1: The First Problem

(a) Logistics model Output:

```
Call:
glm(formula = type ~ npreg + glu + bp + skin + bmi + ped + age,
    family = binomial, data = pima1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0100  -0.6613  -0.3692   0.6433   2.4795

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.554651   0.994217  -9.610  < 2e-16 ***
npreg        0.122517   0.043743   2.801 0.005097 **
glu          0.035321   0.004244   8.322  < 2e-16 ***
bp          -0.007695   0.010314  -0.746 0.455602
skin         0.006774   0.014759   0.459 0.646242
bmi          0.082678   0.023334   3.543 0.000395 ***
ped          1.308708   0.364040   3.595 0.000324 ***
age          0.026375   0.014000   1.884 0.059581 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 676.79  on 531  degrees of freedom
Residual deviance: 466.32  on 524  degrees of freedom
AIC: 482.32

Number of Fisher Scoring iterations: 5
```

Confusion Table

```
    type
pred.dia  No Yes
   FALSE 317  75
   TRUE   38 102
```

Backward selection procedure
As variable skin has larger p value we want to drop it first. After dropping skin the revised model AIC output is:

```
AIC: 480.53
```

Now dropping variable bp as it has largest p value here now.

```
AIC: 479.08
```

After doing backward selection logreg3=(type npreg + glu + bmi + ped + age) is better model as it contains smaller AIC.
So, number of pregnancies, plasma glucose concentration in an oral glucose tolerance test, body mass index, diabetes pedigree function, age in years are significantly affect the occurrence of diabetes.

(b) After using LDA we obtain confusion table for cutoff point 0.5

```
 pred.class
type  FALSE TRUE
  No    315   40
  Yes    74  103
```

and for cutoff point 0.8 confusion table is

```
  pred.class2
type  FALSE TRUE
  No    241  114
  Yes    24  153
```

We can see that there is significant change in the truth table when we are using 0.5 and 0.8 as cut off point in predicting false negative.
From our first truth table we get 0.271 where now we get 0.418 and 0.136 as false negative proportion.

(c) To reduce the proportion of false negatives (when people are predicted to not have diabetes, but they actually do) to about 20% of all actual diabetes cases we need to change cut off point to 0.73 . That gives us proportion of false negative 0.21.Which is approximately 20% of all actual diabetes cases. The truth table is following:

```
    print(t3)
     pred.class3
type  FALSE TRUE
  No    270   85
  Yes    38  139
```

(d) Coefficients of the first discriminant function are following

```
                     LD1
npreg      0.089453937
glu        0.026797651
bp        -0.004028686
skin       0.002668636
bmi        0.052832219
ped        0.801972183
pima1$age  0.019100341
```

The logistic regression coefficients for the full model

```
(Intercept)        npreg          glu           bp         skin          bmi
-9.554650535  0.122516579  0.035321081 -0.007695037  0.006774419  0.082678188
         ped          age
 1.308708298  0.026374756
```

So, there is significant decrease in coefficients value of the variables when we use Linear Discriminant Analysis.

(e) As the prediction accuracy is not improving substantially for cutoff point 0.73 QDA is not advantageous here. Truth table is following

```
  pred.class4
type  FALSE TRUE
  No    271   84
  Yes    38  139
```

2: The second problem

(a) Best model is (y horsepower+weight+year) because we get our best model by using exhaustive method.

```
Selection Algorithm: exhaustive
         cylinders displacement horsepower weight acceleration year
1  ( 1 ) " "       " "          " "        "*"    " "          " "
2  ( 1 ) " "       " "          " "        "*"    " "          "*"
3  ( 1 ) " "       " "          "*"        "*"    " "          "*"
4  ( 1 ) " "       " "          "*"        "*"    "*"          "*"
5  ( 1 ) "*"       " "          "*"        "*"    "*"          "*"
6  ( 1 ) "*"       "*"          "*"        "*"    "*"          "*"
```

Then we calculate BIC and get

```
-587.5637 -790.7595 -810.2816 -807.6497 -806.2862 -802.2813
```

Here, minimum BIC is for number 3 combination. So that is our best model. Then we fit linear regression for our best model and another model combined of nearest smaller BIC value and calculate anova for both model.

```
Analysis of Variance Table

Model 1: y ~ horsepower + weight + year
Model 2: y ~ horsepower + weight + year + acceleration
  Res.Df      RSS Df  Sum of Sq      F Pr(>F)
1    388 0.012891
2    387 0.012782  1 0.00010935 3.3108 0.0696 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From here we can justify that our choice of first model is perfect.

(b) For ridge regression

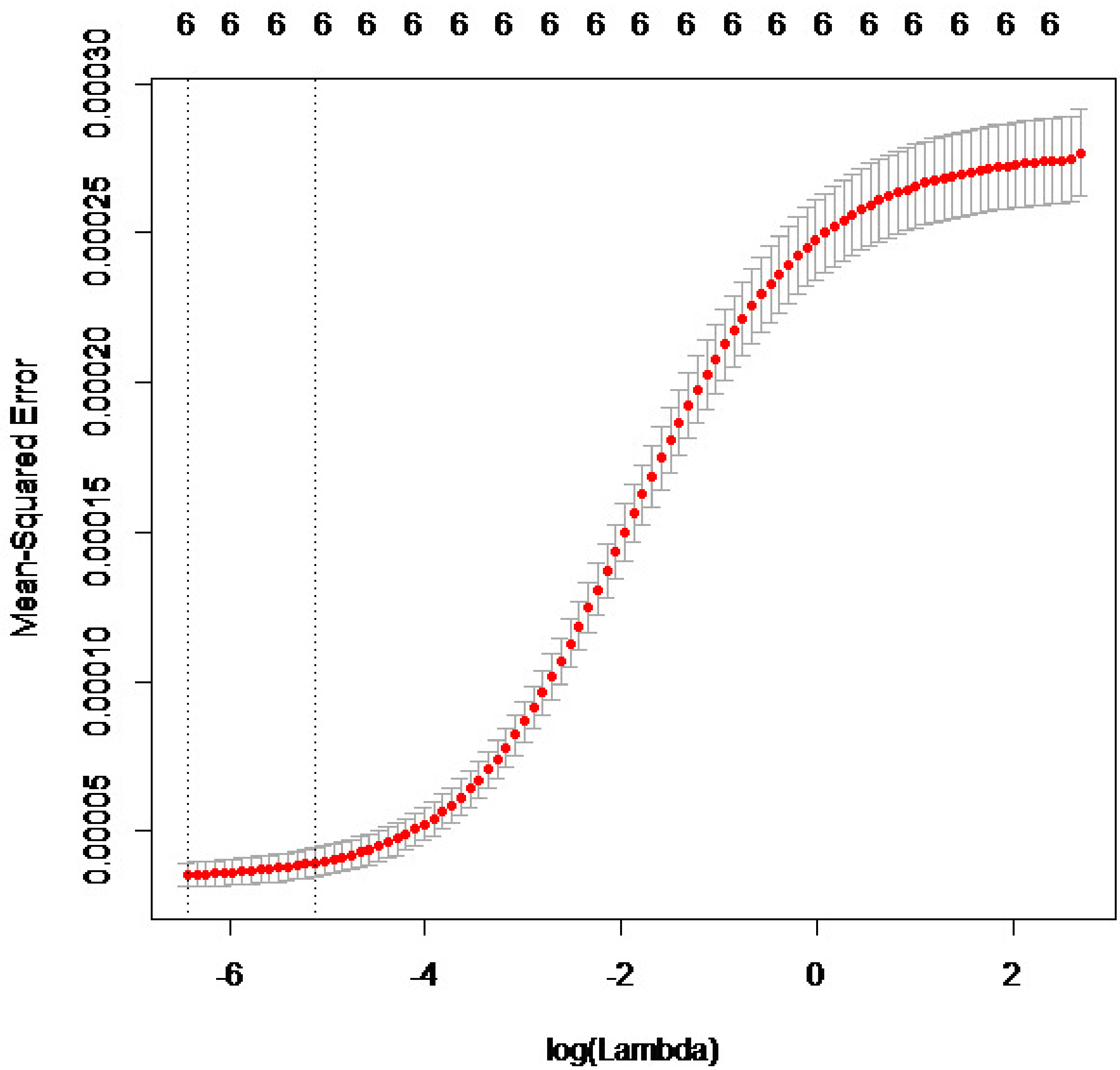

Figure 1: Ridge Plot

Mean squared error first remains fairly constant and then rises sharply. Best value for λ is 0.001614205.

Again for some other values of lambda

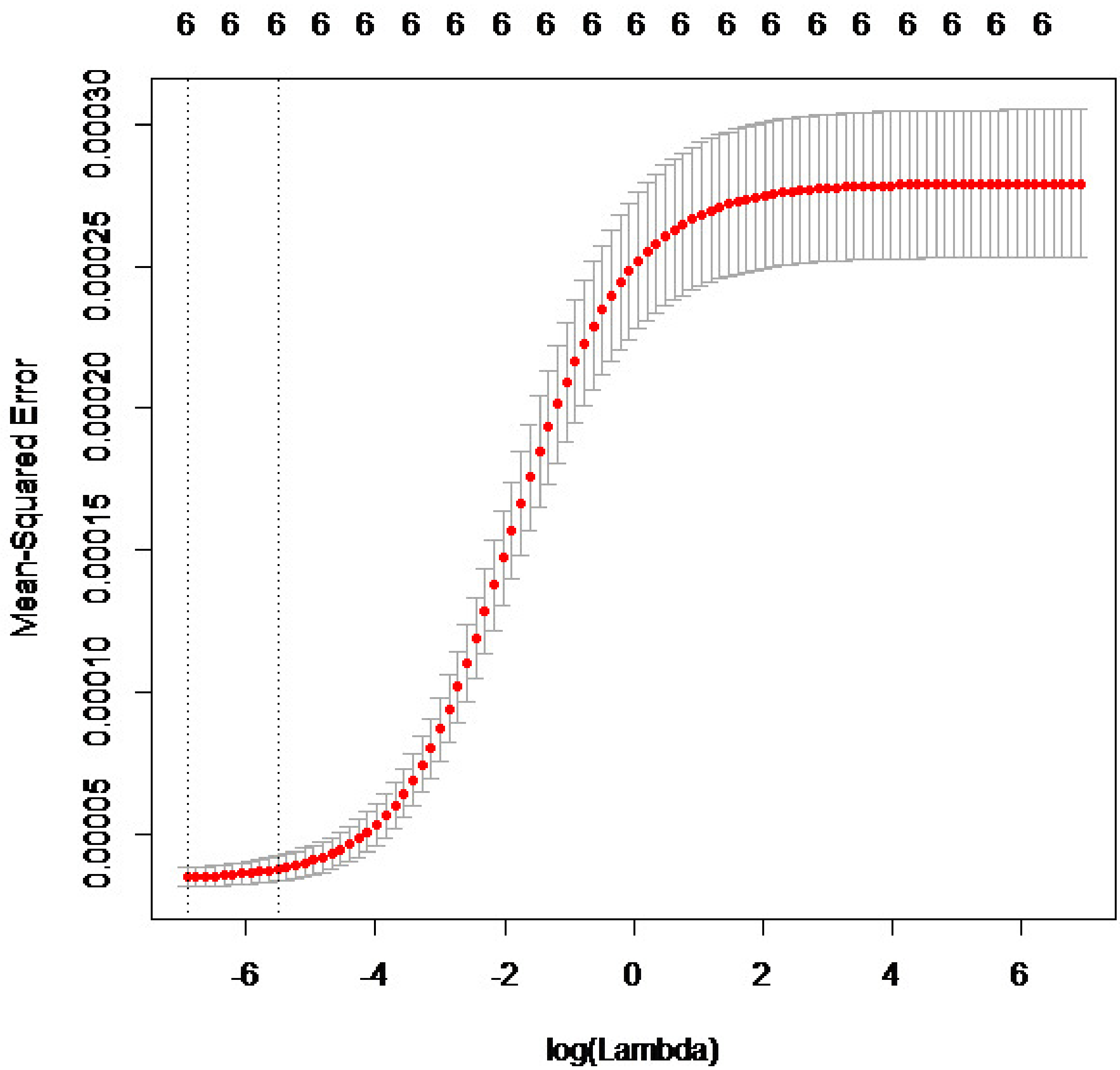

Figure 2: Ridge Plot

Mean squared error first remains fairly constant and then rises sharply. Best value for λ is 0.001.

Best coefficients are

```
(Intercept)   8.739178e-02
cylinders     1.207334e-03
displacement  1.265109e-05
horsepower    1.057848e-04
weight        8.000307e-06
acceleration  3.622330e-04
year         -1.173180e-03
```

Now, let's compare these to the "full model"

```
 (Intercept)      cylinders     displacement   horsepower       weight
  8.952961e-02   1.392059e-03  -1.702831e-05   1.137564e-04   1.109096e-05
  acceleration      year
  3.388008e-04  -1.265967e-03
```

Coefficient values from Ridge Reg. are mostly smaller, but not dramatically different from the coefficient values of full model.

Now for LASSO, plot for different values of λ is

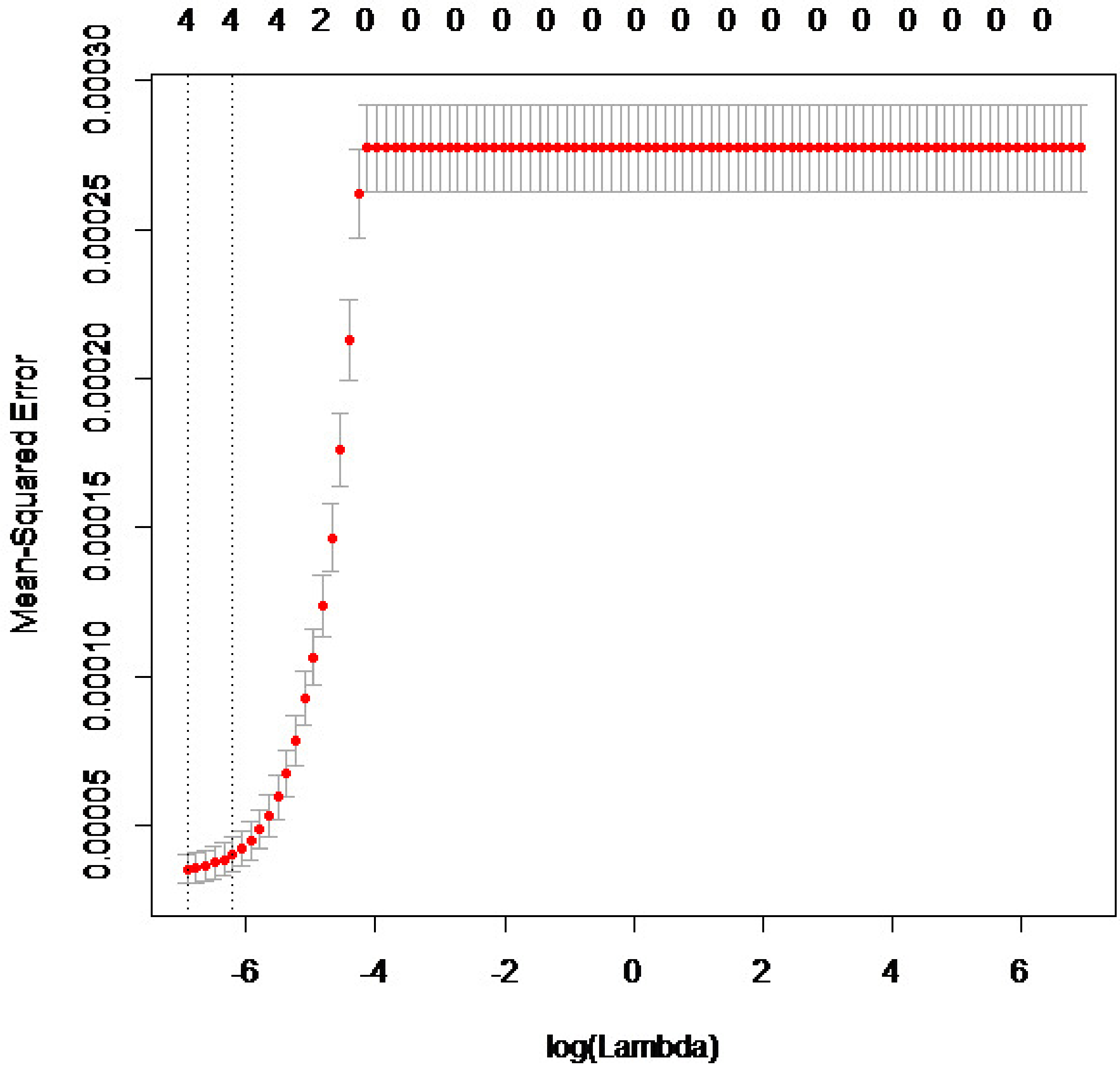

Figure 3: LASSO Plot

Best value for λ is 0.001.

Best coefficient values are

```
(Intercept)   8.605828e-02
cylinders     5.670347e-04
displacement  .
horsepower    7.243597e-05
weight        1.086538e-05
acceleration  .
year         -1.069479e-03
```

These are the best estimated coefficients by using LASSO. If we compare these values with full model we will see that they are not same but approximately close.
From LASSO we can select our best model for predicting y. In this model predictors are cylinders, horsepower, weight and year.

3: The third problem

Best model found by LASSO is

```
                        1
(Intercept)  0.554773065
Year         .
Lag1        -0.002014028
Lag2         0.007186792
Lag3         .
Lag4         .
Lag5         .
Volume       .
```

To check whether this model is predicting stock market movements effectively or not we need to find truth table.

```
        pred.dia
Direction FALSE TRUE
     Down    94  390
     Up      78  527
```

From this truth table we find the misclassification rate is 0.5702479. So, it is effectively working here for predicting the stock market movements. But one important thing we need to remember that we are dealing with training data set here. So, there is element of chance that is going to change suddenly. Another important thing is we took 0.54 as our threshold point not 0.5.

1 R Code

```
####Problem-1

require(MASS)
data(Pima.tr)
data(Pima.te)
pima1=merge(Pima.tr,Pima.te,all=TRUE)
head(pima1)
attach(pima1)
plot(pima1)
 logreg1 = glm(type ~ npreg+glu+ bp+skin+bmi+ped+pima1$age , family = binomial, data = pima1)
  summary(logreg1)
names(pima1)

# Getting the "truth table"/ confusion table
   probs1 = predict(logreg1, type="response")
   pred.dia = (probs1 > 0.5)     # classify all probs > 0.5 as "failure"
   t1 = table(type,pred.dia)
   print(t1)
##backward selection
logreg2 = glm(type ~ npreg+glu+ bp+bmi+ped+age , family = binomial, data = pima1)
  summary(logreg2)

logreg3 = glm(type ~ npreg+glu+bmi+ped+age , family = binomial, data = pima1)
  summary(logreg3)

logreg4 = glm(type ~ npreg+glu+bmi+ped , family = binomial, data = pima1)
  summary(logreg4)

###So logreg3 is better model as it contains smaller AIC. So,number of pregnancies,plasma gluc
##body mass index ,diabetes pedigree function,age in years are significantly affect the occurr

####b
attach(pima1)
lda1 = lda(type ~ npreg+glu+bmi+ped+pima1$age)
 plot(lda1, dimen = 2)

  p1 = predict(lda1)
  head(p1$post)

# "confusion table", with different cutoffs

     cutoff = 0.5
 pred.class = (predict(lda1)$post[,1] < cutoff)
 t1 = table(type, pred.class)
 n = length(type)
```

```
 mis.prob1 = 1 - sum(diag(t1))/n

 print(t1)
 print(mis.prob1)

    cutoff = 0.8
 pred.class2 = (predict(lda1)$post[,1] < cutoff)
 t2 = table(type, pred.class2)
 mis.prob2 = 1 - sum(diag(t2))/n

     print(t2)
 print(mis.prob2)

##c

cutoff = 0.73
 pred.class3 = (predict(lda1)$post[,1] < cutoff)
 t3 = table(type, pred.class3)
 mis.prob3 = 1 - sum(diag(t3))/n

     print(t3)
 print(mis.prob3)

#d
lda2 = lda(type ~ npreg+glu+ bp+skin+bmi+ped+pima1$age)
lda2$scaling
#e
 qda1 = qda(type ~ npreg+glu+ bp+skin+bmi+ped+pima1$age)
  #Getting the "truth table"/ confusion table
  cutoff=0.73
   pred.class4 = (predict(qda1)$post[,1] < cutoff)
 t4 = table(type, pred.class4)
 mis.prob4 = 1 - sum(diag(t4))/n

     print(t4)
 print(mis.prob4)

#############Problem-2
auto = read.csv("Auto.csv", na.strings="?")
Auto = na.omit(auto)
xx = as.matrix(Auto[,2:7])
y = 1/Auto$mpg
head(Auto)
install.packages("leaps")
require(leaps)

b <- regsubsets(xx,y , nbest=1, nvmax=6, method="exhaustive")
 summary(b)
```

```
summary(b)$bic
       which.min(summary(b)$bic)
  summary(b)$rss
  summary(b)$adjr2
  summary(b)$cp


 b1 <- regsubsets(xx, y, nbest=1, nvmax=6, method="backward")
 summary(b1)

  summary(b1)$bic
which.min(summary(b1)$bic)


lm1 = lm(y  ~ horsepower+weight+year, data = Auto)
  summary(lm1)
 lm2 = lm(y  ~ horsepower+weight+year+acceleration, data = Auto)
   anova(lm1, lm2)

####b
#Regularizations: ridge regression and LASSO
install.packages("glmnet")
require(glmnet)
 cgrid =10^seq (3,-3, length =100)
xx = as.matrix(Auto[,2:7])
y = 1/Auto$mpg
 ridge.mod = glmnet(xx,y,alpha =0, lambda = cgrid)
coefs = coef(ridge.mod)
 dim(coefs)
plot(cgrid, coefs[2,], type="l", log="x", ylim=c(-0.1,0.1))     # smoothing parameter lambda =
    lines(cgrid, coefs[3,], lty=2)
lines(cgrid, coefs[4,], lty=3)
lines(cgrid, coefs[5,], lty=4)
      lines(cgrid, coefs[6,], lty=5)
      lines(cgrid, coefs[7,], lty=6)
    lines(cgrid, cgrid*0,col="magenta")


  cv1 = cv.glmnet(xx,y,alpha=0)

  plot(cv1)
  cv1$lambda.min  # these are results for Ridge Regression

 cv1b = cv.glmnet(xx,y,alpha=0, lambda = cgrid)
     plot(cv1b)    # since there are no serious problems with colinearity, lambda is small
 (best1 = cv1b$lambda.min)
      cv1b$glmnet.fit
```

```
out = glmnet(xx,y,alpha=0, lambda = cgrid)
 predict(out,type="coefficients", s = best1)




# let's compare these to the "full model"
     lm1b = lm(y~xx)
 lm1b$coef     # values from Ridge Reg. are mostly smaller, but not dramatically different

 predict(out,type="coefficients", s = 0)  # this is close to lm1b (but not 100% the same ==> nu


## LASSO

 lasso1 = glmnet(xx,y,alpha = 1, lambda = cgrid)
   # alpha = 0 in this function corresponds to ridge regression, alpha = 1 corresponds to LASS
 coefs = coef(lasso1)
 dim(coefs)

 plot(cgrid, coefs[2,], type="l", log="x", ylim=c(-0.1,0.1))     # smoothing parameter lambda =
    lines(cgrid, coefs[3,], lty=2)
lines(cgrid, coefs[4,], lty=3)
lines(cgrid, coefs[5,], lty=4)
      lines(cgrid, coefs[6,], lty=5)
      lines(cgrid, coefs[7,], lty=6)

lines(cgrid, cgrid*0,col="magenta")


cv2 = cv.glmnet(xx,y,alpha=1, lambda = cgrid)   # now for LASSO
    plot(cv2)
    (best2 = cv2$lambda.min)
     out = glmnet(xx,y,alpha=1, lambda = cgrid)
 predict(out,type="coefficients", s = best2)




#####Problem3

require(ISLR)
data(Weekly)
head(Weekly)
attach(Weekly)
xx = as.matrix(Weekly[,1:7])
y = (Weekly$Direction == "Up")
cv3 = cv.glmnet(xx,y,alpha=1 , family="binomial")
  plot(cv3)
```

```
(best3 = cv3$lambda.min)
   out = glmnet(xx,y,alpha=1)
   predict(out,type="coefficients", s = best3)
# Getting the "truth table"/ confusion table
  probs=predict(out,type="response",newx=xx, s = best3)

   pred.dia = (probs > 0.54)
   t1 = table(Direction,pred.dia)
   print(t1)
sum(diag(t1)/length(y))
```