STAT460 – Homework 10
Due: April 22 at the start of class

1. Returning to the Digits Example.

   For this problem, I want to go through what was done in class for the '3' with different digit. Use the following code to load the data and to plot the digits.

```
ones = read.table("digits1.txt",sep=',')

plot.digit = function(x,zlim=c(-1,1)) {
  cols = gray.colors(100)[100:1]
  image(matrix(x,nrow=16)[,16:1],col=cols,
        zlim=zlim,axes=FALSE)
}
```

   a. Make a plot of the first 16 1's. Use this code:

```
par(mfrow=c(4,4))
for(i in 1:16){
plot.digit(digit[i,])
}
```

   b. Now, we want to get a plot of all the coordinates (ie: the scores) of all the of digits for PC1 versus PC2. Comment on this plot relative to the corresponding plot involving the 3s.

   c. Make a plot with lines at each of these quantiles:

```
quantile.vec = c(0.05,0.25,0.5,0.75,0.95)
quant.score1 = quantile(scores[,1],quantile.vec)
quant.score2 = quantile(scores[,2],quantile.vec)

plot(scores[,1],scores[,2],xlab = 'PC1',ylab='PC2')
for(i in 1:5){
abline(h = quant.score2[i])
}
for(i in 1:5){
abline(v = quant.score1[i])
}
```

   d. Now, we wish to find the 1s that correspond to where the lines intersect. Use this code:

```
plot(scores[,1],scores[,2],xlab = 'PC1',ylab='PC2',col='yellow')
for(i in 1:5){
abline(h = quant.score2[i])
}
for(i in 1:5){
abline(v = quant.score1[i])
}
identify = identify(scores[,1],scores[,2],n=25,tol=1)
```

Some notes: the identify function turns your cursor into a plus sign. Click at every intersection, going from left to right and top to bottom. Unfortunately, the output of the function is not in the proper order. Manually make an object with the indices in the same order as your clicking. The yellow part is just to make the numbers easier to see. The tol=1 part is because there are some observations not quite close enough to the intersections.

Produce a plot with all 25 1s, in the order indicated above (like in lecture). Comment on how the 1s are evolving as PC1 and PC2 are changing.

e. Let's get a scree plot. Make vertical lines at the 50% and 90% of variance explained. What are the components at the lines (do not guess from your plot, tell me exactly. Ask me if you can't figure out how to do this.)

f. Lastly, let's plot some eigenvectors (by that I mean, principal components). Using this code, plot the first 9 PCs. Write down at least 1 comment about what you see.

```
par(mfrow=c(3,3))
par(mar=c(.2,.2,.2,.2))
for(i in 1:9){
plot.digit(pcs[,i])
}
```

2. In this problem we are going to investigate model selection and prediction. Define the 'test error' to be the MSE on the test data from the estimator formed using the training data.

   a. Let's make some simulated data with 'training' and 'testing' sets.

```
nTrain = 500
nTest  = 100
n = nTrain + nTest
p = 100
set.seed(10)
X         = rnorm(n*p)
Xmat      = matrix(X,nrow=n,ncol=p)
X         = data.frame(Xmat)
Y         = rnorm(n,0,.25)
Xtrain    = X[1:nTrain,]
XtrainMat = Xmat[1:nTrain,]
Xtest     = X[(nTrain+1):n,]
XtestMat  = Xmat[(nTrain+1):n,]
Ytrain    = Y[1:nTrain]
Ytest     = Y[(nTrain+1):n]
```

   For each of the next parts, fit the model on the training data, get the test error, and (if appropriate for the method) report the number of selected variables/components. **Use set.seed(100)**.

   b. i) Linear Model (use P-value $< .05$ as model selection criterion).
      ii) Forward Selection.

   c. Ridge Regression with λ chosen by cross-validation.

   d. Lasso with λ chosen by cross-validation.

   e. PCR model with the number of PCs (call it M) chosen by cross-validation.

   f. PLS model with the number of PCs (call it M) chosen by cross-validation.

   g. Comment on the results obtained. How accurately can we predict the response? Is there much difference among the test errors resulting from these six approaches?

3. Now, we are going to redo the previous problem, but with some correlation between Y and X.

```
nTrain = 500
nTest  = 100
n = nTrain + nTest
p = 100
set.seed(10)
X         = rnorm(n*p)
Xmat      = matrix(X,nrow=n,ncol=p)
X         = data.frame(Xmat)

Xtrain    = X[1:nTrain,]
XtrainMat = Xmat[1:nTrain,]
Xtest     = X[(nTrain+1):n,]
XtestMat  = Xmat[(nTrain+1):n,]

beta1   = .01
beta10  = .025
beta100 = .05
Y = Y + X[,1]*beta1 + X[,10]*beta10 + X[,100]*beta100
Ytrain    = Y[1:nTrain]
Ytest     = Y[(nTrain+1):n]
```

Report all the same information as requested in problem 2. Additionally, do the methods include the appropriate variables (i.e.: X1, X10, X100)?

b. i) Linear Model (use P-value $< .05$ as model selection criterion).
   ii) Forward Selection.

c. Ridge Regression with λ chosen by cross-validation.

d. Lasso with λ chosen by cross-validation.

e. PCR model with the number of PCs (call it M) chosen by cross-validation.

f. PLS model with the number of PCs (call it M) chosen by cross-validation.

g. Comment on the results obtained. How accurately can we predict the response? Is there much difference among the test errors resulting from these six approaches?

4. Compare the methods' ability to do prediction and variable selection when Y is uncorrelated with X vs. when Y is correlated with X.