##################
Book Problems
##################

#########################
Chapter 2: Question 10
#########################

```
rm(list = ls())
library(MASS)
myData = Boston
attach(myData)
```

(a)

R Script:

```
dim(myData)
```

Output:

```
[1] 506  14
```

Explanation:

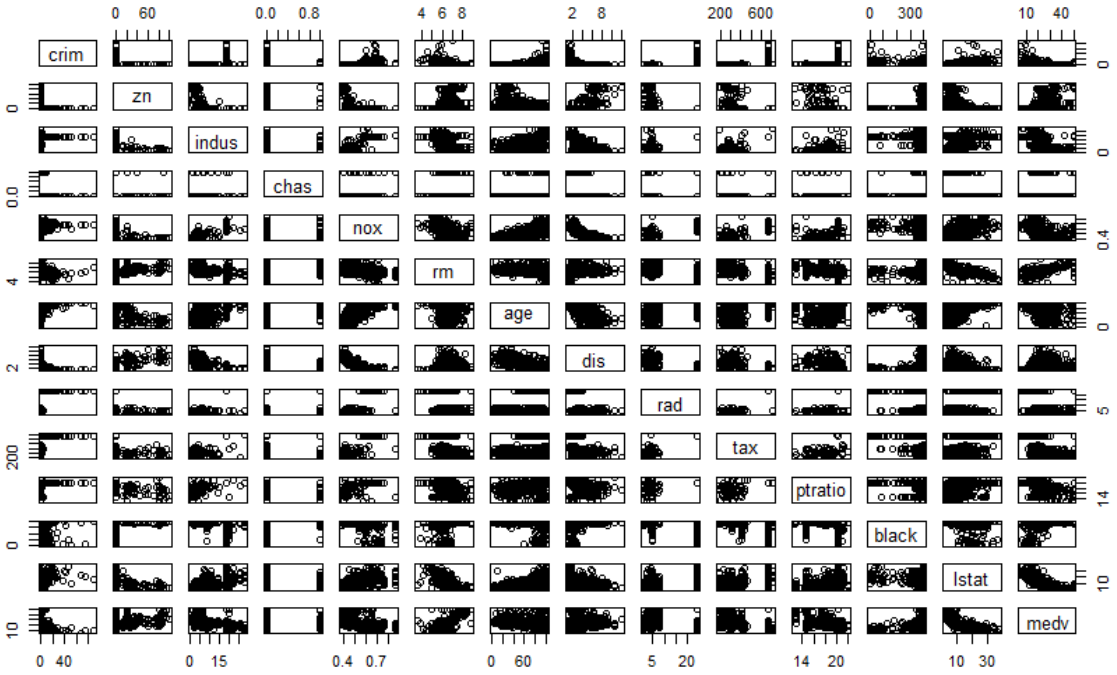
Rows: 506 Boston Suburbs
Columns: 14 Features

(b)

R Script:

```
pairs(myData)
```

Output:



Explanation:

- Crim correlates with Age, Dis, Rad, Tax and Ptratio
- Zn correlates with Indus, Nox, Age, Lstat
- Indus correlates with Age, Dis
- Nox correlates with Age, Dis
- Dis correlates with Lstat
- Lstat correlates with Medv

(c)

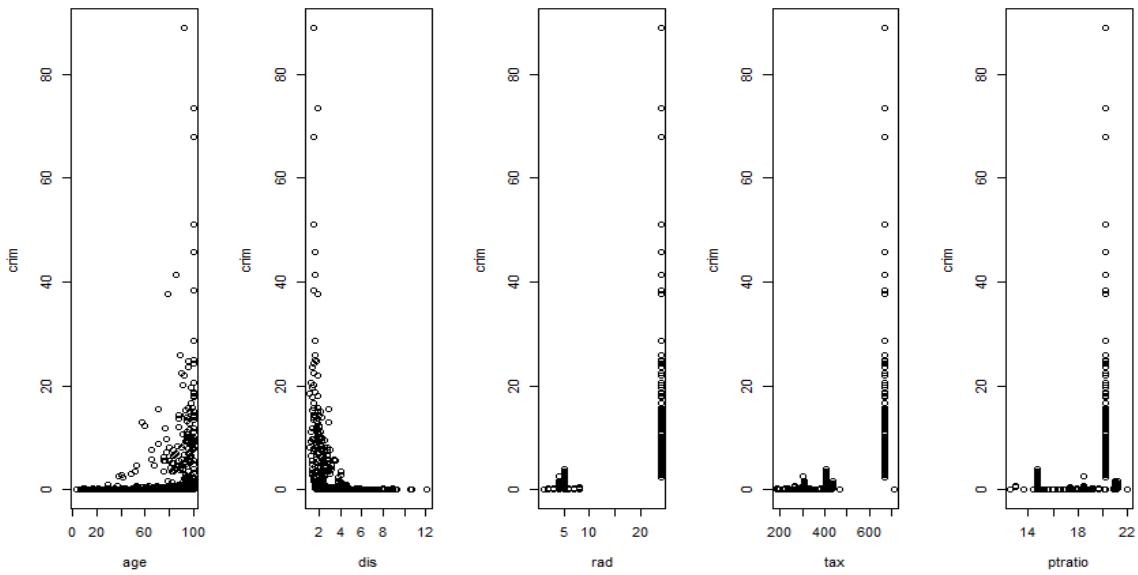
R Script:

```
x <- myData[1]
y <- myData[2 : 7]
z <- myData[8 : 14]

cor(x, y)
cor(x, z)

par(mfrow=c(1, 5))
plot(age, crim)
plot(dis, crim)
plot(rad, crim)
plot(tax, crim)
plot(ptratio, crim)
```

Output:

	zn	indus	chas	nox	rm	age
crim	-0.2004692	0.4065834	-0.05589158	0.4209717	-0.2192467	0.3527343

	dis	rad	tax	ptratio	black	lstat	medv
crim	-0.3796701	0.6255051	0.5827643	0.2899456	-0.3850639	0.4556215	-0.3883046



Explanation:

- Older homes, More crime
- Closer to work-area, More crime
- Higher accessibility to highways, More crime
- Higher tax rate, More crime
- Higher pupil-teacher ratio, More crime

(d)

R Script:

```
summary(crim)
summary(tax)
summary(ptratio)

par(mfrow=c(1,3))
boxplot(crim, main="Crime Rates")
boxplot(tax, main="Tax Rates")
boxplot(ptratio, main="Pupil-teacher ratios")
```

Output:

Crim

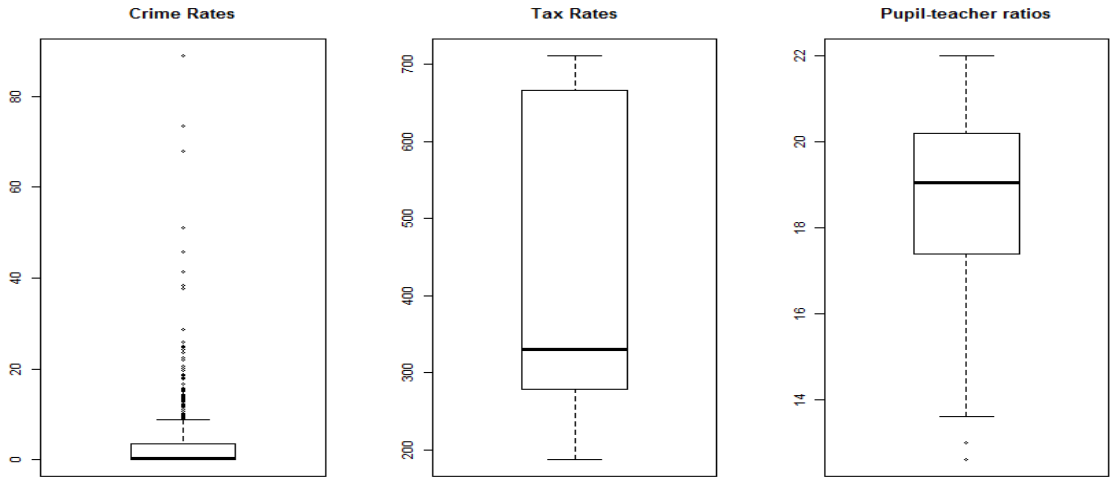
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00632	0.08204	0.25650	3.61400	3.67700	88.98000

Tax

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
187.0	279.0	330.0	408.2	666.0	711.0

Ptratio

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
12.60	17.40	19.05	18.46	20.20	22.00



Explanation:

- Crim: (0.00632 – 88.98), very wide range
- Tax: (187 – 711), Not so wide range
- Ptratio: (12.6 – 22), Small range

(e)

R Script:

```
margin.table(table(crim,chas),2)
```

Output:

```
chas
  0   1
471  35
```

Explanation:

35 suburbs bound the Charles river

(f)

R Script:

```
median(ptratio)
```

Output:

```
[1] 19.05
```

(g)

R Script:

```
temp <- myData[order(medv), ]
x <- sapply(myData, summary)
y <- temp[c(1, 2), ]

z <- as.data.frame(rbind(y, x))

z1 <- z[, 1:7]
z2 <- z[, 8:14]
z1
z2
```

Output:

	crim	zn	indus	chas	nox	rm	age
399	38.35180	0.00	18.10	0.00000	0.6930	5.453	100.00
406	67.92080	0.00	18.10	0.00000	0.6930	5.683	100.00
Min.	0.00632	0.00	0.46	0.00000	0.3850	3.561	2.90
1st Qu.	0.08204	0.00	5.19	0.00000	0.4490	5.886	45.02
Median	0.25650	0.00	9.69	0.00000	0.5380	6.208	77.50
Mean	3.61400	11.36	11.14	0.06917	0.5547	6.285	68.57
3rd Qu.	3.67700	12.50	18.10	0.00000	0.6240	6.624	94.07
Max.	88.98000	100.00	27.74	1.00000	0.8710	8.780	100.00

	dis	rad	tax	ptratio	black	lstat	medv
399	1.4896	24.000	666.0	20.20	396.90	30.59	5.00
406	1.4254	24.000	666.0	20.20	384.97	22.98	5.00
Min.	1.1300	1.000	187.0	12.60	0.32	1.73	5.00
1st Qu.	2.1000	4.000	279.0	17.40	375.40	6.95	17.02
Median	3.2070	5.000	330.0	19.05	391.40	11.36	21.20
Mean	3.7950	9.549	408.2	18.46	356.70	12.65	22.53
3rd Qu.	5.1880	24.000	666.0	20.20	396.20	16.96	25.00
Max.	12.1300	24.000	711.0	22.00	396.90	37.97	50.00

Explanation:

- Suburbs 399 and 406 have the lowest median value of owner occupied homes
- Per capita crime rate is high
- Not the best place to live

(h)

R Script:

```
length(rm[rm > 7])
length(rm[rm > 8])

summary(myData[which(rm > 8),])
```

Output:

```
[1] 64
```

```
[1] 13
```

crim	zn	indus	chas	nox	rm	age
Min. :0.02009	Min. : 0.00	Min. : 2.680	Min. :0.0000	Min. :0.4161	Min. :8.034	Min. : 8.40
1st Qu.:0.33147	1st Qu.: 0.00	1st Qu.: 3.970	1st Qu.:0.0000	1st Qu.:0.5040	1st Qu.:8.247	1st Qu.:70.40
Median :0.52014	Median : 0.00	Median : 6.200	Median :0.0000	Median :0.5070	Median :8.297	Median :78.30
Mean :0.71879	Mean :13.62	Mean : 7.078	Mean :0.1538	Mean :0.5392	Mean :8.349	Mean :71.54
3rd Qu.:0.57834	3rd Qu.:20.00	3rd Qu.: 6.200	3rd Qu.:0.0000	3rd Qu.:0.6050	3rd Qu.:8.398	3rd Qu.:86.50
Max. :3.47428	Max. :95.00	Max. :19.580	Max. :1.0000	Max. :0.7180	Max. :8.780	Max. :93.90

dis	rad	tax	ptratio	black	lstat	medv
Min. :1.801	Min. : 2.000	Min. :224.0	Min. :13.00	Min. :354.6	Min. :2.47	Min. :21.9
1st Qu.:2.288	1st Qu.: 5.000	1st Qu.:264.0	1st Qu.:14.70	1st Qu.:384.5	1st Qu.:3.32	1st Qu.:41.7
Median :2.894	Median : 7.000	Median :307.0	Median :17.40	Median :386.9	Median :4.14	Median :48.3
Mean :3.430	Mean : 7.462	Mean :325.1	Mean :16.36	Mean :385.2	Mean :4.31	Mean :44.2
3rd Qu.:3.652	3rd Qu.: 8.000	3rd Qu.:307.0	3rd Qu.:17.40	3rd Qu.:389.7	3rd Qu.:5.12	3rd Qu.:50.0
Max. :8.907	Max. :24.000	Max. :666.0	Max. :20.20	Max. :396.9	Max. :7.44	Max. :50.0

Explanation:

- Relatively lower per capita crime rate
- Relatively lower Lstat

```
#########################
#  Chapter 3: Question 15  #
#########################

rm(list = ls())

library(MASS)
myData = Boston
attach(myData)
```

(a)

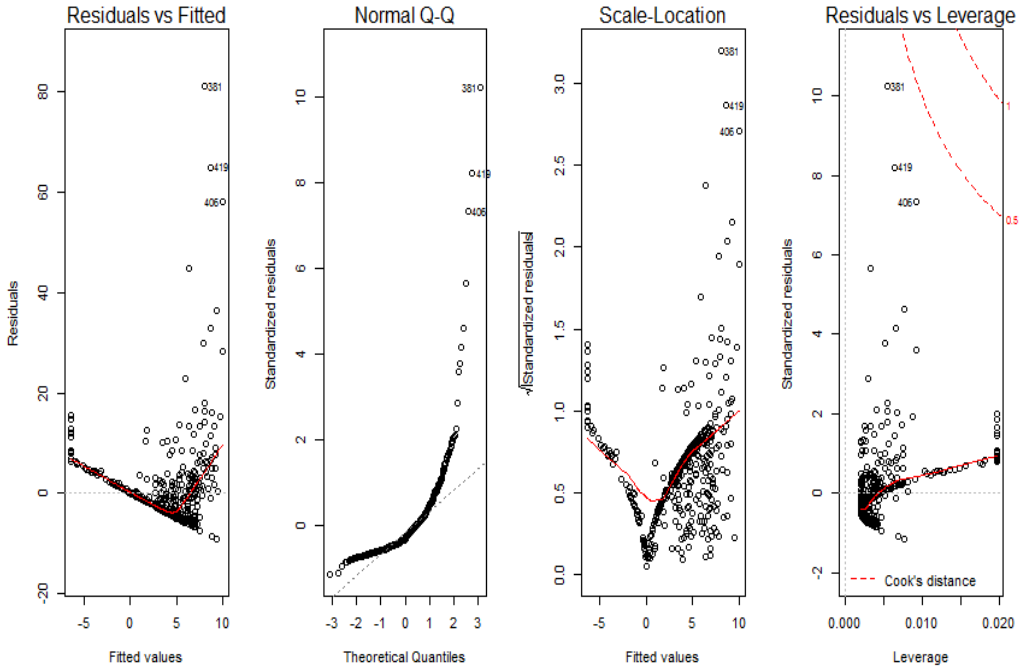
R Script:

```
par(mfrow=c(1, 4))
chas <- factor(chas, labels = c("N","Y"))

lm.zn = lm(crim~zn)
summary(lm.zn)
plot(lm.zn)

lm.indus = lm(crim~indus)
summary(lm.indus)
plot(lm.indus)

lm.chas = lm(crim~chas)
summary(lm.chas)
plot(lm.chas)

lm.nox = lm(crim~nox)
summary(lm.nox)
plot(lm.nox)

lm.rm = lm(crim~rm)
summary(lm.rm)
plot(lm.rm)

lm.age = lm(crim~age)
summary(lm.age)
plot(lm.age)

lm.dis = lm(crim~dis)
summary(lm.dis)
plot(lm.dis)

lm.rad = lm(crim~rad)
summary(lm.rad)
plot(lm.rad)
```

```
lm.tax = lm(crim~tax)
summary(lm.tax)
plot(lm.tax)

lm.ptratio = lm(crim~ptratio)
summary(lm.ptratio)
plot(lm.ptratio)

lm.black = lm(crim~black)
summary(lm.black)
plot(lm.black)

lm.lstat = lm(crim~lstat)
summary(lm.lstat)
plot(lm.lstat)

lm.medv = lm(crim~medv)
summary(lm.medv)
plot(lm.medv)
```

Output:

```
Call:
lm(formula = crim ~ medv)

Residuals:
   Min     1Q Median     3Q    Max
-9.071 -4.022 -2.343  1.298 80.957

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.79654    0.93419   12.63   <2e-16 ***
medv        -0.36316    0.03839   -9.46   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.934 on 504 degrees of freedom
Multiple R-squared:  0.1508,    Adjusted R-squared:  0.1491
F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```



Explanation:

- A sample summary output and residual plot of linear regression using medv as the predictor variable is shown above
- All variable except chas are statistically significant

(b)

R Script:

```
lm.fit = lm(crim~., data = myData)
summary(lm.fit)
```

Output:

```
Call:
lm(formula = crim ~ ., data = myData)

Residuals:
    Min      1Q  Median      3Q     Max
-9.924  -2.120  -0.353   1.019  75.051

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  17.033228   7.234903   2.354 0.018949 *
zn            0.044855   0.018734   2.394 0.017025 *
indus        -0.063855   0.083407  -0.766 0.444294
chas         -0.749134   1.180147  -0.635 0.525867
nox         -10.313535   5.275536  -1.955 0.051152 .
rm            0.430131   0.612830   0.702 0.483089
age           0.001452   0.017925   0.081 0.935488
dis          -0.987176   0.281817  -3.503 0.000502 ***
rad           0.588209   0.088049   6.680 6.46e-11 ***
tax          -0.003780   0.005156  -0.733 0.463793
ptratio      -0.271081   0.186450  -1.454 0.146611
black        -0.007538   0.003673  -2.052 0.040702 *
lstat         0.126211   0.075725   1.667 0.096208 .
medv         -0.198887   0.060516  -3.287 0.001087 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.439 on 492 degrees of freedom
Multiple R-squared:  0.454,      Adjusted R-squared:  0.4396
F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

Explanation:

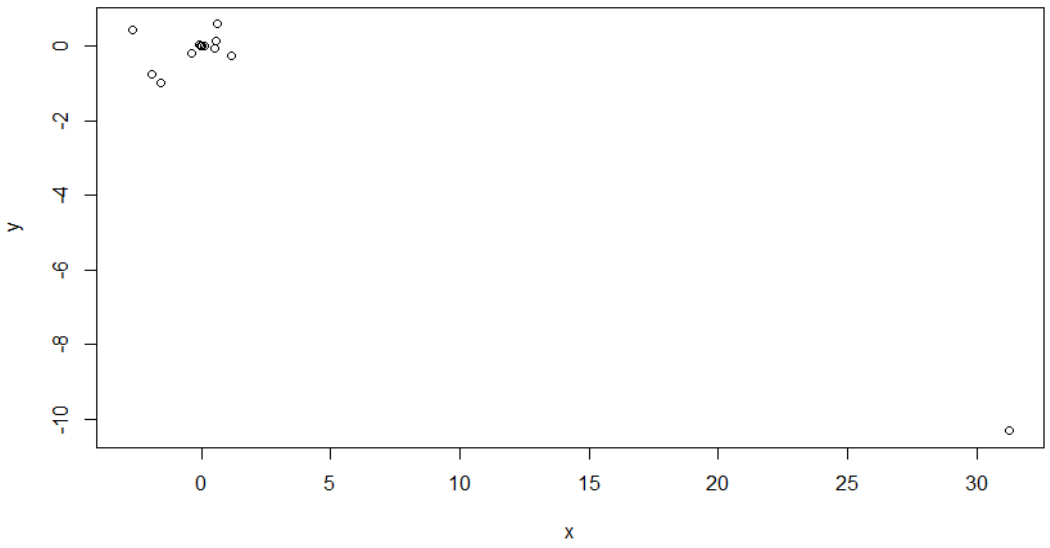
- We can reject the null hypothesis H0 : βj = 0 for zn, dis, rad, black and medv because |t| is greater than 2

(c)

R Script:

```
x = c(coefficients(lm.zn)[2], coefficients(lm.indus)[2], coefficients(lm.chas)[2], coefficients(lm.nox)[2], coefficients(lm.rm)[2], coefficients(lm.age)[2], coefficients(lm.dis)[2], coefficients(lm.rad)[2], coefficients(lm.tax)[2], coefficients(lm.ptratio)[2], coefficients(lm.black)[2], coefficients(lm.lstat)[2], coefficients(lm.medv)[2])
y = coefficients(lm.fit)[2:14]

plot(x, y)
```

Output:



Explanation:

- Coefficients for all the variables, except nox, is comparable in both univariate and linear regression model

(d)

R Script:

```
lm.zn = lm(crim~poly(zn,3))
summary(lm.zn)

lm.indus = lm(crim~poly(indus,3))
summary(lm.indus)

lm.nox = lm(crim~poly(nox,3))
summary(lm.nox)

lm.rm = lm(crim~poly(rm,3))
```

```
summary(lm.rm)

lm.age = lm(crim~poly(age,3))
summary(lm.age)

lm.dis = lm(crim~poly(dis,3))
summary(lm.dis)

lm.rad = lm(crim~poly(rad,3))
summary(lm.rad)

lm.tax = lm(crim~poly(tax,3))
summary(lm.tax)

lm.ptratio = lm(crim~poly(ptratio,3))
summary(lm.ptratio)

lm.black = lm(crim~poly(black,3))
summary(lm.black)

lm.lstat = lm(crim~poly(lstat,3))
summary(lm.lstat)

lm.medv = lm(crim~poly(medv,3))
summary(lm.medv)
```

Output:

```
Call:
lm(formula = crim ~ poly(medv, 3))

Residuals:
    Min      1Q  Median      3Q     Max
-24.427  -1.976  -0.437   0.439  73.655

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      3.614      0.292  12.374  < 2e-16 ***
poly(medv, 3)1 -75.058      6.569 -11.426  < 2e-16 ***
poly(medv, 3)2  88.086      6.569  13.409  < 2e-16 ***
poly(medv, 3)3 -48.033      6.569  -7.312 1.05e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.569 on 502 degrees of freedom
Multiple R-squared:  0.4202,    Adjusted R-squared:  0.4167
F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Explanation:

- A sample summary output for non-linear association using linear regression of medv as the predictor variable is shown above
- Except for black and chas, all the other variables have either quadratic or cubic relationship with the response variable i.e. per capita crime rate

```
#########################
#  Chapter 6: Question 9    #
#########################

rm(list = ls())

library(ISLR)
myData = College
attach(myData)
```

(a)

R Script:

```
set.seed(1)
train = sample(1:nrow(myData), nrow(myData)/2)
test = (-train)
myData.train = myData[train, ]
myData.test = myData[test, ]
```

(b)

R Script:

```
lm.fit = lm(Apps~., data = myData[train,])

lm.pred = predict(lm.fit, myData[test,])
val.errors = mean((myData$Apps[test] - lm.pred) ^ 2)
val.errors
```

Output:

```
[1] 1108531
```

Explanation:

Obtained test error is 1108531

(c)

R Script:

```
library(glmnet)

train.mat = model.matrix(Apps~., data=myData[train,])
test.mat = model.matrix(Apps~., data=myData[test,])
grid = 10 ^ seq(4, -2, length=100)

cv.out.ridge = cv.glmnet(train.mat, myData$Apps[train], alpha = 0, lambda=grid, thresh=1e-12)

bestIam.ridge = cv.out.ridge$lambda.min
bestIam.ridge

ridge.pred = predict(cv.out.ridge, s=bestIam.ridge, newx = test.mat)
ridge.errors = mean((myData$Apps[test] - ridge.pred)^2)
ridge.errors
```

Output:

```
[1] 1108514
```

Explanation:
Obtained test error is 1108514

(d)

R Script:

```
cv.out.lasso = cv.glmnet(train.mat, myData$Apps[train], alpha = 1, lambda=grid, thresh=1e-12)

bestIam.lasso = cv.out.lasso$lambda.min
bestIam.lasso

lasso.pred = predict(cv.out.lasso, s=bestIam.lasso, newx = test.mat)
lasso.errors = mean((myData$Apps[test] - lasso.pred)^2)
lasso.errors

lasso.mod = glmnet(model.matrix(Apps~., data=myData), myData[, "Apps"], alpha=1)
predict(lasso.mod, s=bestIam.lasso, type="coefficients")
```

Output:

```
[1] 1028718
```

```
19 x 1 sparse Matrix of class "dgCMatrix"
```

	1
(Intercept)	-6.491800e+02
(Intercept)	.
PrivateYes	-3.910803e+02
Accept	1.414561e+00
Enroll	-6.880133e-02
Top10perc	2.975286e+01
Top25perc	-5.898729e-03
F.Undergrad	.
P.Undergrad	8.118367e-03
Outstate	-4.802638e-02
Room.Board	1.154551e-01
Books	.
Personal	.
PhD	-4.573050e+00
Terminal	-3.263399e+00
S.F.Ratio	5.152699e-01
perc.alumni	-1.066814e+00
Expend	6.615233e-02
Grad.Rate	4.204566e+00

Explanation:
Obtaines test error is 1028718

(e)

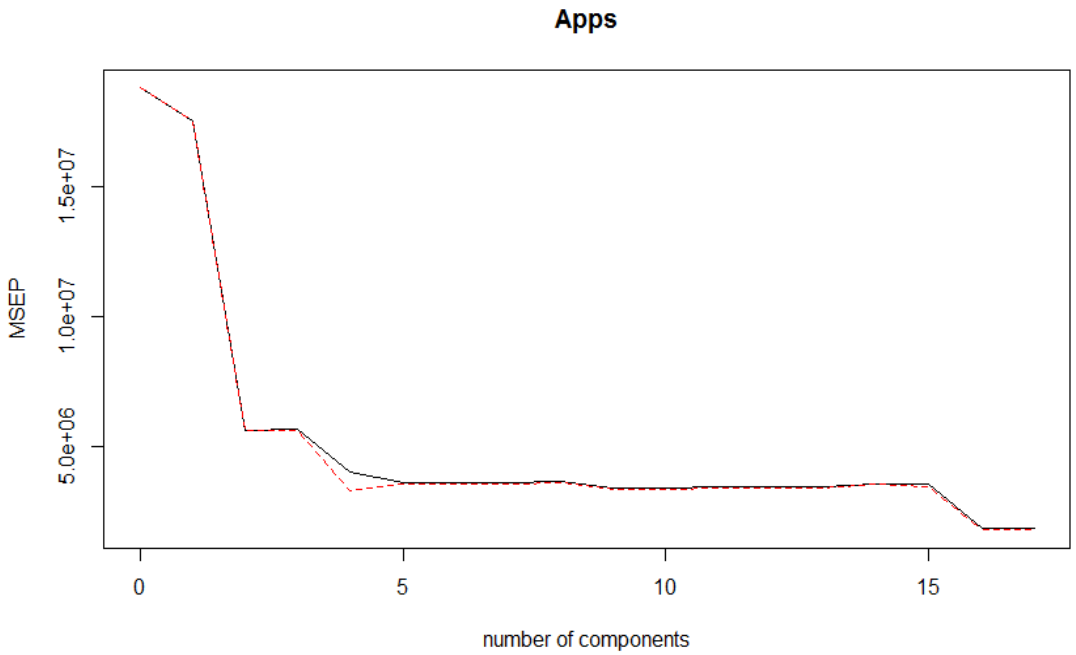
R Script:

```
library(pls)

pcr.fit = pcr(Apps~., data = myData, subset = train, scale = TRUE, validation = "CV")
validationplot(pcr.fit, val.type="MSEP")
```

```
summary(pcr.fit)

pcr.pred = predict(pcr.fit, myData[test,], ncomp = 16)
pcr.error = mean((myData$Apps[test] - pcr.pred)^2)
pcr.error
```

Output:

```
[1] 1166897
```



Explanation:

Obtained test error is 1166897

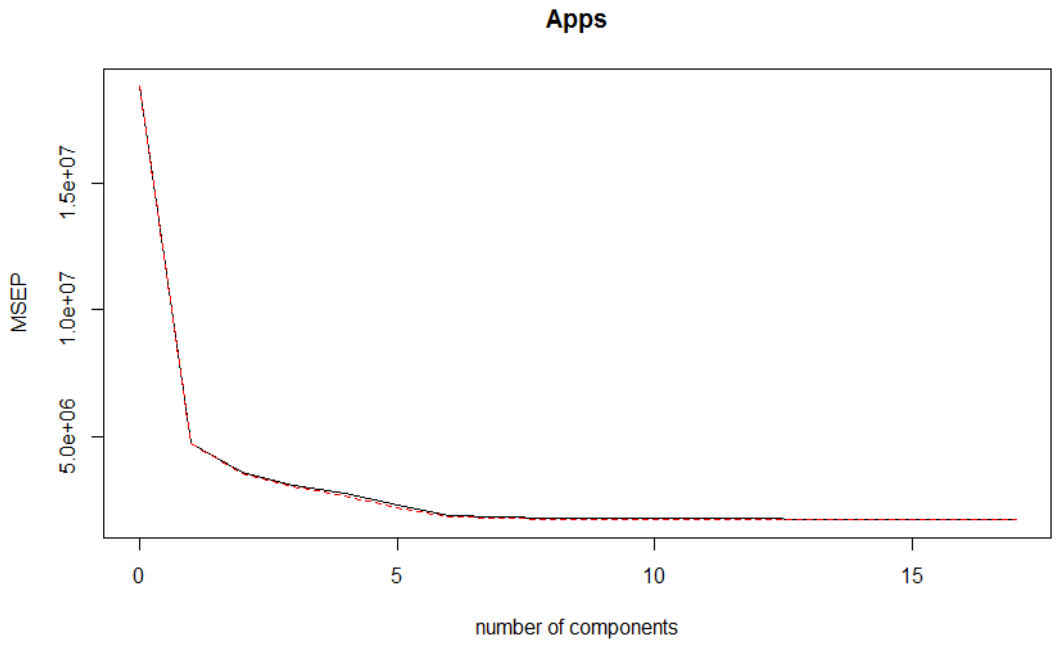
(f)

R Script:

```
pls.fit = plsr(Apps~., data = myData, subset = train, scale = TRUE, validation = "CV")
validationplot(pls.fit, val.type="MSEP")
summary(pls.fit)

pls.pred = predict(pls.fit, myData[test,], ncomp = 10)
pls.error = mean((myData$Apps[test] - pls.pred)^2)
pls.error
```

Output:

```
[1] 1134531
```



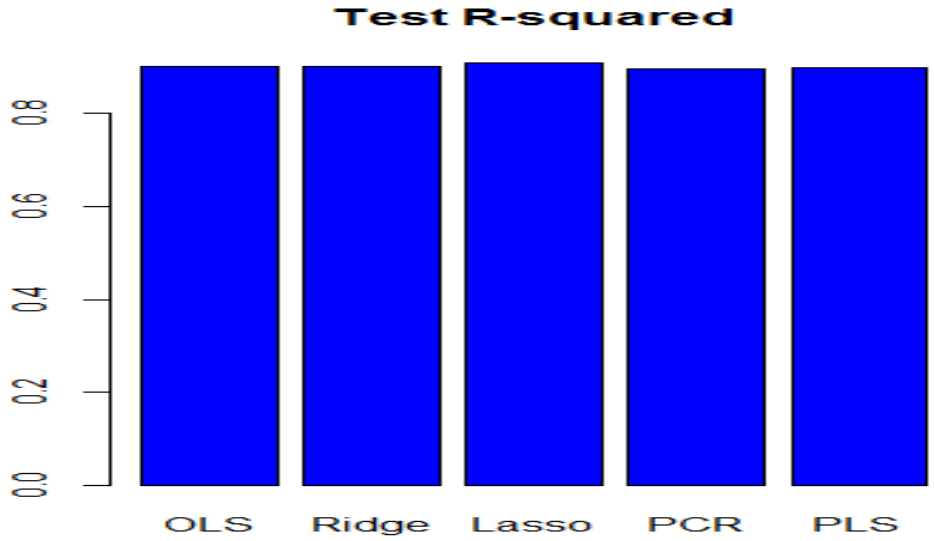
Explanation:
Obtained test error is 1134531

(g)

R Script:

```
test.avg = mean(myData.test[, "Apps"])
lm.test.r2 = 1 - mean((myData.test[, "Apps"] - lm.pred)^2) /mean((myData.test[, "Apps"] - test.avg)^2)
ridge.test.r2 = 1 - mean((myData.test[, "Apps"] - ridge.pred)^2) /mean((myData.test[, "Apps"] - test.avg)^2)
lasso.test.r2 = 1 - mean((myData.test[, "Apps"] - lasso.pred)^2) /mean((myData.test[, "Apps"] - test.avg)^2)
pcr.test.r2 = 1 - mean((myData.test[, "Apps"] - data.frame(pcr.pred))^2) /mean((myData.test[, "Apps"] - test.avg)^2)
pls.test.r2 = 1 - mean((myData.test[, "Apps"] - data.frame(pls.pred))^2) /mean((myData.test[, "Apps"] - test.avg)^2)
barplot(c(lm.test.r2, ridge.test.r2, lasso.test.r2, pcr.test.r2, pls.test.r2), col="blue", names.arg=c("OLS", "Ridge", "Lasso", "PCR", "PLS"), main="Test R-squared")
```

Output:



Explanation:

- Test R^2 is close to 0.9 for all the models
- Lasso has slightly higher test R^2 than others
- PCR uses more number of components (16) to give the same result as PLS with less number of components (10). Therefore, for this dataset PLS is preferable over PCR

```
########################
#  Chapter 6: Question 11  #
########################

rm(list = ls())
```

(a)

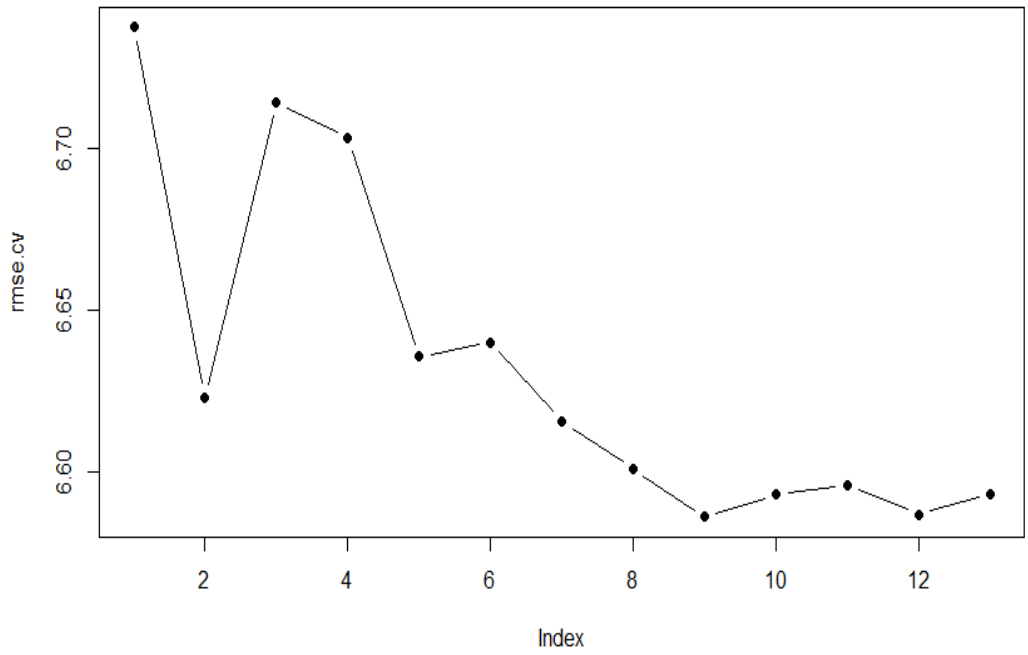
R Script:

```
set.seed(1)
library(MASS)
library(leaps)
library(glmnet)
library(pls)

myData = Boston
attach(myData)

# Best subset selection

predict.regsubsets = function(object, newdata, id, ...) {
  form = as.formula(object$call[[2]])
  mat = model.matrix(form, newdata)
  coefi = coef(object, id = id)
  nvars = names(coefi)
  mat[, nvars] %*% coefi
}

k = 10
p = ncol(myData)-1
folds = sample(rep(1:k, length=nrow(myData)))
cv.errors = matrix(NA, k, ncol(myData)-1)
for (i in 1:k) {
  best.fit = regsubsets(crim~., data=Boston[folds!=i,], nvmax=p)
  for (j in 1:p) {
    pred = predict(best.fit, Boston[folds==i, ], id=j)
    cv.errors[i,j] = mean((Boston$crim[folds==i] - pred)^2)
  }
}
rmse.cv = sqrt(apply(cv.errors, 2, mean))
plot(rmse.cv, pch=19, type="b")
which.min(rmse.cv)
rmse.cv[which.min(rmse.cv)]


# Lasso
x = model.matrix(crim~.-1, data=myData)
```

```
y = crim
cv.lasso = cv.glmnet(x, y, type.measure="mse")
plot(cv.lasso)
coef(cv.lasso)
sqrt(cv.lasso$cvm[cv.lasso$lambda == cv.lasso$lambda.1se])

# Ridge
x = model.matrix(crim~.-1, data=myData)
y = crim
cv.ridge = cv.glmnet(x, y, type.measure="mse", alpha=0)
plot(cv.ridge)
coef(cv.ridge)
sqrt(cv.ridge$cvm[cv.ridge$lambda == cv.ridge$lambda.1se])

# PCR
pcr.fit = pcr(crim~., data=myData, scale=TRUE, validation="CV")
validationplot(pcr.fit, val.type="MSEP")
summary(pcr.fit)
```

Output:

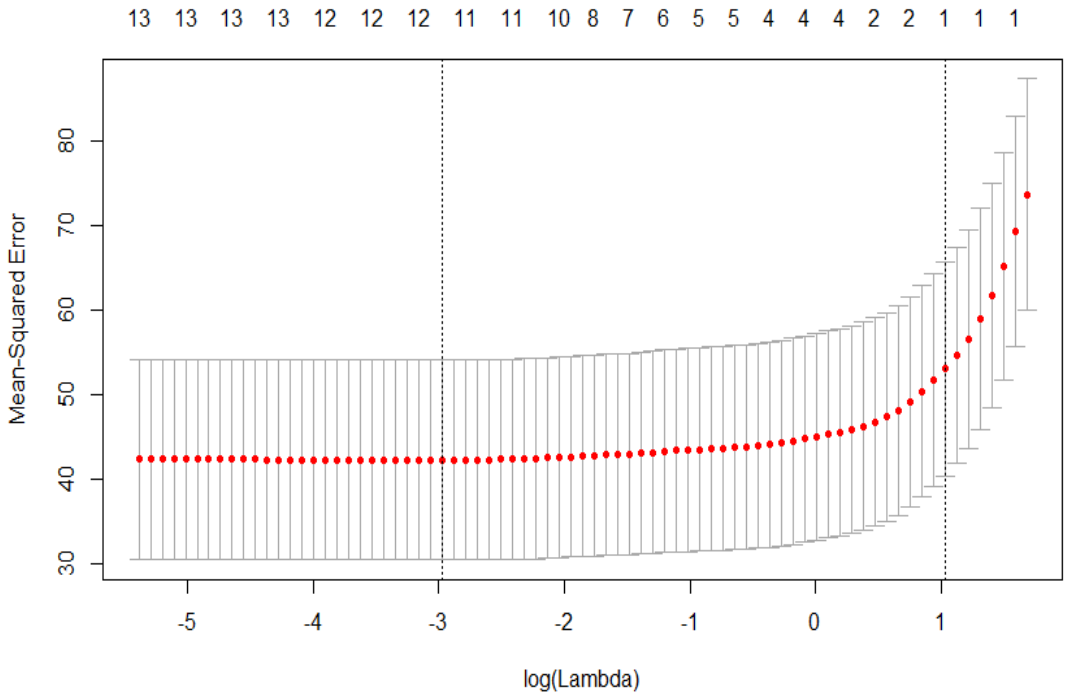
Subset Selection:




```
[1] 9
```

```
[1] 6.586008
```

Lasso:

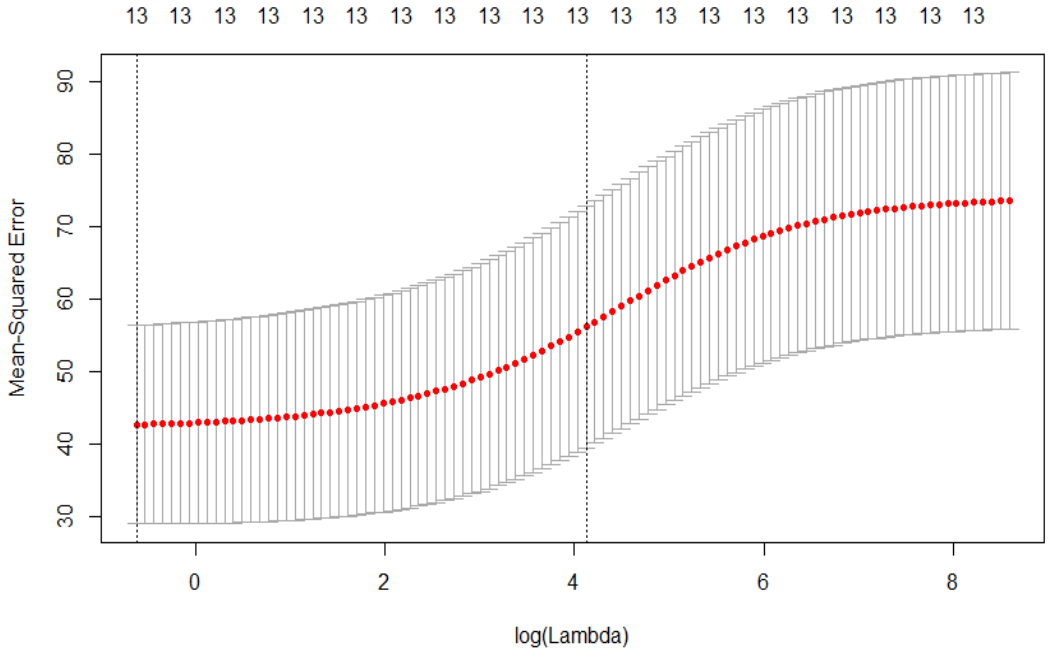


14 x 1 sparse Matrix of class "dgCMatrix"

	1
(Intercept)	0.7894616
zn	.
indus	.
chas	.
nox	.
rm	.
age	.
dis	.
rad	0.2957317
tax	.
ptratio	.
black	.
lstat	.
medv	.

[1] 7.281141

Ridge:

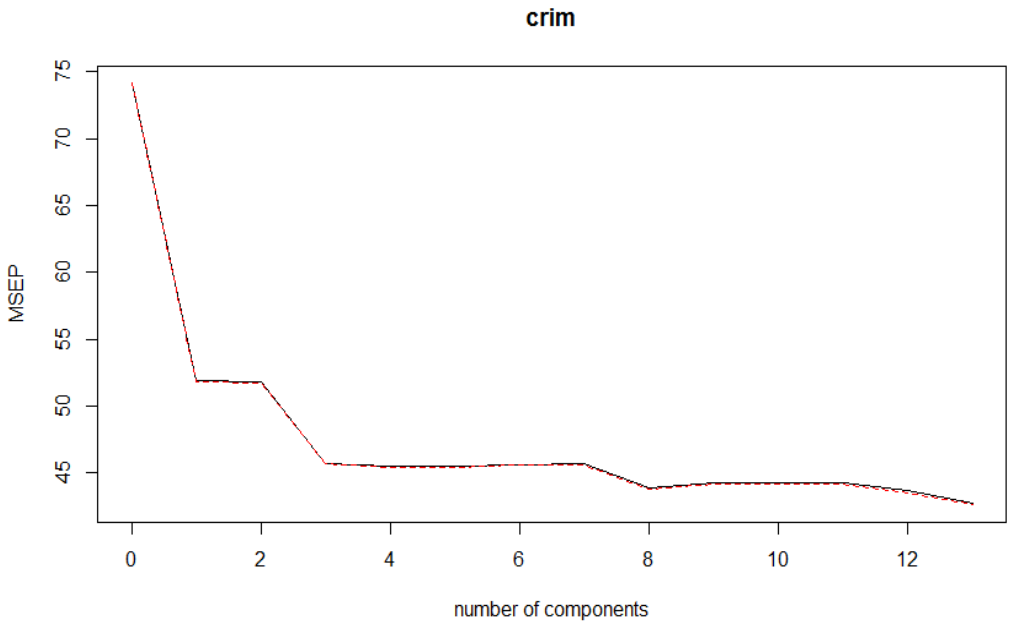


14 x 1 sparse Matrix of class "dgCMatrix"

	1
(Intercept)	1.146730398
zn	-0.002889389
indus	0.033208330
chas	-0.209288622
nox	2.158437385
rm	-0.158326514
age	0.007072375
dis	-0.109819199
rad	0.056100087
tax	0.002508948
ptratio	0.082673533
black	-0.003147919
lstat	0.042243863
medv	-0.027678989

[1] 7.494297

PCR:



Explanation:

- Best Subset model and best PCR model has comparable cross validated RMSE
- Lasso and Ridge have comparatively higher cross validated RMSE

(b)

Explanation:

- As mentioned in the answer of part (a) the 9 component best subset model has the best cross-validated RMSE.
- PCR with 13 components model has comparable cross validated RMSE. Since 9 components model is simpler that 13 components model, I would recommend 9 component subset model for this dataset

(c)

Explanation:

Refer part (b)

```
#########################
#  Chapter 4: Question 10  #
#########################

rm(list = ls())

library(ISLR)
myData = Weekly
attach(Weekly)
```

(a)

R Script:

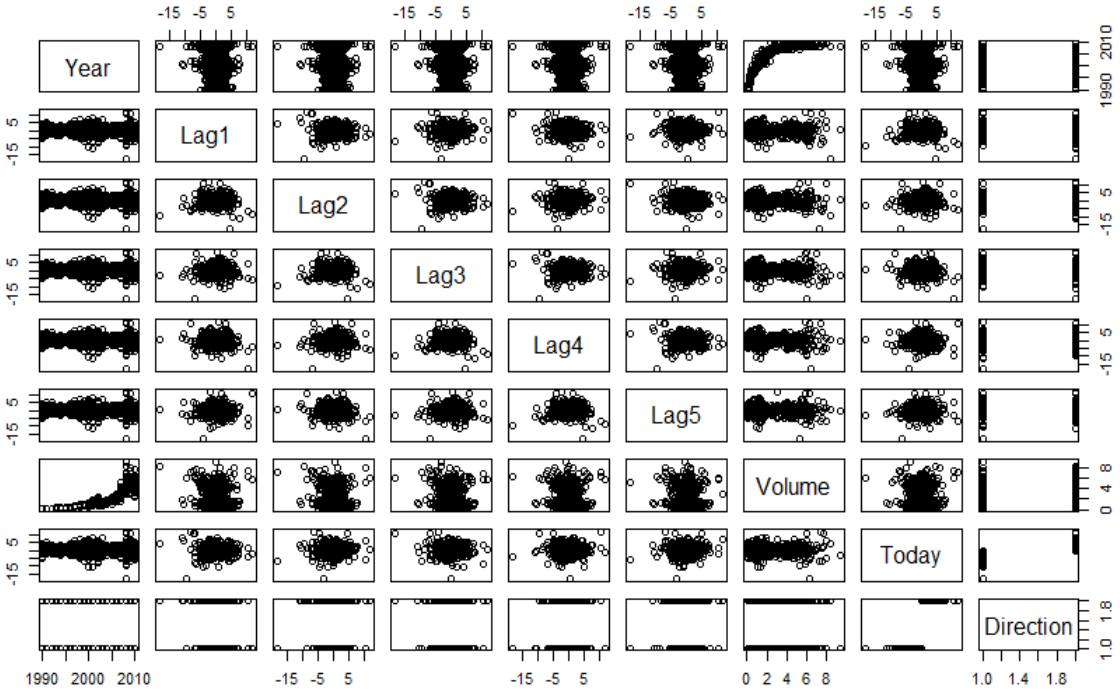
```
summary(myData)
cor(myData[,-9])
pairs(myData)
```

Output:

Year	Lag1	Lag2	Lag3	Lag4	Lag5
Min. :1990	Min. :-18.1950	Min. :-18.1950	Min. :-18.1950	Min. :-18.1950	Min. :-18.1950
1st Qu.:1995	1st Qu.: -1.1540	1st Qu.: -1.1540	1st Qu.: -1.1580	1st Qu.: -1.1580	1st Qu.: -1.1660
Median :2000	Median : 0.2410	Median : 0.2410	Median : 0.2410	Median : 0.2380	Median : 0.2340
Mean :2000	Mean : 0.1506	Mean : 0.1511	Mean : 0.1472	Mean : 0.1458	Mean : 0.1399
3rd Qu.:2005	3rd Qu.: 1.4050	3rd Qu.: 1.4090	3rd Qu.: 1.4090	3rd Qu.: 1.4090	3rd Qu.: 1.4050
Max. :2010	Max. : 12.0260	Max. : 12.0260	Max. : 12.0260	Max. : 12.0260	Max. : 12.0260

Volume	Today	Direction
Min. :0.08747	Min. :-18.1950	Down:484
1st Qu.:0.33202	1st Qu.: -1.1540	Up :605
Median :1.00268	Median : 0.2410	
Mean :1.57462	Mean : 0.1499	
3rd Qu.:2.05373	3rd Qu.: 1.4050	
Max. :9.32821	Max. : 12.0260	

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today
Year	1.00000000	-0.032289274	-0.03339001	-0.03000649	-0.031127923	-0.030519101	0.84194162	-0.032459894
Lag1	-0.03228927	1.000000000	-0.07485305	0.05863568	-0.071273876	-0.008183096	-0.06495131	-0.075031842
Lag2	-0.03339001	-0.074853051	1.00000000	-0.07572091	0.058381535	-0.072499482	-0.08551314	0.059166717
Lag3	-0.03000649	0.058635682	-0.07572091	1.00000000	-0.075395865	0.060657175	-0.06928771	-0.071243639
Lag4	-0.03112792	-0.071273876	0.05838153	-0.07539587	1.000000000	-0.075675027	-0.06107462	-0.007825873
Lag5	-0.03051910	-0.008183096	-0.07249948	0.06065717	-0.075675027	1.000000000	-0.05851741	0.011012698
Volume	0.84194162	-0.064951313	-0.08551314	-0.06928771	-0.061074617	-0.058517414	1.00000000	-0.033077783
Today	-0.03245989	-0.075031842	0.05916672	-0.07124364	-0.007825873	0.011012698	-0.03307778	1.000000000



Explanation:

- Year and Volume are related
- No other pattern can be identified

(b)

R Script:

```
glm.fit = glm(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data = myData, family = binomial)
summary(glm.fit)
```

Output:

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = myData)

Deviance Residuals:
    Min      1Q  Median      3Q     Max
-1.6949  -1.2565  0.9913  1.0849  1.4579

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1486.4  on 1082  degrees of freedom
AIC: 1500.4

Number of Fisher Scoring iterations: 4
```

Explanation:

- Lag2 has some statistical significance

(c)

R Script:

```
glm.probs = predict(glm.fit, type = "response")

glm.pred = rep("Down", length(glm.probs))
glm.pred[glm.probs > .5] = "Up"
table(glm.pred, myData$Direction)
```

Output:

glm.pred	Down	Up
Down	54	48
Up	430	557

Explanation:

- %age of correct predictions(Overall): 56.1%
- %age of correct predictions(Market - Up): 92.1%
- %age of correct predictions(Market - Down): 11.2%

(d)

R Script:

```
train = (Year < 2009)
test = myData[!train,]

glm.fit = glm(Direction ~ Lag2, data = myData, family = binomial, subset = train)
glm.probs = predict(glm.fit, test, type = "response")

glm.pred = rep("Down", length(glm.probs))
glm.pred[glm.probs > .5] = "Up"

Dir = Direction[!train]
table(glm.pred, Dir)
mean(glm.pred == Dir)
```

Output:

	Dir	
glm.pred	Down	Up
Down	9	5
Up	34	56

```
[1] 0.625
```

(e) NA

(f) NA

(g)

R Script:

```
library(class)

train.X = as.matrix(Lag2[train])
test.X = as.matrix(Lag2[!train])
```

```
train.Direction = Direction[train]

set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k=1)

table(knn.pred, Dir)
mean(knn.pred == Dir)
```

Output:

	Dir	
knn.pred	Down	Up
Down	21	30
Up	22	31

```
[1] 0.5
```

(h)

Explanation:

- Since the overall fraction of correct prediction is higher for logistic regression, it provides better results than KNN for K = 1

(i)

R Script:

```
# Logistic regression - (Lag2:Lag1)
glm.fit = glm(Direction ~ Lag2:Lag1, data = Weekly, family = binomial, subset = train)

glm.probs = predict(glm.fit, test, type = "response")

glm.pred = rep("Down", length(glm.probs))
glm.pred[glm.probs > .5] = "Up"

Dir = Direction[!train]
table(glm.pred, Dir)
mean(glm.pred == Dir)

# KNN - (k = 10)
knn.pred = knn(train.X, test.X, train.Direction, k=10)

table(knn.pred, Dir)
mean(knn.pred == Dir)

# KNN - (k = 100)
knn.pred = knn(train.X, test.X, train.Direction, k=100)

table(knn.pred, Dir)
mean(knn.pred == Dir)
```

Output:

Logistic regression - (Lag2:Lag1):

	Dir	
glm.pred	Down	Up
Down	1	1
Up	42	60

[1] 0.5865385

KNN - (k = 10):

	Dir	
knn.pred	Down	Up
Down	17	18
Up	26	43

[1] 0.5769231

KNN - (k = 100):

	Dir	
knn.pred	Down	Up
Down	9	12
Up	34	49

[1] 0.5576923

Explanation:

- All these models have comparable performance. Logistic regression is slightly better
- KNN model having k = 100 has k almost equal to number of data points. Therefore, it will give mean as the prediction. As expected it has the worst performance among all the models tried

```
########################
#   Chapter 8: Question 8    #
########################

rm(list = ls())

library(tree)
library(randomForest)
library(ISLR)
myData = Carseats
attach(myData)
set.seed(1)
```

(a)

R Script:

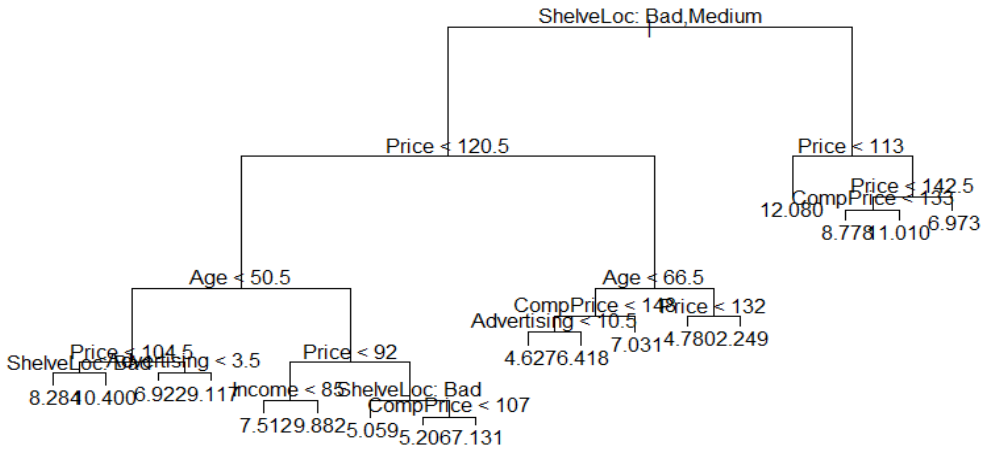
```
train = sample(dim(myData)[1], dim(myData)[1] / 2)
myData.train = myData[train, ]
myData.test = myData[-train, ]
```

(b)

R Script:

```
tree.myData = tree(Sales~., data=myData.train)
summary(tree.myData)
plot(tree.myData)
text(tree.myData, pretty=0)

pred.myData = predict(tree.myData, myData.test)
mean((myData.test$Sales - pred.myData)^2)
```

Output:



```
Regression tree:
tree(formula = Sales ~ ., data = myData.train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"         "Advertising" "Income"      "CompPrice"
Number of terminal nodes:  18
Residual mean deviance:  2.36 = 429.5 / 182
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-4.2570 -1.0360  0.1024  0.0000  0.9301  3.9130
[1] 4.148897
```

Explanation:

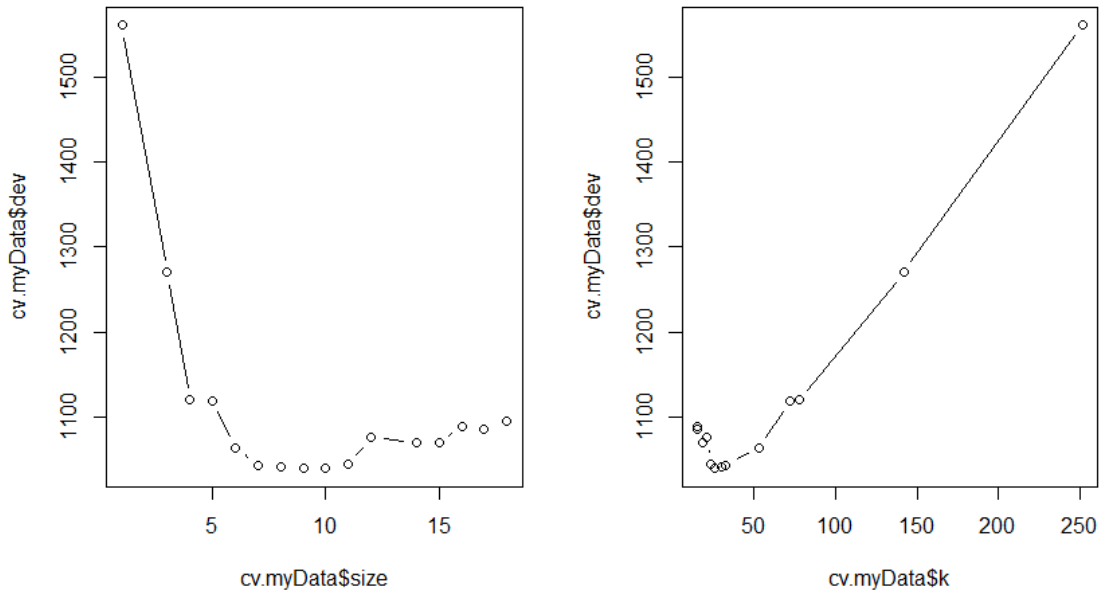
- Obtained test error rate is 4.148897

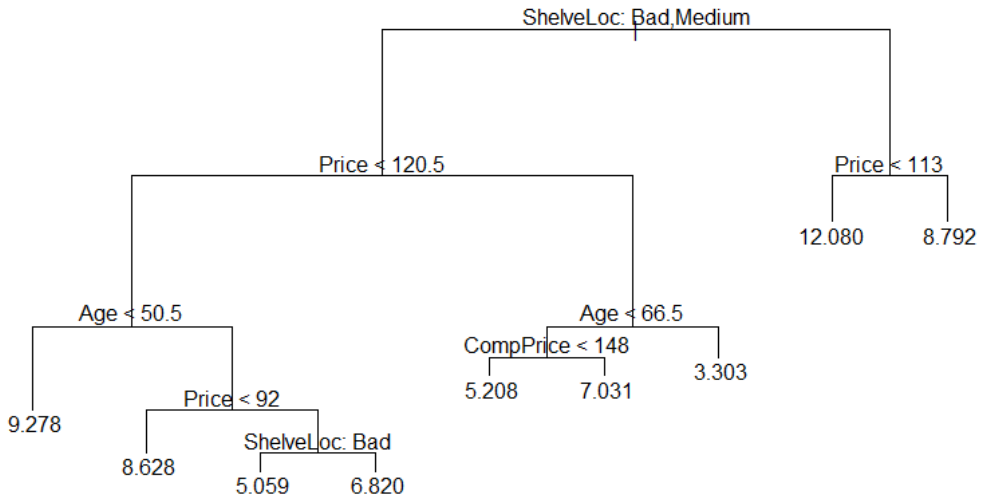
(c)

R Script:

```
cv.myData = cv.tree(tree.myData, FUN=prune.tree)
par(mfrow=c(1, 2))
plot(cv.myData$size, cv.myData$dev, type="b")
plot(cv.myData$k, cv.myData$dev, type="b")

pruned.myData = prune.tree(tree.myData, best=9)
par(mfrow=c(1, 1))
plot(pruned.myData)
text(pruned.myData, pretty=0)
pred.pruned = predict(pruned.myData, myData.test)
mean((myData.test$Sales - pred.pruned)^2)
```

Output:






```
[1] 4.993124
```

Explanation:

- Pruning the tree increases the test error rate to 4.993124

(d)

R Script:

```
bag.myData = randomForest(Sales~., data=myData.train, mtry=10, ntree=500,
importance=TRUE)
bag.pred = predict(bag.myData, myData.test)
mean((myData.test$Sales - bag.pred)^2)
importance(bag.myData)
```

Output:

```
[1] 2.604369
```

	%IncMSE	IncNodePurity
CompPrice	14.4124562	133.731797
Income	6.5147532	74.346961
Advertising	15.7607104	117.822651
Population	0.6031237	60.227867
Price	57.8206926	514.802084
ShelveLoc	43.0486065	319.117972
Age	19.8789659	192.880596
Education	2.9319161	39.490093
Urban	-3.1300102	8.695529
US	7.6298722	15.723975

Explanation:

- The test error rate improved to 2.604369

- According to importance() function, Price and ShelveLoc are the important parameters

(e)

R Script:

```
rf.myData = randomForest(Sales~., data=myData.train, mtry=5, ntree=500, importance=TRUE)
rf.pred = predict(rf.myData, myData.test)
mean((myData.test$Sales - rf.pred)^2)
importance(rf.myData)
```

Output:

```
[1] 2.802383
```

	%IncMSE	IncNodePurity
CompPrice	12.0259791	124.81403
Income	5.5542673	106.15418
Advertising	12.0466048	136.15204
Population	0.3136897	81.68162
Price	45.9639857	457.15711
ShelveLoc	36.2789679	271.76488
Age	20.8537727	196.72182
Education	2.9005332	54.16980
Urban	-0.6888196	11.86848
US	6.9739759	23.64075

Explanation:

- Obtained test error rate is 2.802383
- According to importance() function, Price and ShelveLoc are the important parameters

```
#######################
#   Chapter 8: Question 11  #
#######################

rm(list = ls())

library(gbm)
library(ISLR)
myData = Caravan
attach(myData)
set.seed(1)
```

(a)

R Script:

```
train = 1:1000
myData$Purchase = ifelse(myData$Purchase == "Yes", 1, 0)

myData.train = myData[train, ]
myData.test = myData[-train, ]
```

(b)

R Script:

```
boost.myData = gbm(Purchase~., data=myData.train, n.trees=1000, shrinkage=0.01,
distribution="bernoulli")
summary(boost.myData)
```

Output:

	var	rel.inf
PPERSAUT	PPERSAUT	14.63504779
MKOOPKLA	MKOOPKLA	9.47091649
MOPLHOOG	MOPLHOOG	7.31457416
MBERMIDD	MBERMIDD	6.08651965
PBRAND	PBRAND	4.66766122
MGODGE	MGODGE	4.49463264
ABRAND	ABRAND	4.32427755
MINK3045	MINK3045	4.17590619
MOSTYPE	MOSTYPE	2.86402583
PWAPART	PWAPART	2.78191075
MAUT1	MAUT1	2.61929152
MBERARBG	MBERARBG	2.10480508
MSKA	MSKA	2.10185152
MAUT2	MAUT2	2.02172510
MSKC	MSKC	1.98684345
MINKGEM	MINKGEM	1.92122708
MGODPR	MGODPR	1.91777542
MBERHOOG	MBERHOOG	1.80710618
MGODOV	MGODOV	1.78693913
PBYSTAND	PBYSTAND	1.57279593
MSKB1	MSKB1	1.43551401
MFWEKIND	MFWEKIND	1.37264255
MRELGE	MRELGE	1.20805179
MOPLMIDD	MOPLMIDD	0.93791970
MINK7512	MINK7512	0.92590720
MINK4575	MINK4575	0.91745993
MGODRK	MGODRK	0.90765539
MFGEKIND	MFGEKIND	0.85745374
MZPART	MZPART	0.82531066
MRELOV	MRELOV	0.80731252
MINKM30	MINKM30	0.74126812
MHKOOP	MHKOOP	0.73690793
MZFONDS	MZFONDS	0.71638323
MAUT0	MAUT0	0.71388052
MHHUUR	MHHUUR	0.59287247
APERSAUT	APERSAUT	0.58056986
MOSHOOFD	MOSHOOFD	0.58029563
MSKB2	MSKB2	0.53885275
PLEVEN	PLEVEN	0.53052444
MINK123M	MINK123M	0.50660603
MBERARBO	MBERARBO	0.48596479
MGEMOMV	MGEMOMV	0.47614792
PMOTSCO	PMOTSCO	0.46163590
MSKD	MSKD	0.39735297
MBERBOER	MBERBOER	0.36417546
MGEMLEEF	MGEMLEEF	0.26166240
MFALLEEN	MFALLEEN	0.21448118
MBERZELF	MBERZELF	0.15906143
MOPLLAAG	MOPLLAAG	0.05263665
MAANTHUI	MAANTHUI	0.03766014
MRELSA	MRELSA	0.00000000
PWABEDR	PWABEDR	0.00000000
PWALAND	PWALAND	0.00000000
PBESAUT	PBESAUT	0.00000000
PVRAAUT	PVRAAUT	0.00000000
PAANHANG	PAANHANG	0.00000000
PTRACTOR	PTRACTOR	0.00000000
PWERKT	PWERKT	0.00000000
PBROM	PBROM	0.00000000
PPERSONG	PPERSONG	0.00000000
PGEZONG	PGEZONG	0.00000000
PWAOREG	PWAOREG	0.00000000
PZEILPL	PZEILPL	0.00000000
PPLEZIER	PPLEZIER	0.00000000
PFIETS	PFIETS	0.00000000
PINBOED	PINBOED	0.00000000
AWAPART	AWAPART	0.00000000
AWABEDR	AWABEDR	0.00000000
AWALAND	AWALAND	0.00000000
ABESAUT	ABESAUT	0.00000000
AMOTSCO	AMOTSCO	0.00000000
AVRAAUT	AVRAAUT	0.00000000
AAANHANG	AAANHANG	0.00000000
ATRACTOR	ATRACTOR	0.00000000
AWERKT	AWERKT	0.00000000
ABROM	ABROM	0.00000000
ALEVEN	ALEVEN	0.00000000
APERSONG	APERSONG	0.00000000
AGEZONG	AGEZONG	0.00000000
AWAOREG	AWAOREG	0.00000000
AZEILPL	AZEILPL	0.00000000
APLEZIER	APLEZIER	0.00000000
AFIETS	AFIETS	0.00000000
AINBOED	AINBOED	0.00000000
ABYSTAND	ABYSTAND	0.00000000

Explanation:

- PPERSAUT, MKOOPKLA, MOPLHOOG and MBERMIDD are most important predictors

(c)

R Script:

```
boost.prob = predict(boost.myData, myData.test, n.trees=1000, type="response")
boost.pred = ifelse(boost.prob > 0.2, 1, 0)
table(myData.test$Purchase, boost.pred)

lm.myData = glm(Purchase~., data=myData.train, family=binomial)
lm.prob = predict(lm.myData, myData.test, type="response")
lm.pred = ifelse(lm.prob > 0.2, 1, 0)
table(myData.test$Purchase, lm.pred)
```

Output:

boost.pred

	0	1
0	4410	123
1	256	33

lm.pred

	0	1
0	4183	350
1	231	58

Explanation:

- Boosting: 21.15% Prediction Accuracy
- Logistic Regression: 14.21% Prediction Accuracy
- Therefore, Boosting is better than logistic regression

Problem 1: Beauty Pays!

R Script:

```
beauty_data = read.csv("C:/Users/Neerav Basant/Desktop/Summer/Predictive Modeling/Part 1/BeautyData.csv")
attach(beauty_data)

lm.fit = lm(CourseEvals~., data = beauty_data)

summary(lm.fit)
```

1.

Explanation:

- We need to estimate the effect of "beauty" into course ratings. But as the question suggests, in order to measure the effect of "beauty", we also "need to adjust for many other determinants". We can do that by building a linear model using course rating as response variable and all the other relevant variables as predictor variable.
- In the given dataset (i.e. BeautyData.csv), CourseEvals is dependent/response variable; and BeautyScore, Female, Lower, NonEnglish and TenureTrack are independent/predictor variables.
  $CourseEvals = \beta_0 + \beta_1 BeautyScore + \beta_2 Female + \beta_3 Lower + \beta_4 NonEnglish + \beta_5 TenureTrack + \epsilon,$
  $\epsilon \sim N(0,\sigma^2)$

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.31385 -0.30202  0.01011  0.29815  1.04929
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.06542    0.05145  79.020  < 2e-16 ***
BeautyScore  0.30415    0.02543  11.959  < 2e-16 ***
female      -0.33199    0.04075  -8.146 3.62e-15 ***
lower       -0.34255    0.04282  -7.999 1.04e-14 ***
nonenglish  -0.25808    0.08478  -3.044  0.00247 **
tenuretrack -0.09945    0.04888  -2.035  0.04245 *
```

```
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.4273 on 457 degrees of freedom
Multiple R-squared:  0.3471,    Adjusted R-squared:  0.3399
F-statistic: 48.58 on 5 and 457 DF,  p-value: < 2.2e-16
```

- We will analyze the results by considering one variable at a time. BeautyScore has a positive coefficient which suggests that keeping all the other parameters constant, higher the beauty score of the instructor the higher his/her rating.

- On the contrary, holding everything else constant female instructors receive lower ratings than male instructor. We all know that gender has no influence on the teaching capability of instructor. This clearly suggests that UT students has a potential bias towards male instructors.
- Similarly, lower classes have a negative coefficient. Students are mostly less interested in lower division classes (because it is mandatory) which could be the reason for giving less grades to instructors of the respective classes.
- Negative coefficient for Non English instructors is pretty intuitive. A non-native speaker might face difficulty in communicating in English which could be possible reason for lower evaluation of instructors.
- Coefficient of Tenure track is small (and negative). Therefore, it has minimal impact on course rating of the instructors.

2.

Explanation:

- Dr. Hamermesh is basically posing a very simple question here: "Are beautiful instructors actually better teachers or is it just a perception?" Result of linear regression cannot answer this question.
- In my opinion, beauty like gender does not have any impact on capability to teach. This is clearly a case of bias on the part of students. But unfortunately, we cannot prove this with this model/experiment.
- We can conduct various controlled experiments to prove this point. One such experiment could be conducting audio lectures so that instructor's beauty would not have any impact on the perception of students.

Problem 2: Housing Price Structure

R Script:

```
Housing_Data = read.csv("C:/Users/Neerav Basant/Desktop/Summer/Predictive Modeling/Part 1/MidCity.csv")
attach(Housing_Data)

n = dim(Housing_Data)[1]

dn1 = rep(0,n)
dn1[Nbhd==1]=1

dn2 = rep(0,n)
dn2[Nbhd==2]=1

dn3 = rep(0,n)
dn3[Nbhd==3]=1

BR = rep(0,n)
BR[Brick=="Yes"]=1

Price = Price/1000
SqFt = SqFt/1000

MidCityModel = lm(Price~BR+dn2+dn3+SqFt+Offers+SqFt+Bedrooms+Bathrooms)

summary(MidCityModel)
confint(MidCityModel)

model2 = lm(Price~BR+dn2+dn3+SqFt+Offers+SqFt+Bedrooms+Bathrooms + BR:dn3)

summary(model2)
confint(model2)
confint(model2, level = 0.99)
```

1. **Explanation:**
   - As per discussion in the class, we need to create dummy variable N_1, N_2 and N_3 to indicate if a house is from neighborhood 1, 2 or 3. We can also create a dummy variable, say Brick to indicate if a house is made of Brick.
   - We can build linear regression model, considering Price as response variable and other relevant variables from the dataset and dummy variable as predictor variable.

     $Price = \beta_0 + \beta_1 BR + \beta_2 dn2 + \beta_3 dn3 + \beta_4 Offers + \beta_5 SqFt + \beta_6 Bedrooms + \beta_7 Bathrooms + \epsilon, \epsilon \sim N(0,\sigma^2)$

Note: Only 2 neighborhoods (dn2 and dn3) are passed as the predictor variable. For dn2 = dn3 = 0, we can predict the values for dn1.

- Following is the summary and Confidence interval output based on the linear regression model output:

```
Residuals:
     Min       1Q   Median       3Q      Max
-27337.3  -6549.5    -41.7   5803.4  27359.3

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2159.498   8877.810   0.243  0.80823
BR          17297.350   1981.616   8.729 1.78e-14 ***
dn2         -1560.579   2396.765  -0.651  0.51621
dn3         20681.037   3148.954   6.568 1.38e-09 ***
SqFt           52.994      5.734   9.242 1.10e-15 ***
Offers      -8267.488   1084.777  -7.621 6.47e-12 ***
Bedrooms     4246.794   1597.911   2.658  0.00894 **
Bathrooms    7883.278   2117.035   3.724  0.00030 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10020 on 120 degrees of freedom
Multiple R-squared:  0.8686,    Adjusted R-squared:  0.861
F-statistic: 113.3 on 7 and 120 DF,  p-value: < 2.2e-16
```

Confidence Interval:

```
                  2.5 %      97.5 %
(Intercept) -15417.94711 19736.94349
BR           13373.88702 21220.81203
dn2          -6306.00785  3184.84961
dn3          14446.32799 26915.74671
SqFt            41.64034    64.34714
Offers      -10415.27089 -6119.70575
Bedrooms      1083.04162  7410.54616
Bathrooms     3691.69572 12074.86126
```

- β_1 = 17297.350
- To find the premium associated with brick houses everything else being equal, we need to test the null hypothesis of β_1=0. From the 95% confidence interval table, it is quite obvious that 0 cannot be a part of confidence interval of β_1 (13373.89 – 21220.81). Therefore, null hypothesis of β_1 = 0 is rejected.
- So, brick is significant factor in determining the price of a house. Moreover, the parameter estimate is positive. Therefore, we conclude that people pay a premium for brick house.

2.

Explanation:

- $\beta_3 = 20681.037$
- To find the premium associated with houses in neighborhood 3 everything else being equal, we need to test the null hypothesis of $\beta_3=0$. From the 95% confidence interval table, it is quite obvious that 0 cannot be a part of confidence interval of β_3 (14446.33 – 26915.75). Therefore, null hypothesis of $\beta_3 = 0$ is rejected.
- Since the parameter estimate is positive and entire confidence interval is greater than 0, we can conclude that people pay a premium to live in neighborhood 3.

3.

Explanation:

- We need to check the interaction of Brick and neighborhood 3 to figure out the premium associated with brick houses in neighborhood 3. In order to do that, we need to add an interaction term (BR * dn3) to our model:
  $Price = \beta_0 + \beta_1 BR + \beta_2 dn2 + \beta_3 dn3 + \beta_4 Offers + \beta_5 SqFt + \beta_6 Bedrooms + \beta_7 Bathrooms + \beta_8 BR.dn3 + \epsilon, \epsilon \sim N(0,\sigma^2)$
- Following is the summary and Confidence interval output (95% and 99%)based on the linear regression model output:

```
Residuals:
     Min       1Q   Median       3Q      Max
-26939.1  -5428.7   -213.9   4519.3  26211.4

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3009.993   8706.264   0.346  0.73016
BR          13826.465   2405.556   5.748 7.11e-08 ***
dn2          -673.028   2376.477  -0.283  0.77751
dn3         17241.413   3391.347   5.084 1.39e-06 ***
SqFt           54.065      5.636   9.593  < 2e-16 ***
Offers      -8401.088   1064.370  -7.893 1.62e-12 ***
Bedrooms     4718.163   1577.613   2.991  0.00338 **
Bathrooms    6463.365   2154.264   3.000  0.00329 **
BR:dn3      10181.577   4165.274   2.444  0.01598 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9817 on 119 degrees of freedom
Multiple R-squared:  0.8749,    Adjusted R-squared:  0.8665
F-statistic:   104 on 8 and 119 DF,  p-value: < 2.2e-16
```

Confidence Interval (95%):

	2.5 %	97.5 %
(Intercept)	-14229.27947	20249.26635
BR	9063.22323	18589.70668
dn2	-5378.69058	4032.63406
dn3	10526.20666	23956.61921
SqFt	42.90493	65.22463
Offers	-10508.64698	-6293.52887
Bedrooms	1594.33302	7841.99385
Bathrooms	2197.70794	10729.02197
BR:dn3	1933.91810	18429.23657

Confidence Interval (99%):

	0.5 %	99.5 %
(Intercept)	-19781.05615	25801.04303
BR	7529.25747	20123.67244
dn2	-6894.11333	5548.05681
dn3	8363.62557	26119.20030
SqFt	39.31099	68.81858
Offers	-11187.37034	-5614.80551
Bedrooms	588.32720	8847.99967
Bathrooms	823.98555	12102.74436
BR:dn3	-722.17781	21085.33248

- The 95% confidence interval table does not include 0 for β_8 (1933.92 – 18429.24). This implies that there is a premium for brick houses in neighborhood three at 95% confidence level.
- However, if we look at 99% confidence interval table, it includes 0 for β_8 (-722.18 – 21085.33). Therefore, in a stricter environment, it can be concluded that there is no premium for brick houses in neighborhood three.

4.

Explanation:

- We need to determine that is it reasonable to combine N1 and N2 for the purposes of prediction. If either of them is not significant then it can be concluded that it is reasonable to combine them.
- B_2 is the coefficient for N2. If it is insignificant, then our null hypothesis $\beta_2 = 0$ should be true.
- From the 95% confidence interval table (from 1st part of the question), it is evident that 0 is a possible value for β_2 (-6306.01 – 3184.85). Thus, we can conclude that it is reasonable to make β_2 as 0.
- Hence, N1 and N2 can be combined for the purpose of prediction.

Problem 3: What causes what??

1.

Explanation:

- In an ideal world, we would like to have less crime in our cities. But if there is less crime, there is no need for more police. So, if there is more police in a city, it might mean that crime rates in that city are high. So, we expect to see positive correlation between "crime" and "police".
- Datasets having "crime" and "police" details cannot differentiate between more crime leading to more police or more police leading to more crime. Therefore, we cannot understand how more cops affect crime by having this dataset from different cities. We need to perform more natural experiment to understand this problem.

2.

Explanation:

- Researchers from UPENN collected better and more natural data to solve this problem.
- They collected crime data for DC and also identified the days in which there was a higher alert due to information about potential terrorist attacks. Important thing to note here is crime had nothing to do with number of police on a particular day.
- From table 2, we can see that the coefficient for dummy variable of high alert is negative. This suggests that the crime rate was lower on such days.

3.

Explanation:

- We need to control the ridership to validate the fact that the people were out in spite of the news about terrorist activities. Holding the ridership constant, it was made sure that unavailability of opportunities to commit crime was not the reason for less crime on high alert days.
- Some might argue that criminals were afraid of terrorists and decided not to go out on high alert days.
- I believe this is a very strong assumption. As far as I am concerned, the results from the table strongly suggests that actual reason for reduced crime was more number of police and not anything else.

4.

Explanation:

- In table 4, researchers are calculating interaction of high alert days and various districts.
- District 1 (DC) has a lot of potential terrorist targets. As a result, number of police deployed in district 1 should be comparatively high compared to other districts.
- A more negative coefficient value for District 1 compared to other districts is in line with the above mentioned fact. Crime in District 1 reduces more on high alert days compared to other districts.