Exercise 8

Question 1

1.

```
library(boot)

carsData <- read.table('Cars2Data.txt', header = T)
carsData <- na.omit(carsData)

set.seed(13)

cars.mod <- glm(mpg ~ weight + year + origin, data = carsData)

cars.mod1 <- glm(mpg ~ weight + year + origin + horsepower, data = carsData)

cars.poly <- glm(mpg ~ poly(weight + year + origin, 2), data=carsData)
```

2.

```
set.seed(13)
cv1 <- cv.glm(carsData, cars.mod, K=10)
cv2 <- cv.glm(carsData, cars.mod1, K=10)

cv.error <- cv.glm(carsData, cars.mod, K=10)$delta[1]
cv1.error <- cv.glm(carsData, cars.mod1, K=10)$delta[1]
cvp.error <- cv.glm(carsData, cars.poly, K=10)$delta[1]

cat("Model 1 cv.error =", cv.error, "\n")
```

```
## Model 1 cv.error = 11.3021
```

```
cat("Model 2 cv.error = ", cv1.error, "\n")
```

```
## Model 2 cv.error =  11.2336
```

```
cat("Model poly cv.error = ", cvp.error, "\n")
```

```
## Model poly cv.error =  17.74012
```

Question 2

1.

```
library(boot)
set.seed(3)
cancerData <- read.table('Cancer.txt', header = T)
cancerData <- cancerData[c('Diagnostic','Texture','Perimeter','Symmetry','Smooth')]
cancer.mod <- glm(Diagnostic ~ ., data = cancerData, family = binomial)
summary(cancer.mod)
```

```
##
## Call:
## glm(formula = Diagnostic ~ ., family = binomial, data = cancerData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0878  -0.1666  -0.0351   0.0339   3.1702
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -42.37228    4.60847  -9.194  < 2e-16 ***
## Texture       0.37778    0.05885   6.420 1.37e-10 ***
## Perimeter     0.20893    0.02351   8.889  < 2e-16 ***
## Symmetry     24.34436   10.11512   2.407   0.0161 *
## Smooth      109.24348   20.68290   5.282 1.28e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 175.72  on 564  degrees of freedom
## AIC: 185.72
##
## Number of Fisher Scoring iterations: 8
```

Coefficient estimates are statistically different from 0 because $\mathbf{Pr(>|z|)}$ are very small($<<0.05$).

The iterative computation was done after 8 iterations.

2.

```
cv.error <- cv.glm(cancerData, cancer.mod, K=10)$delta[1]
cat("Error rate(test error) =", cv.error)
```

```
## Error rate(test error) = 0.04958319
```

Question 3

1.

```
library(boot)
set.seed(13)
vertebralData <- read.table('VertebralData.2C.txt', sep = ",", header=T)

cor(vertebralData[-7])
```

```
##            Incidence       Tilt       Angle       Slope      Radius
## Incidence  1.0000000 0.62919877  0.71728236  0.81495999 -0.24746721
## Tilt       0.6291988 1.00000000  0.43276386  0.06234529  0.03266781
## Angle      0.7172824 0.43276386  1.00000000  0.59838689 -0.08034361
## Slope      0.8149600 0.06234529  0.59838689  1.00000000 -0.34212835
## Radius    -0.2474672 0.03266781 -0.08034361 -0.34212835  1.00000000
## Degree     0.6387427 0.39786228  0.53366701  0.52355746 -0.02606501
##                Degree
## Incidence  0.63874275
## Tilt       0.39786228
## Angle      0.53366701
## Slope      0.52355746
## Radius    -0.02606501
## Degree     1.00000000
```

```
vertebral.mod <- glm(Status ~ Incidence + Radius + Tilt + Angle, data=vertebralData, family=binomial)
```

Logistic regression is unstable when the classes are well seperated So if we include all the variables we will get an error (**glm.fit: fitted probabilities numerically 0 or 1 occurred**) which I think is an over-fitting problem: too many variables in our model, leading to a perfect separation of the cases.

So, we tried to exclude the correlated variables, by finding pairs of predictors that are correlated.

2.

```
library(MASS)
set.seed(13)
vertebral.lda.mod <- lda(Status ~ Incidence + Radius + Tilt + Angle, data=vertebralData)
vertebral.lda.mod
```

```
## Call:
## lda(Status ~ Incidence + Radius + Tilt + Angle, data = vertebralData)
##
## Prior probabilities of groups:
##  Abnormal    Normal
## 0.6774194 0.3225806
##
## Group means:
##          Incidence   Radius     Tilt    Angle
## Abnormal  64.69256 115.0777 19.79111 55.92537
## Normal    51.68524 123.8908 12.82141 43.54260
##
## Coefficients of linear discriminants:
##                    LD1
```

```
## Incidence  0.005006361
## Radius     0.055267493
## Tilt      -0.064782560
## Angle     -0.024722500
```

3. To compare the two models we will compute test error rate (LOOCV) for both models:

```
library(MASS)
set.seed(13)
#CV=T then it performs LOOCV
vertebral.lda.mod <- lda(Status ~ Incidence + Radius + Tilt + Angle, data=vertebralData, CV=T)
cat("\nLDA Test Error rate = ", mean(vertebral.lda.mod$class != vertebralData$Status))
```

```
##
## LDA Test Error rate =  0.2290323
```

```
#if we don't specify K then by default is K=n - LOOCV
cv.error <- cv.glm(vertebralData, vertebral.mod)$delta[1]
cat("\nLogistic reg. Test Error rate =", cv.error,"\n")
```

```
##
## Logistic reg. Test Error rate = 0.1619202
```

So, we see that even why the two models LDA and Logistic regression are very similar and we expect that they accuracy will be also very similar sometimes this is not the case. LDA assumes that the observations are drawn from a Gaussian distribution with a common covariance matrix in each class and can provide improvements over Logistic regression when that assumption holds, but it also will be outperformed when the assumption does not hold.