts Assists Errors
## 1  FALSE  TRUE FALSE  FALSE   FALSE     FALSE   FALSE   FALSE  FALSE
## 2  FALSE FALSE FALSE  FALSE   FALSE     FALSE   FALSE   FALSE  FALSE
## 3  FALSE FALSE FALSE  FALSE   FALSE     FALSE   FALSE   FALSE  FALSE
## 4  FALSE FALSE FALSE  FALSE   FALSE     FALSE   FALSE   FALSE  FALSE
## 5  FALSE FALSE FALSE  FALSE   FALSE      TRUE   FALSE   FALSE  FALSE
##   NewLeagueN
## 1      FALSE
## 2      FALSE
## 3      FALSE
## 4      FALSE
## 5      FALSE
```

Explanation:

After running the regression model, we found that the best model is the one with 3 predictors as these predictors have the lowest BIC values: -159.2777.
The predictors are namely Hits, Walks and Years.

Question 5:

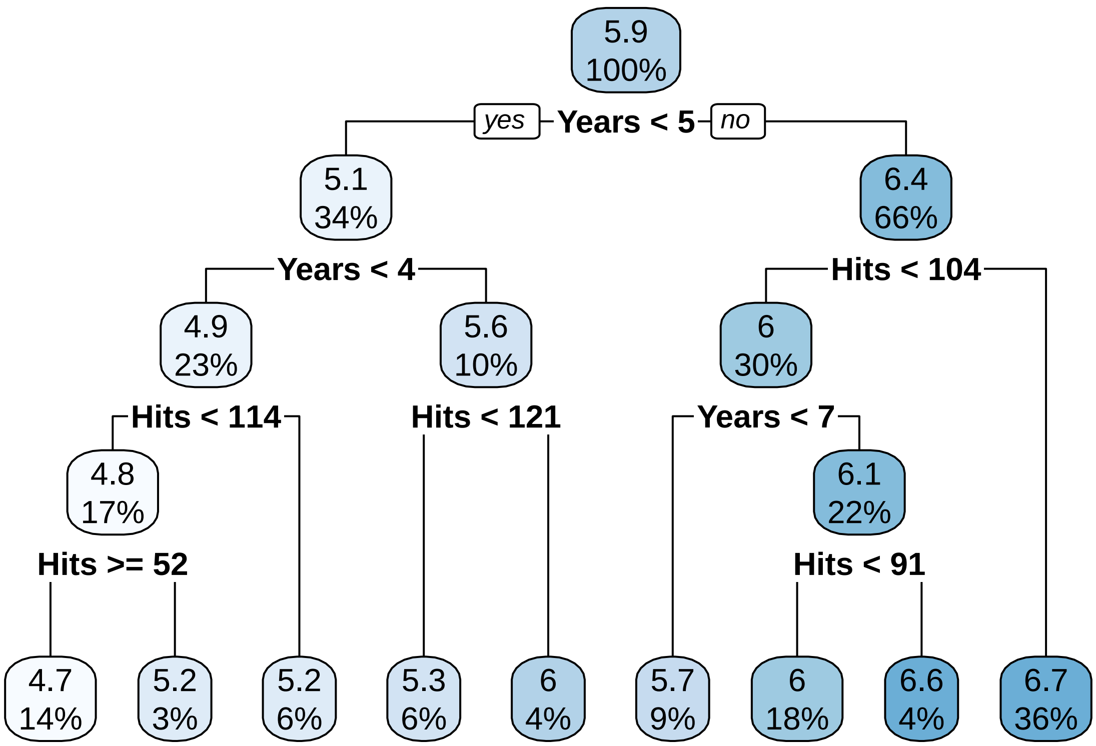
Now create a training data set consisting of 80 percent of the observations, and a test data set consisting of the remaining observations.

```
#library(caTools)
set.seed(42)
sample = sample.split(hitters.na.out$log_salary, SplitRatio = .80)
train.df= subset(hitters.na.out, sample == TRUE)
valid.df= subset(hitters.na.out, sample == FALSE)
```

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.

```
#library(MASS),library(rpart),library(rpart.plot)
tree.baseball <- rpart(log_salary ~ Hits + Years,data =train.df)
#summary(tree.baseball)

rpart.plot(tree.baseball)
```




```
tree.baseball$variable.importance
```

```
##    Years     Hits
## 90.12296 28.46303
```

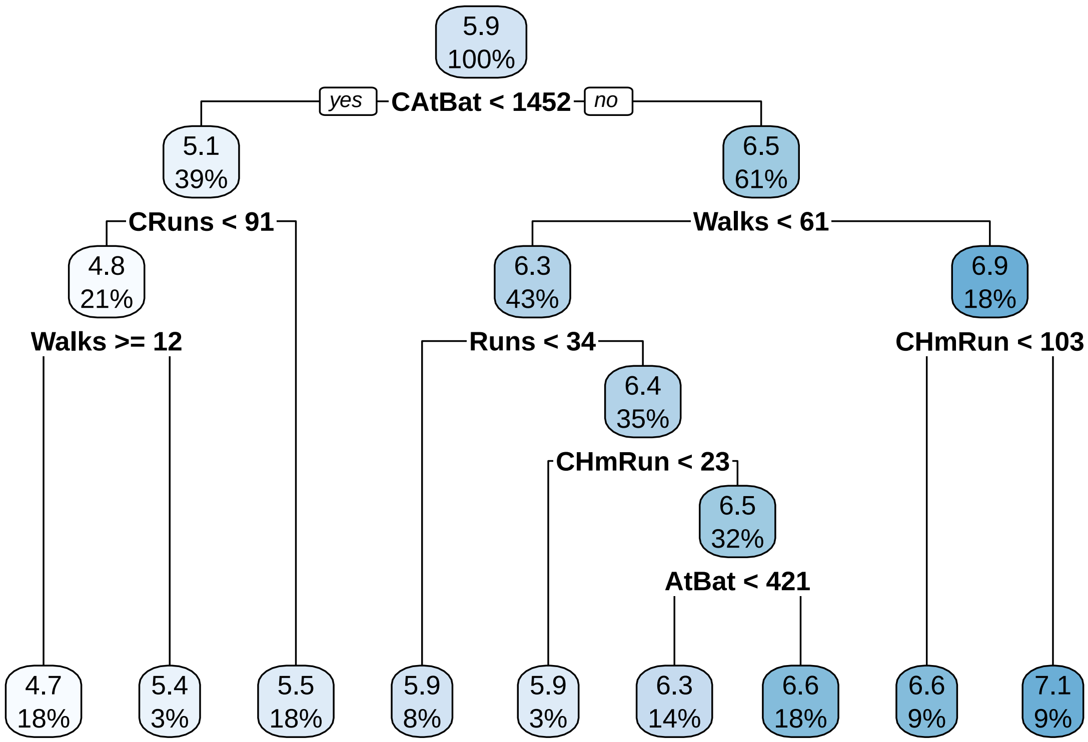
Explanation:

Rule : IF(Years >= 5) and (Hits >= 104) THEN Salary = 6.7
If the Years is more than 5 years and hits more than 104 Hits, they will receive a salary of 6.7 which is the highest.

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 lambdas. Produce a plot with different shrinkage values on the x-axis and the corresponding training set MSE on the y-axis.

```
#Regression Tree
tree.baseball_1<-rpart(log_salary~.,data=train.df)
#summary(tree.baseball_1)
rpart.plot(tree.baseball_1)
```

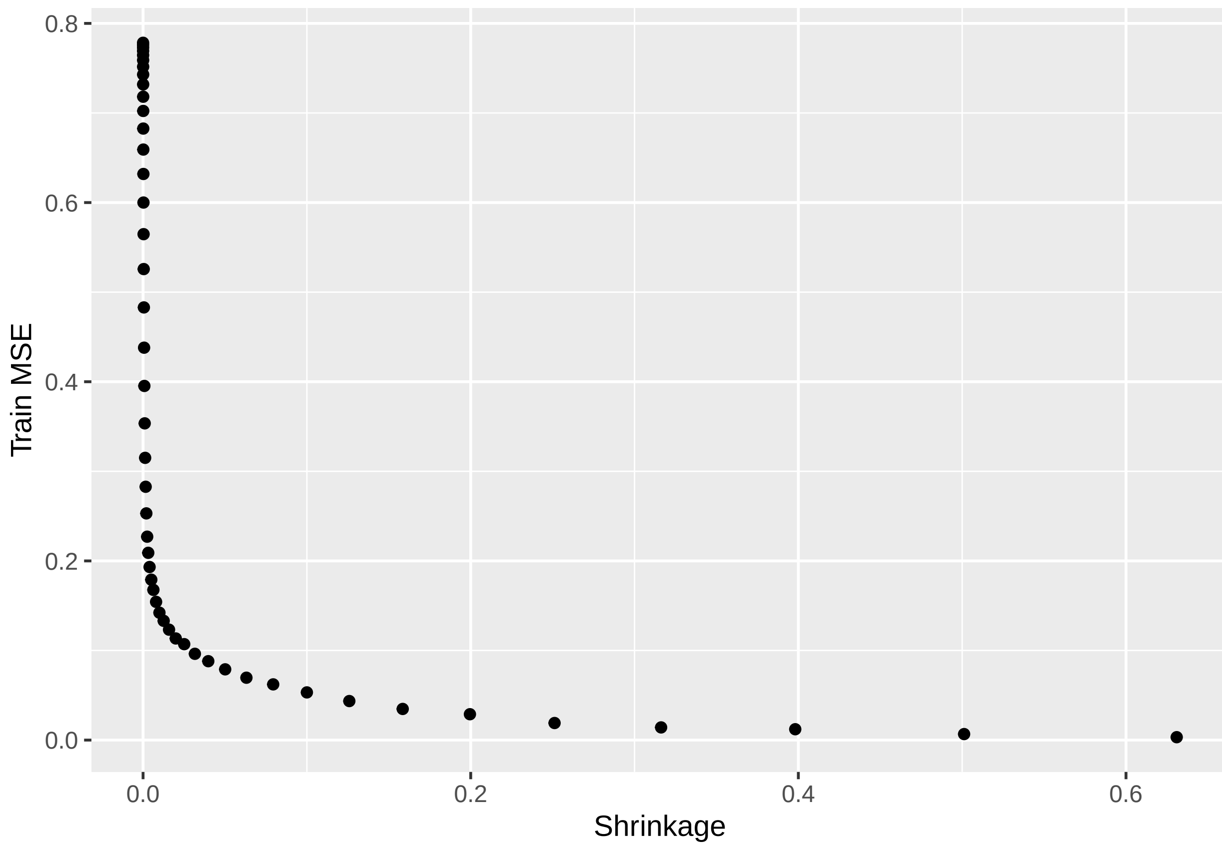



```
tree.baseball_1$variable.importance
```

```
##     CAtBat      CRuns      CHits     CWalks       CRBI      Years
## 108.924555 103.480074 103.376594  94.930015  92.385335  73.080618
##      Walks       Runs       Hits        RBI      AtBat     CHmRun
##  12.542353   9.101631   5.580050   5.537222   5.438644   4.924160
##      HmRun    PutOuts
##   4.564161   1.775615
```

```
#lambdas <- c(c(), seq(0.002, 0.01, by=0.001))
#lambdas <- c(lambdas, seq(0.02, 0.1, by=0.01))
#lambdas <- c(lambdas, seq(0.2, 1, by=0.1))
set.seed(42)
p = seq(-5, -0.2, by = 0.1)
lambdas = 10^p
length.lambdas <- length(lambdas)
train_errors <- rep(NA, length.lambdas)
```

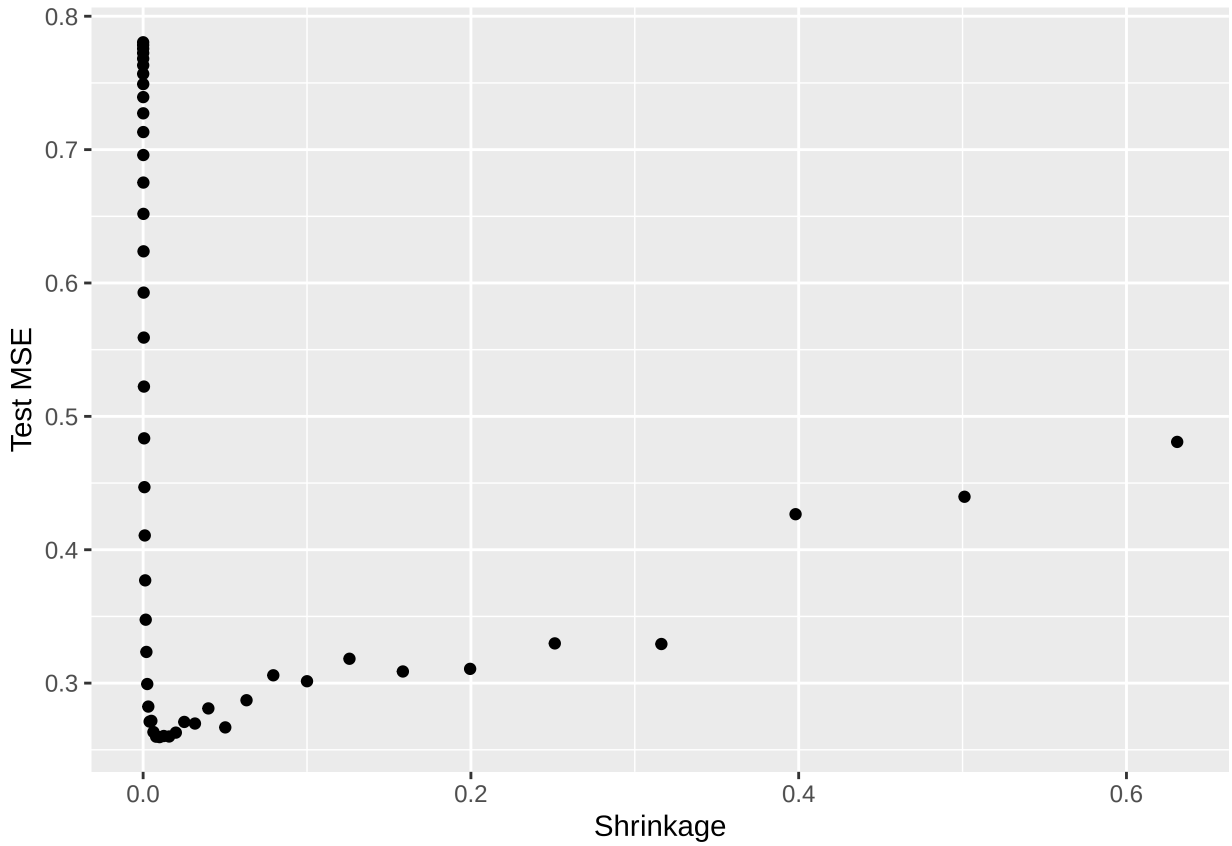
```
test_errors <- rep(NA, length.lambdas)

for (i in 1:length.lambdas) {
    boost.hitters <- gbm(log_salary~ ., data = train.df, distribution = "gaussian",
        n.trees = 1000, shrinkage = lambdas[i])

    train_pred <- predict(boost.hitters, train.df, n.trees = 1000)
    test_pred <- predict(boost.hitters, valid.df, n.trees = 1000)

    train_errors[i] = mean((train.df$log_salary - train_pred)^2)
    test_errors[i] = mean((valid.df$log_salary - test_pred)^2)
}

ggplot(data.frame(x=lambdas, y=train_errors), aes(x=x, y=y)) + xlab("Shrinkage") + ylab("Train MSE") + g
```



Question 8:

Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.

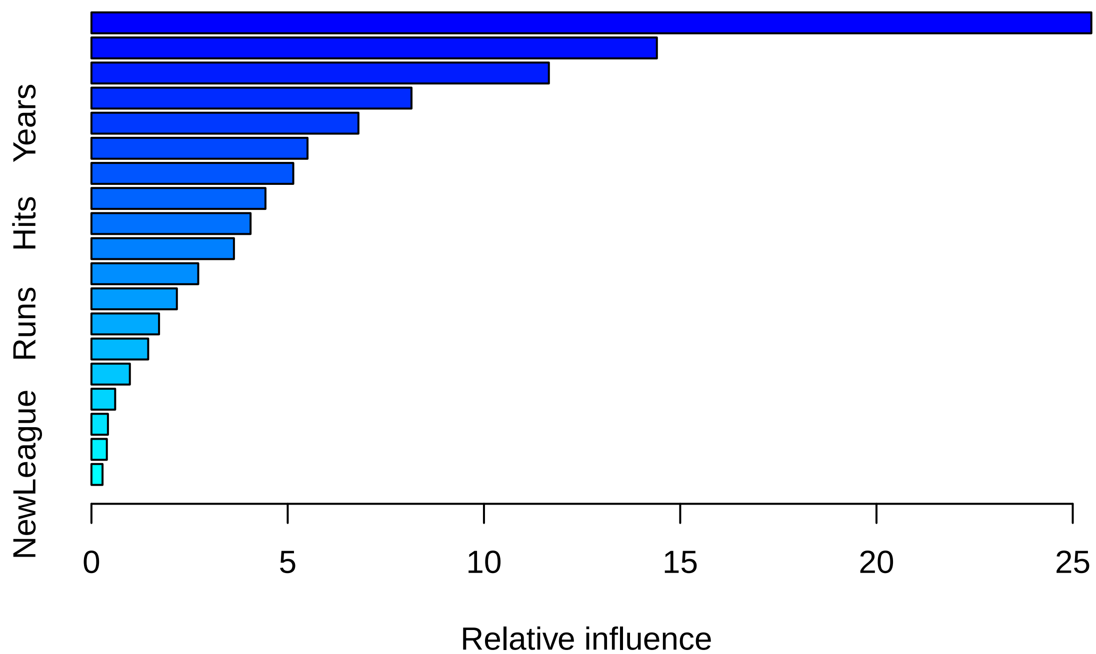
```
ggplot(data.frame(x=lambdas, y=test_errors), aes(x=x, y=y)) + xlab("Shrinkage") + ylab("Test MSE") + ge
```



Question 9:

Which variables appear to be the most important predictors in the boosted model?

```
boost_best_model <- gbm(log_salary ~ ., data = train.df, distribution = "gaussian",
    n.trees = 1000, shrinkage = lambdas[which.min(test_errors)])
head(summary(boost_best_model))
```




```
##           var   rel.inf
## CAtBat CAtBat 25.474170
## CRBI     CRBI 14.405370
## CHits   CHits 11.654105
## CRuns   CRuns  8.154037
## Years   Years  6.801863
## Walks   Walks  5.504852
```

Explanation:

As the result, the most important predictors in the boosted model is "CAtBat".

Question 10:

Now apply bagging to the training set. What is the test set MSE for this approach?

```
set.seed(42)
lm_Hitter <- lm(log_salary ~ ., data = train.df)
lm_pred <- predict(lm_Hitter, valid.df)
mean((valid.df$log_salary - lm_pred)^2)
```

```
## [1] 0.3814122
```

```
bagging <- randomForest(log_salary~., data = train.df, importance = T, mtry = 19)
bagging_pred <- predict(bagging, valid.df)
mean((bagging_pred-valid.df$log_salary)^2)
```

```
## [1] 0.2528047
```

Explanation:

Before doing the bagging, the MSE is 0.3814122 and after bagging the MSE is lower to 0.2528047.