R Notebook

Code

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Hide

```
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

HW-4 ECG-590 ……………………………………………………………………………………………………..

Moving Beyond Linear Models ……………………………………………………………………………………………..

1. It was mentioned in the chapter that a cubic regression spline with one knot at ?? can be obtained using a basis of the form x, x2, x3, (x ??? ??)3 +, where (x ??? ??)3

- = (x ??? ??)3 if x > ?? and equals 0 otherwise. We will now show that a function of the form f(x) = ??0 + ??1x + ??2x2 + ??3x3 + ??4(x ??? ??)3
- is indeed a cubic regression spline, regardless of the values of ??0, ??1, ??2, ??3, ??4.

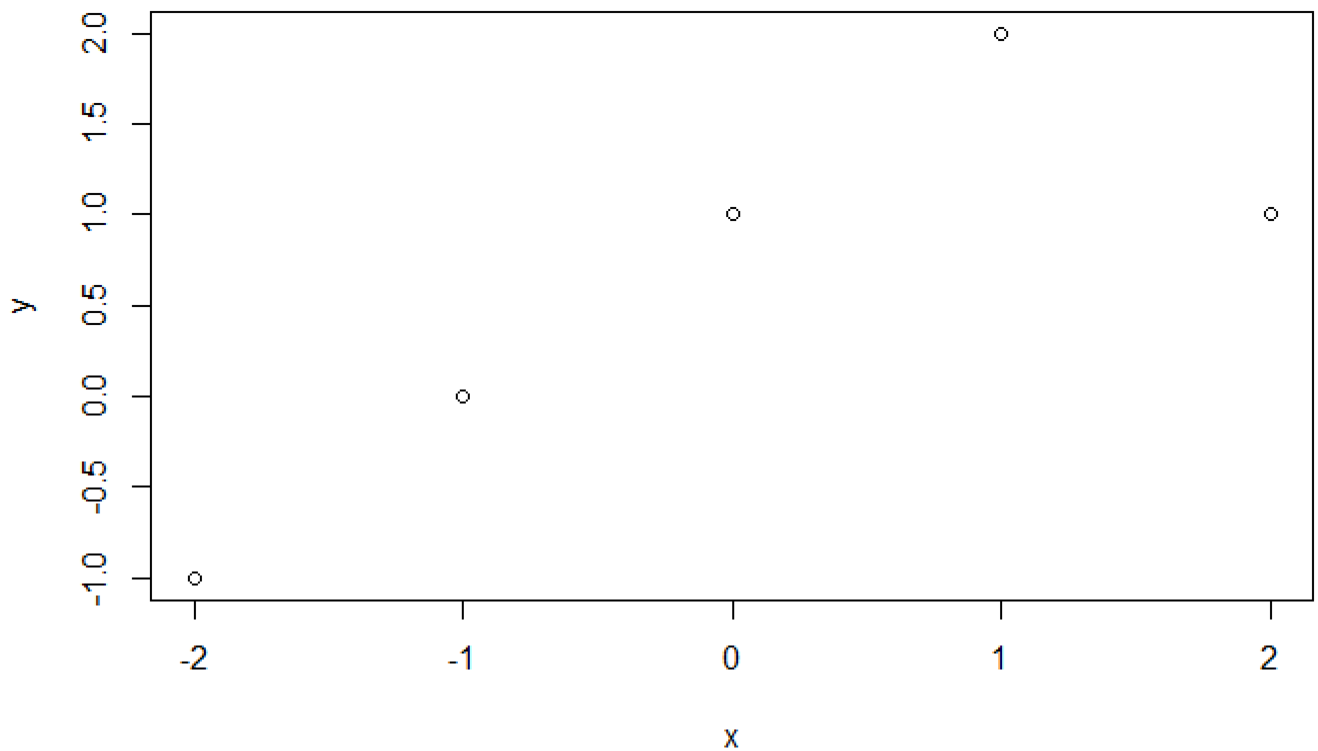
a. Find a cubic polynomial f1(x) = a1 + b1x + c1x2 + d1x3 such that f(x) = f1(x) for all x ??? ??. Express a1, b1, c1, d1 in terms of ??0, ??1, ??2, ??3, ??4.
b. Find a cubic polynomial f2(x) = a2 + b2x + c2x2 + d2x3 such that f(x) = f2(x) for all x > ??. Express a2, b2, c2, d2 in terms of ??0, ??1, ??2, ??3, ??4. We have now established that f(x) is a piecewise polynomial.
c. Show that f1(??) = f2(??). That is, f(x) is continuous at ??.
d. Show that f□1(??) = f□2(??). That is, f□(x) is continuous at ??. 298 7. Moving Beyond Linearity
e. Show that f□.1 (??) = f□□(??). That is, f□□(x) is continuous at ??.

Solution is written in the note.

3. Suppose we fit a curve with basis functions b1(X) = X, b2(X) = (X ??? 1)2I(X ??? 1). (Note that I(X ??? 1) equals 1 for X ??? 1 and 0 otherwise.) We fit the linear regression model Y = ??0 + ??1b1(X) + ??2b2(X) + error, and obtain coefficient estimates ^ ??0 = 1, ^ ??1 = 1, ^??2 = ???2. Sketch the estimated curve between X = ???2 and X = 2. Note the intercepts, slopes, and other relevant information.

Hide

```
x = -2:2
y = 1 + x + -2 * (x-1)^2 * I(x>1)
plot(x, y)
```



10. This question relates to the College data set.

a. Split the data into a training set and a test set. Using out-of-state tuition as the response and the other variables as the predictors, perform forward stepwise selection on the training set in order to identify a satisfactory model that uses just a subset of the predictors.
b. Fit a GAM on the training data, using out-of-state tuition as the response and the features selected in the previous step as the predictors. Plot the results, and explain your findings.

c. Evaluate the model obtained on the test set, and explain the results obtained.
d. For which variables, if any, is there evidence of a non-linear relationship with the response?

Getting the data

Hide

```
library(ISLR)
```

```
package <U+393C><U+3E31>ISLR<U+393C><U+3E32> was built under R version 3.5.3
```

Hide

```
set.seed(1)
attach(College)
```

A. Dividing data into traing and test dataset

Hide

```
train <- sample(length(Outstate), length(Outstate) / 2)
test <- -train
College.train <- College[train, ]
College.test <- College[test, ]
```

Fitting the model and model accuracy on training dataset

Hide

```
library(leaps)
fit <- regsubsets(Outstate ~ ., data = College.train, nvmax = 17, method = "forward")
fit.summary <- summary(fit)
par(mfrow = c(1, 3))
plot(fit.summary$cp, xlab = "Number of variables", ylab = "Cp", type = "l")
min.cp <- min(fit.summary$cp)
std.cp <- sd(fit.summary$cp)
abline(h = min.cp + 0.2 * std.cp, col = "red", lty = 2)
```

Hide

```
abline(h = min.cp - 0.2 * std.cp, col = "red", lty = 2)
plot(fit.summary$bic, xlab = "Number of variables", ylab = "BIC", type='l')
```

Hide

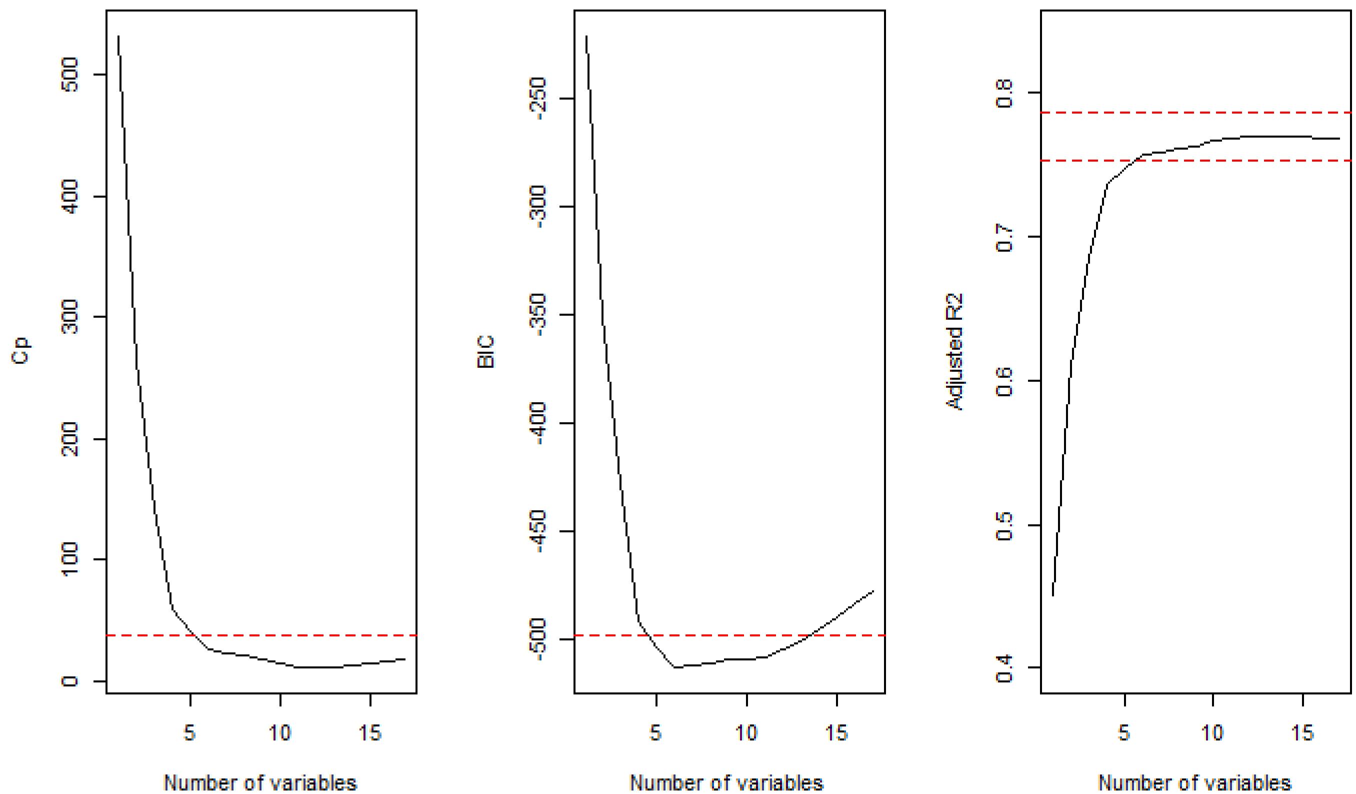
```
min.bic <- min(fit.summary$bic)
std.bic <- sd(fit.summary$bic)
abline(h = min.bic + 0.2 * std.bic, col = "red", lty = 2)
abline(h = min.bic - 0.2 * std.bic, col = "red", lty = 2)
```

Hide

```
plot(fit.summary$adjr2, xlab = "Number of variables", ylab = "Adjusted R2", type = "l", ylim = c(0.4, 0.84))
max.adjr2 <- max(fit.summary$adjr2)
std.adjr2 <- sd(fit.summary$adjr2)
abline(h = max.adjr2 + 0.2 * std.adjr2, col = "red", lty = 2)
```

Hide

```
abline(h = max.adjr2 - 0.2 * std.adjr2, col = "red", lty = 2)
```



We can see that best accuracy with minimum number of variables is at subset when number of variables are 6. Cp, BIC and adjr2 show that size 6 is the minimum size for the subset for which the scores are within 0.2 standard devitations of optimum.

Hide

```
fit <- regsubsets(Outstate ~ ., data = College, method = "forward")
coeffs <- coef(fit, id = 6)
names(coeffs)
```

```
[1] "(Intercept)" "PrivateYes"  "Room.Board"  "PhD"         "perc.alumni" "Expend"      "Grad.Rate"
```

B.

Hide

```
install.packages("gam")
```

```
Installing package into <U+393C><U+3E31>C:/Users/deepa/OneDrive/Documents/R/win-library/3.5<U+393C><U+3E32>
(as <U+393C><U+3E31>lib<U+393C><U+3E32> is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/gam_1.16.1.zip'
Content type 'application/zip' length 403310 bytes (393 KB)
downloaded 393 KB
```

```
package □gam□ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\deepa\AppData\Local\Temp\RtmpEnZmM7\downloaded_packages
```

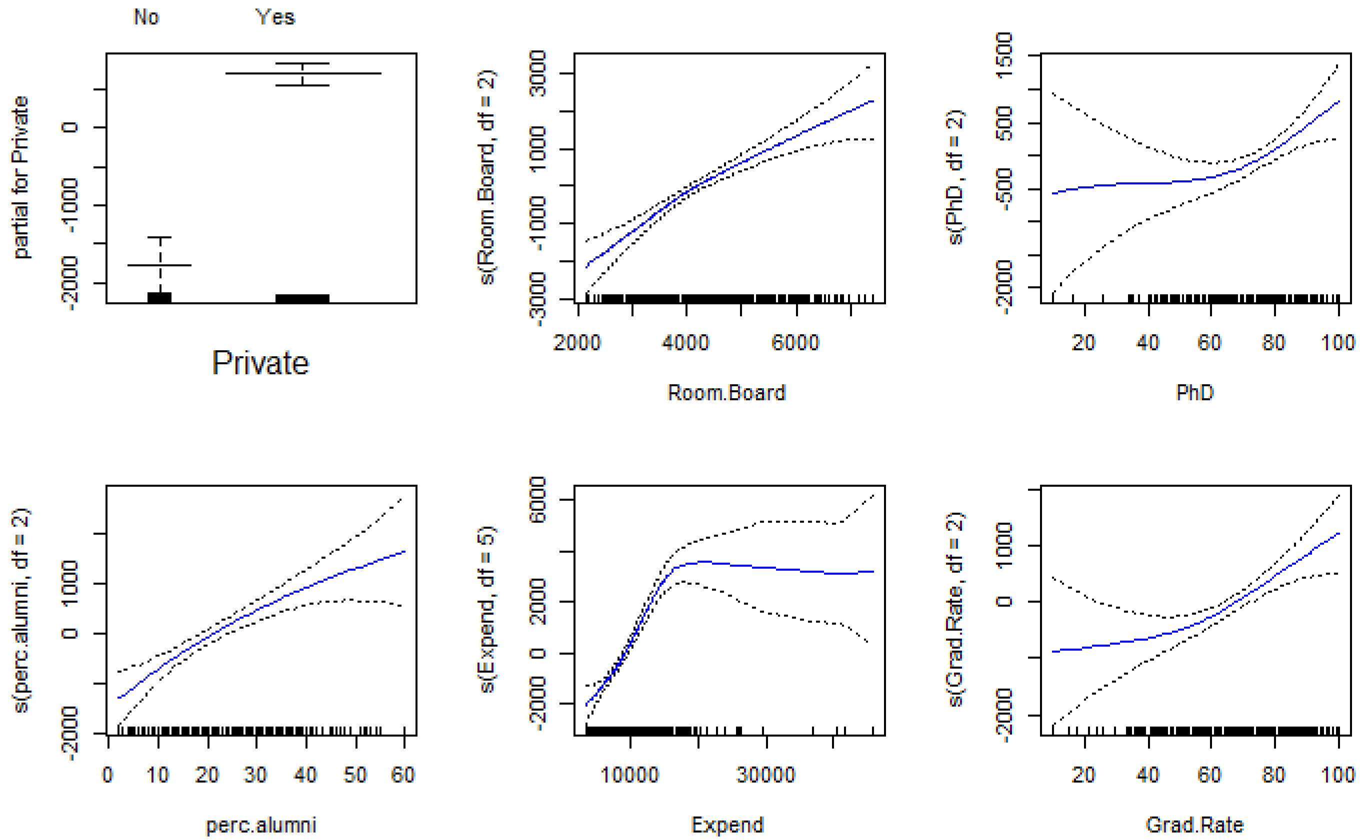
Hide

```
library(gam)
```

```
package <U+393C><U+3E31>gam<U+393C><U+3E32> was built under R version 3.5.3Loading required package: splines
Loading required package: foreach
Loaded gam 1.16.1
```

Hide

```
fit <- gam(Outstate ~ Private + s(Room.Board, df = 2) + s(PhD, df = 2) + s(perc.alumni, df = 2) + s(Expend, df = 5) + s(Gra
d.Rate, df = 2), data=College.train)
par(mfrow = c(2, 3))
plot(fit, se = T, col = "blue")
```



C.

```
preds <- predict(fit, College.test)
err <- mean((College.test$Outstate - preds)^2)
err
```

```
[1] 3745460
```

```
tss <- mean((College.test$Outstate - mean(College.test$Outstate))^2)
rss <- 1 - err / tss
rss
```

```
[1] 0.7696916
```

D.

Hide

```
summary(fit)
```

```
Call: gam(formula = Outstate ~ Private + s(Room.Board, df = 2) + s(PhD,
    df = 2) + s(perc.alumni, df = 2) + s(Expend, df = 5) + s(Grad.Rate,
    df = 2), data = College.train)
Deviance Residuals:
     Min       1Q   Median       3Q      Max
-4977.74 -1184.52    58.33  1220.04  7688.30

(Dispersion Parameter for gaussian family taken to be 3300711)

    Null Deviance: 6221998532 on 387 degrees of freedom
Residual Deviance: 1231165118 on 373 degrees of freedom
AIC: 6941.542

Number of Local Scoring Iterations: 2

Anova for Parametric Effects
                        Df     Sum Sq    Mean Sq F value    Pr(>F)
Private                  1 1779433688 1779433688 539.106 < 2.2e-16 ***
s(Room.Board, df = 2)    1 1221825562 1221825562 370.171 < 2.2e-16 ***
s(PhD, df = 2)           1  382472137  382472137 115.876 < 2.2e-16 ***
s(perc.alumni, df = 2)   1  328493313  328493313  99.522 < 2.2e-16 ***
s(Expend, df = 5)        1  416585875  416585875 126.211 < 2.2e-16 ***
s(Grad.Rate, df = 2)     1   55284580   55284580  16.749 5.232e-05 ***
Residuals              373 1231165118    3300711
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Anova for Nonparametric Effects
                       Npar Df  Npar F     Pr(F)
(Intercept)
Private
s(Room.Board, df = 2)        1  3.5562   0.06010 .
s(PhD, df = 2)               1  4.3421   0.03786 *
s(perc.alumni, df = 2)       1  1.9158   0.16715
s(Expend, df = 5)            4 16.8636 1.016e-12 ***
s(Grad.Rate, df = 2)         1  3.7208   0.05450 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Tree Based Methods ………………………………………………………………………………………………………..

3. Consider the Gini index, classification error, and cross-entropy in a simple classification setting with two classes. Create a single plot that displays each of these quantities as a function of ^pm1. The xaxis should display ^pm1, ranging from 0 to 1, and the y-axis should display the value of the Gini index, classification error, and entropy. Hint: In a setting with two classes, ^pm1 = 1??? ^pm2. You could make this plot by hand, but it will be much easier to make in R.

Hide

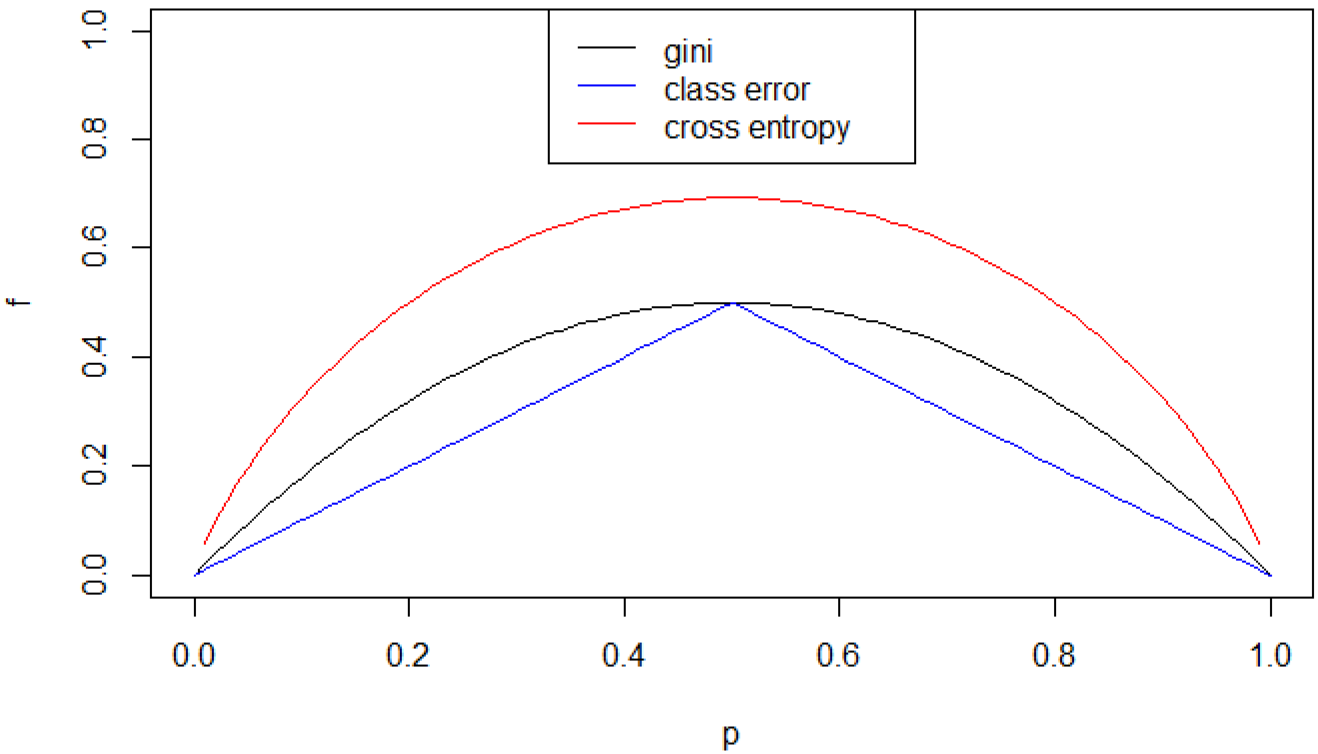
```
p=seq(0,1,0.01)
gini= 2*p*(1-p)
classerror= 1-pmax(p,1-p)
crossentropy= -(p*log(p)+(1-p)*log(1-p))
plot(NA,NA,xlim=c(0,1),ylim=c(0,1),xlab='p',ylab='f')
lines(p,gini,type='l')
```

Hide

```
lines(p,classerror,col='blue')
lines(p,crossentropy,col='red')
```

Hide

```
legend(x='top',legend=c('gini','class error','cross entropy'),
       col=c('black','blue','red'),lty=1,text.width = 0.22)
```



10. We now use boosting to predict Salary in the Hitters data set.

a. Remove the observations for whom the salary information is unknown, and then log-transform the salaries.
b. Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.
c. 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 x-axis and the corresponding training set MSE on the y-axis.
d. Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.
e. Compare the test MSE of boosting to the test MSE that results from applying two of the regression approaches seen in Chapters 3 and 6.
f. Which variables appear to be the most important predictors in the boosted model?
g. Now apply bagging to the training set. What is the test set MSE for this approach?

A.

Hide

```
require(ISLR)
Hitters.unknownSal=is.na(Hitters[,"Salary"])
Hitters=Hitters[!Hitters.unknownSal,]
Hitters[,"Salary"]=log(Hitters[,"Salary"])
summary(Hitters)
```

```
    AtBat           Hits           HmRun           Runs             RBI              Walks
 Min.   : 19.0   Min.   :  1.0   Min.   : 0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
 1st Qu.:282.5   1st Qu.: 71.5   1st Qu.: 5.00   1st Qu.: 33.50   1st Qu.: 30.00   1st Qu.: 23.00
 Median :413.0   Median :103.0   Median : 9.00   Median : 52.00   Median : 47.00   Median : 37.00
 Mean   :403.6   Mean   :107.8   Mean   :11.62   Mean   : 54.75   Mean   : 51.49   Mean   : 41.11
 3rd Qu.:526.0   3rd Qu.:141.5   3rd Qu.:18.00   3rd Qu.: 73.00   3rd Qu.: 71.00   3rd Qu.: 57.00
 Max.   :687.0   Max.   :238.0   Max.   :40.00   Max.   :130.00   Max.   :121.00   Max.   :105.00
    Years            CAtBat            CHits            CHmRun           CRuns             CRBI
 Min.   : 1.000   Min.   :   19.0   Min.   :   4.0   Min.   :  0.00   Min.   :   2.0   Min.   :   3.0
 1st Qu.: 4.000   1st Qu.:  842.5   1st Qu.: 212.0   1st Qu.: 15.00   1st Qu.: 105.5   1st Qu.:  95.0
 Median : 6.000   Median : 1931.0   Median : 516.0   Median : 40.00   Median : 250.0   Median : 230.0
 Mean   : 7.312   Mean   : 2657.5   Mean   : 722.2   Mean   : 69.24   Mean   : 361.2   Mean   : 330.4
 3rd Qu.:10.000   3rd Qu.: 3890.5   3rd Qu.:1054.0   3rd Qu.: 92.50   3rd Qu.: 497.5   3rd Qu.: 424.5
 Max.   :24.000   Max.   :14053.0   Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.0
    CWalks        League  Division    PutOuts           Assists          Errors           Salary
 Min.   :   1.0   A:139   E:129    Min.   :   0.0   Min.   :  0.0   Min.   : 0.000   Min.   :4.212
 1st Qu.:  71.0   N:124   W:134    1st Qu.: 113.5   1st Qu.:  8.0   1st Qu.: 3.000   1st Qu.:5.247
 Median : 174.0                    Median : 224.0   Median : 45.0   Median : 7.000   Median :6.052
 Mean   : 260.3                    Mean   : 290.7   Mean   :118.8   Mean   : 8.593   Mean   :5.927
 3rd Qu.: 328.5                    3rd Qu.: 322.5   3rd Qu.:192.0   3rd Qu.:13.000   3rd Qu.:6.620
 Max.   :1566.0                    Max.   :1377.0   Max.   :492.0   Max.   :32.000   Max.   :7.808
 NewLeague
 A:141
 N:122
```

B.

Hide

```
Hitters.train=Hitters[1:200,]
Hitters.test=Hitters[-c(1:200),]
```

C. and D.

Hide

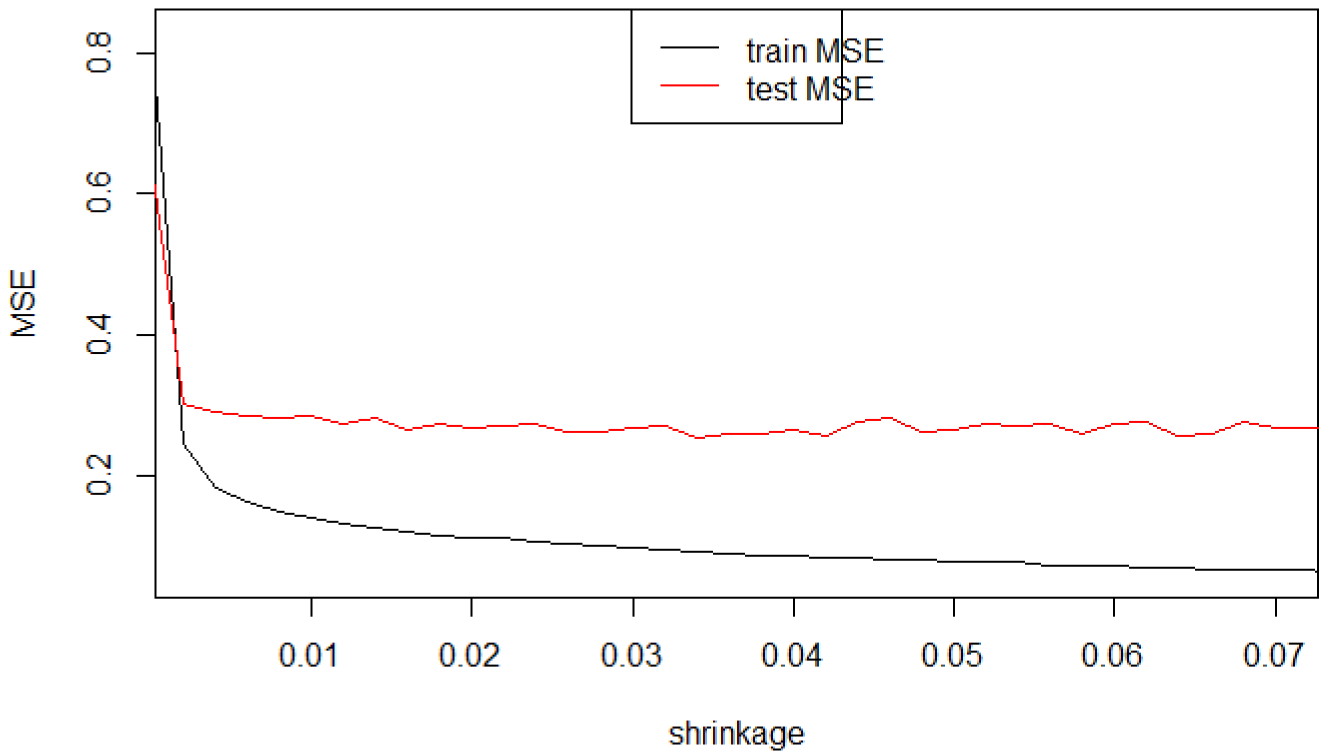
```
require(gbm)
train.mse=c()
test.mse=c()
for(shr in seq(0,0.08,0.002)){
  Hitters.gbm=gbm(Salary~.,data=Hitters.train,shrinkage = shr,n.trees = 1000,distribution = 'gaussian')

  Hitters.pred=predict(Hitters.gbm,Hitters.train,n.trees = 1000)
  train.mse=rbind(train.mse,mean((Hitters.pred-Hitters.train[,'Salary'])^2))

  Hitters.pred=predict(Hitters.gbm,Hitters.test,n.trees = 1000)
  test.mse=rbind(test.mse,mean((Hitters.pred-Hitters.test[,'Salary'])^2))
}
plot(seq(0,0.08,0.002),train.mse,type='l',xlab='shrinkage',xlim = c(0.003,0.07),ylab='MSE')
lines(seq(0,0.08,0.002),test.mse,col='red')
```

Hide

```
legend(x='top',legend = c('train MSE','test MSE'),col=c('black','red'),lty=1,text.width = 0.005)
```


train MSE
test MSE
MSE
0.2
0.4
0.6
0.8
0.01
0.02
0.03
0.04
0.05
0.06
0.07
shrinkage

E.

Hide

```
tb=c()
Hitters.gbm=gbm(Salary~.,data=Hitters.train,shrinkage = 0.01,n.trees = 1000,distribution = 'gaussian')
Hitters.pred=predict(Hitters.gbm,Hitters.test,n.trees = 1000)
tb=cbind(tb,'Boost'=mean((Hitters.pred-Hitters.test[,'Salary'])^2))
Hitters.lm=lm(Salary~.,Hitters.train)
Hitters.pred=predict(Hitters.lm,Hitters.test)
tb=cbind(tb,'Linear'=mean((Hitters.pred-Hitters.test[,'Salary'])^2))
require(glmnet)
x = model.matrix(Salary ~ ., data = Hitters.train)
x.test = model.matrix(Salary ~ ., data = Hitters.test)
y = Hitters.train$Salary
Hitters.glm=glmnet(x,y,alpha = 0)
Hitters.pred=predict(Hitters.glm,x.test)
tb=cbind(tb,'Ridge'=mean((Hitters.pred-Hitters.test[,'Salary'])^2))
tb
```

```
         Boost    Linear     Ridge
[1,] 0.2798657 0.4917959 0.5145349
```

F.

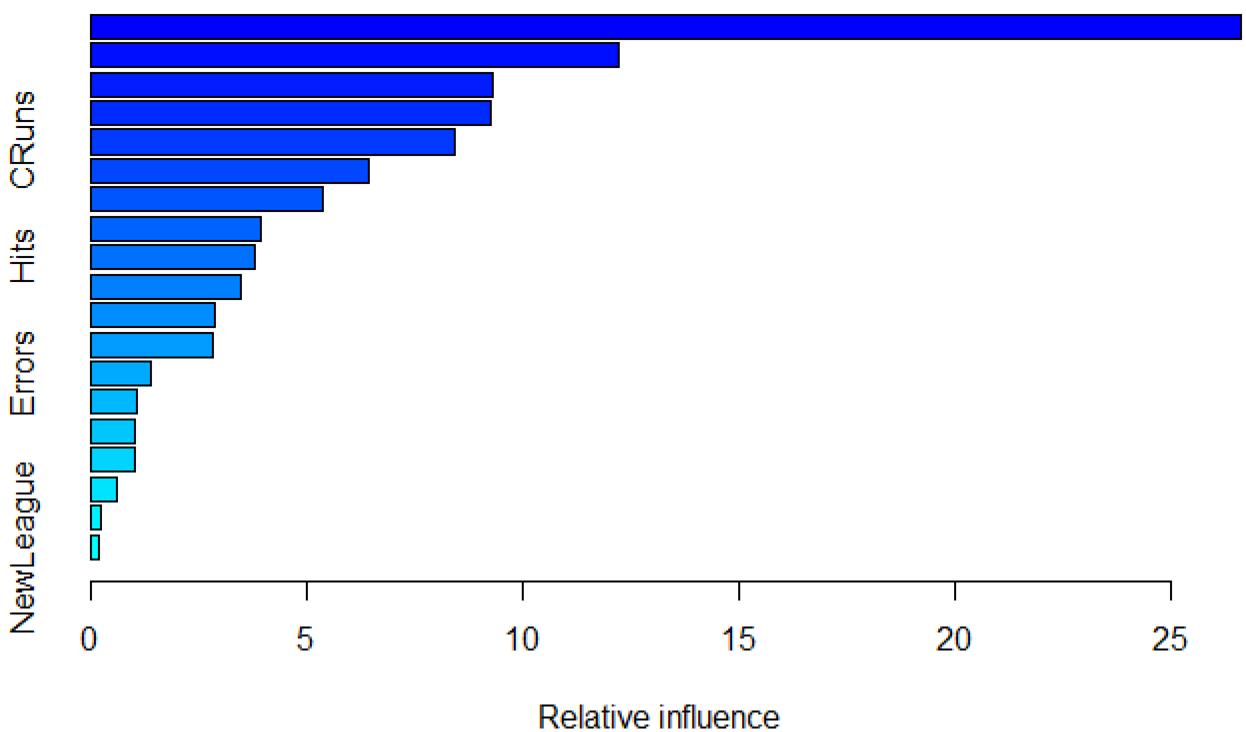
Hide

```
summary(Hitters.gbm)
```

	var <fctr>	rel.inf <dbl>
CAtBat	CAtBat	26.6077354
CRBI	CRBI	12.2331811
CHits	CHits	9.2955985
CWalks	CWalks	9.2487246
CRuns	CRuns	8.4064675
Years	Years	6.4343934

	var <fctr>	rel.inf <dbl>
CHmRun	CHmRun	5.3866154
Walks	Walks	3.9216429
Hits	Hits	3.8058443
RBI	RBI	3.4909732

1-10 of 19 rows Previous **1** 2 Next



G.

Hide

```
#install.packages("randomForest")
library(randomForest)
Hitters.rf=randomForest(Salary~.,data = Hitters.train,mtry=ncol(Hitters.train)-1) # bagging m=p
Hitters.pred=predict(Hitters.rf,Hitters.test)
mean((Hitters.pred-Hitters.test[,'Salary'])^2)
```

```
[1] 0.2290734
```

Thus this is Test Mean Squared Error

Deepak Kumar Tiwari

HW-4 ECG-590

1/

(a) $f_1(x) = a_1 + b_1 x + c_1 x^2 + d_1 x^3$

we need to find $f_1(x)$ in terms of $\beta_0 + \beta_1, \beta_2, \beta_3, \beta_4$
such that $f(x) = f_1(x)$ for all $x \le \xi$

For $x \le \xi$ we have

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3.$$

so, $a_1 = \beta_0$, $b = \beta_1$, $c_1 = \beta_2$ and $d_1 = \beta_3$.

(b) For $x \le \xi$ we have.

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_4 (x - \xi)^3$$

$$= (\beta_0 - \beta_4 \xi^3) + (\beta_1 + 3\xi^2 \beta_4)x + (\beta_2 - 3\beta_4 \xi)x^2$$
$$+ (\beta_3 + \beta_4)x^3$$

then we have

$$a_2 = \beta_0 - \beta_4 \xi^3, \quad b_2 = \beta_1 + 3\xi^2 \beta_4$$
$$c_2 = \beta_2 - 3\beta_4 \xi \quad \text{and} \quad d_2 = \beta_3 + \beta_4$$

(c)

$$f_1(\xi) = f_2(\xi)$$

$$f_1(\xi) = \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3$$

and $f_2(\xi) = (\beta_0 - \beta_4\xi^3) + (\beta_1 + 3\xi^2\beta_4)\xi + (\beta_2 - 3\beta_4\xi)\xi^2$

$+ (\beta_3 + \beta_4)\xi^3$

$= \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3$

so thus $f_1(\xi) = f_2(\xi)$, continuous at ξ.

(d) $f'(\xi) = f_2'(\xi)$

$f'(\xi) = \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2$

$f_2'(\xi) = \beta_1 + 3\xi^2\beta_4 + 2(\beta_2 - 3\beta_4\xi)\xi + 3(\beta_3 + \beta_4)\xi^2$

$= \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2$

thus $f'(x)$ continuous at ξ.

(e) $f''(\xi) = f_2''(\xi)$

$f_1''(\xi) = 2\beta_2 + 6\beta_3\xi^2$.

and $f_2''(\xi) = 2(\beta_2 - 3\beta_4\xi) + 6(\beta_3 + \beta_4)\xi$

$= 2\beta_2 + 6\beta_3\xi$

thus $f_2''(x)$ continuous at ξ