HIV Treatment Analysis using Ridge and LASSO

Rob Leonard (robleonard@tamu.edu (mailto:robleonard@tamu.edu))

Introduction

A major issue with antiretroviral drugs is the mutation of the virus' genes. Because of its high rate of replication (10^9 to 10^{10} virus per person per day) and error-prone polymerase, HIV can easily develop mutations that alter susceptibility to antiretroviral drugs. The emergence of resistance to one or more antiretroviral drugs is one of the more common reasons for therapeutic failure in the treatment of HIV.

In the paper 'Genotypic predictors of human immunodeficiency virus type 1 drug resistance', a sample of in vitro. HIV viruses were grown and exposed to a particular antiretroviral therapy. We have a measurement which is the susceptibility of the virus to treatment, in which larger values indicate less susceptible. It has been log transformed. As well, we have whether there is a genetic mutation on each of 208 genes for each virus (each virus is a different row or observation). It is composed of 0's and 1's, with a 1 indicating a mutation in a particular gene.

Load HIV Data

```
set.seed(1)
load("hiv.rda")

X = hiv.train$x
Y = hiv.train$y

geneLabels = colnames(X)
```

What are n and p in this problem?

```
n <- length(Y)
p <- dim(X)[2]
```

The number of observations is 704 and the number of features is 208.

What are the features in this problem? What are the observations? What is the supervisor? What do larger values for the supervisor indicate in terms of susceptibility?
The features are the 208 genes of each HIV virus grown in the lab. The features are an indicator variable set to 1 if that specfic gene had a mutation. The supervisor is Y, which is a measure of how susceptible that specific HIV virus is to the treatment. A larger value of Y means the virus is less susceptible, that is, the treatment was less effective on this virus. The supervisor was log transformed and standardized. A range of susceptibility (broken into 3 categories) is given in the paper and the category boundaries depend on the antiviral drug type.

Problem 2

Consider the feature matrix X. Look at the output for the following chunk of code.

```
table(X)
```

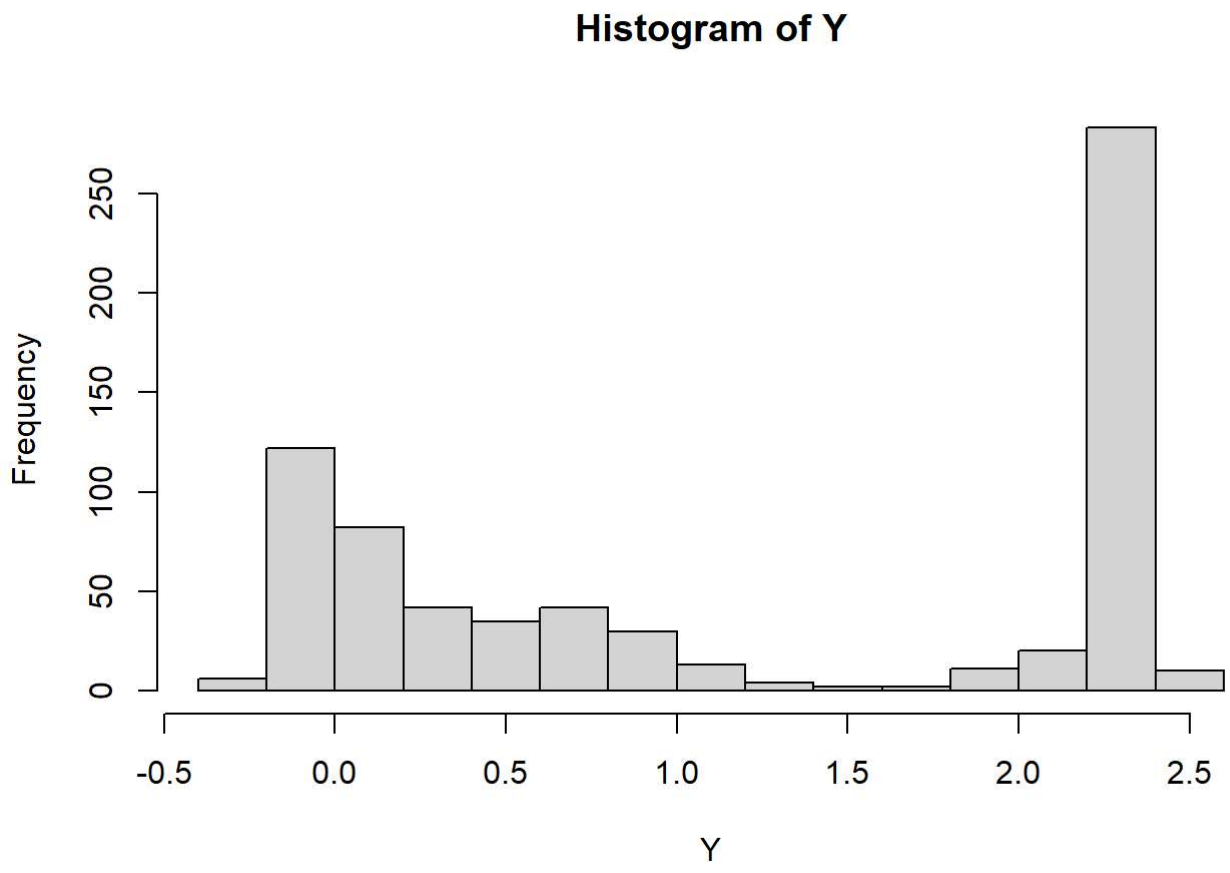
```
## X
##      0      1
## 135589  10843
```

What do these results indicate?
These results indicate that gene mutations were fairly uncommon with only 10,843 genes of the total 146,432 virus genes in the study having had a mutation. So only 7.4% of the lab grown viruses genes tracked in the study had a mutation.

Problem 3

The supervisor is the log transformed susceptibility of a virus to the considered treatment, with larger values indicating the virus is relatively more resistant (that is, not susceptible). Let's look at its distribution via a histogram.

```
hist(Y)
```



What do these results indicate? (Note that even though the supervisor doesn't look symmetric, we will still apply elastic net to it, as did the authors in that paper I included. We won't consider further transformations of the supervisor).
These results indicate that the distribution of the supervisor, the susceptibility measure, is not symmetric and is right skewed as the median is less than the mean. I'm not sure which antiviral drug type was used, but most of the types categorized a susceptibility score of 3 (after exponentiation) or less as being susceptible. From the histogram, only about half of the lab grown viruses were susceptible to the treatment. The rest had low, intermediate or high resistance. There is a group of 300 or so viruses with a susceptibility score of 2 or higher indicating treatment wasn't very effective for this group.

Problem 4

Consider finding the all subsets solution. Discuss any problems or findings you discover. In particular, how many possible models are there?

Since there are 208 features, there are 2^{208} possible models, which is not feasible to evaluate. If we choose k features, then there are ${}_{208}Choose_k$ models to evaluate. This is generally a nonconvex problem. Applying the branch and bound method may help speed up the analysis but with so many possible features to include in the model, all subsets just isn't a feasible approach.

Problem 5

We may have (at least) two goals with a data set such as this:

- inference: can we find some genes whose mutation seems to be most related to viral susceptibility?
- prediction: can we make a model that would predict whether this therapy would be efficacious, given a virus with a set of genetic mutations

Inference

Find the estimated coefficient vectors for the following procedures

- forward selection using BIC as the criterion
- lasso
- refitted lasso

Forward selection with BIC

Get the forward selection solution with risk estimate BIC (this is an object in `summary(outForward)`) (note that there will be a warning produced from using regsubsets, this isn't relevant for our purposes)

```
require(leaps)
```

```
## Loading required package: leaps
```

```
outForward      = regsubsets(x=X, y=Y, nvmax= 50, method = "forward") # limit to 50 per lecture
```

```
## Warning in leaps.setup(x, y, wt = weights, nbest = nbest, nvmax = nvmax, : 12
## linear dependencies found
```

```
## Reordering variables and trying again:
```

```
sumForward      = summary(outForward)
modelForward    = sumForward$which[which.min(sumForward$bic),]
Sforward        = modelForward[-1]
```

Lasso

Now, find the CV minimizing lasso solution

```
require(glmnet)
```

```
## Loading required package: glmnet
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-3
```

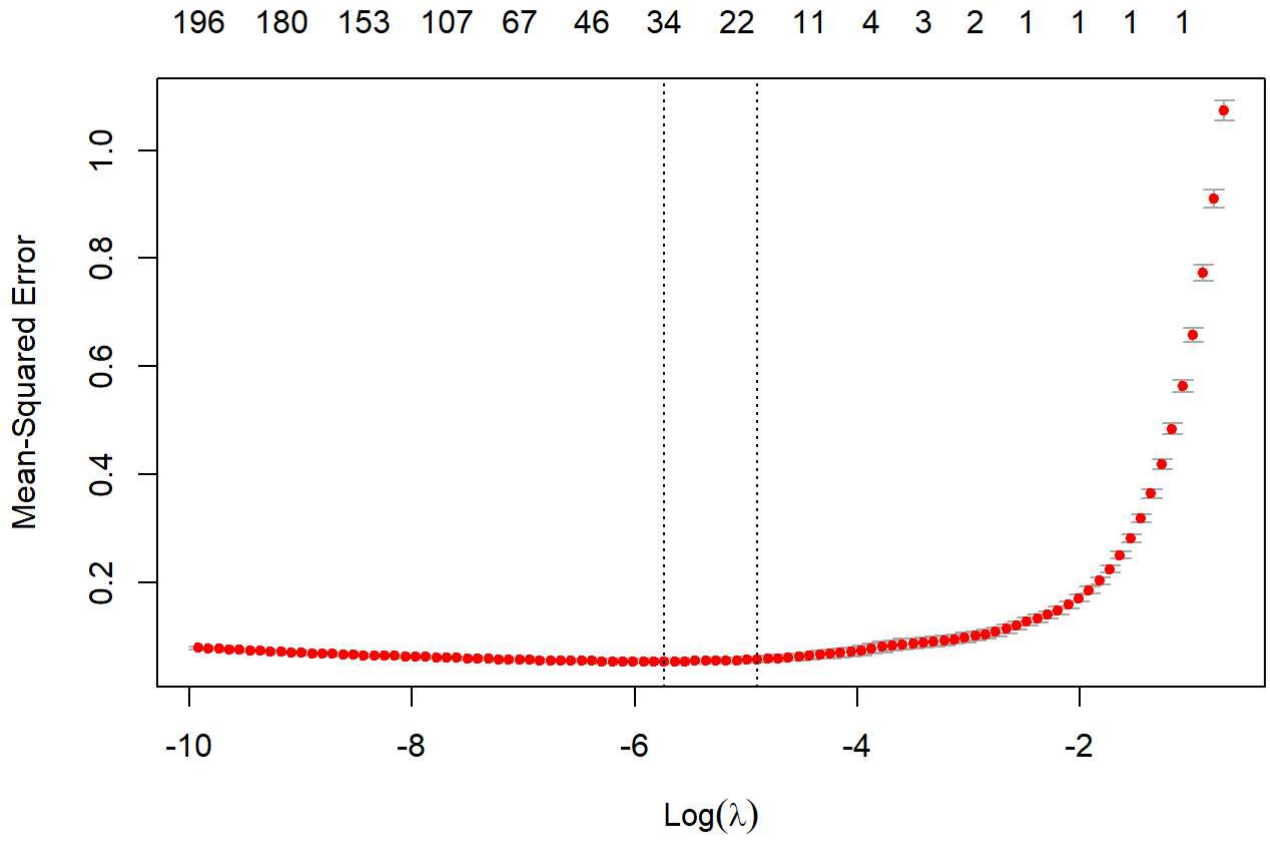
```
lassoOut     = cv.glmnet(X ,Y, alpha=1, standardize = FALSE) #Consider why no standardization...
betaHatLasso = coef(lassoOut, s='lambda.min')[-1] #[-1] to remove intercept
Slasso       = which(abs(betaHatLasso)>1e-16) ####    colnames
```

We don’t apply standardization because these are binary indicator features (0 or 1) indicating whether a mutation is present or not, so standardization is not appropriate/needed as they are all already on the same scale.

Refitted lasso

Now, find the refitted lasso using the ‘1 standard error rule’ lambda and refitting with least squares.

```
plot(lassoOut)
```




```
betaHatTemp     = coef(lassoOut, s='lambda.1se')[-1]
Srefitted       = which(abs(betaHatTemp)>1e-16)
```

Compare the selected models for each of the above methods by printing the 'geneLabels' (the feature names) for each selected model. Which genes are selected by all three of the methods?

```
cat('The selected genes from forward selection + BIC are: \n',
     names(which(Sforward[TRUE])) ,'\n')
```

```
## The selected genes from forward selection + BIC are:
##  p33 p54 p58 p65 p67 p69 p75 p90 p102 p115 p117 p151 p172 p184 p187 p210 p215
```

```
cat('The selected genes from lasso are: \n',
     colnames(X[,Slasso]),'\n')
```

```
## The selected genes from lasso are:
##  p21 p33 p41 p43 p60 p65 p67 p68 p69 p75 p83 p90 p102 p115 p118 p122 p123 p138 p151 p162 p173
p174 p177 p181 p184 p188 p190 p200 p202 p210 p211 p215 p219 p228
```

```
cat('The selected genes from refitted lasso are: \n',
     colnames(X[,Srefitted]),'\n')
```

```
## The selected genes from refitted lasso are:
##  p21 p41 p65 p67 p69 p75 p118 p151 p162 p181 p184 p200 p210 p215 p219 p228
```

```
#This is an example of intersect, replace it to answer the question
cat('The genes selected from all three methods are: \n',
colnames(X[,intersect(which(as.numeric(Sforward)==1), intersect(Slasso, Srefitted))]),'\n')
```

```
## The genes selected from all three methods are:
##  p65 p67 p69 p75 p151 p184 p210 p215
```

For the lasso, which gene is associated with the largest DECREASE in viral susceptibility (note: remember how the supervisor is coded) to this particular drug?

```
colnames(X)[which.max(betaHatLasso)]
```

```
## [1] "p184"
```

```
max(betaHatLasso)
```

```
## [1] 1.879338
```

```
coef(lassoOut, s='lambda.min')[1]
```

```
## [1] 0.1264252
```

Interpret this estimated coefficient within the context of the problem: eg. 'a change from no mutation to mutation at a specific gene…. `

A decrease in viral susceptibility means the virus is less sensitive to the drug treatment and thus has a higher measurement score as higher scores mean "less susceptible". If gene p184 changes from no mutation to a mutation, the viral susceptibility measure changes from $e^{0.1264}$ to $e^{0.1264+(1)(1.879)}$. So the susceptibility score increases from 1.135 to 7.43, holding all of the other features fixed. Depending on the drug type, this moves the susceptibility categorization from "susceptible" to at least "low/intermediate resistance."

Prediction

Now, let's look at some predictions made by these methods. Use the following for the test set:

```
X_0 = hiv.test$x
Y_0 = hiv.test$y
```

Let's compute the test error (that is the loss evaluated on this test data)

- ridge
- forward selection using Cp as the criterion
- lasso

- refitted lasso

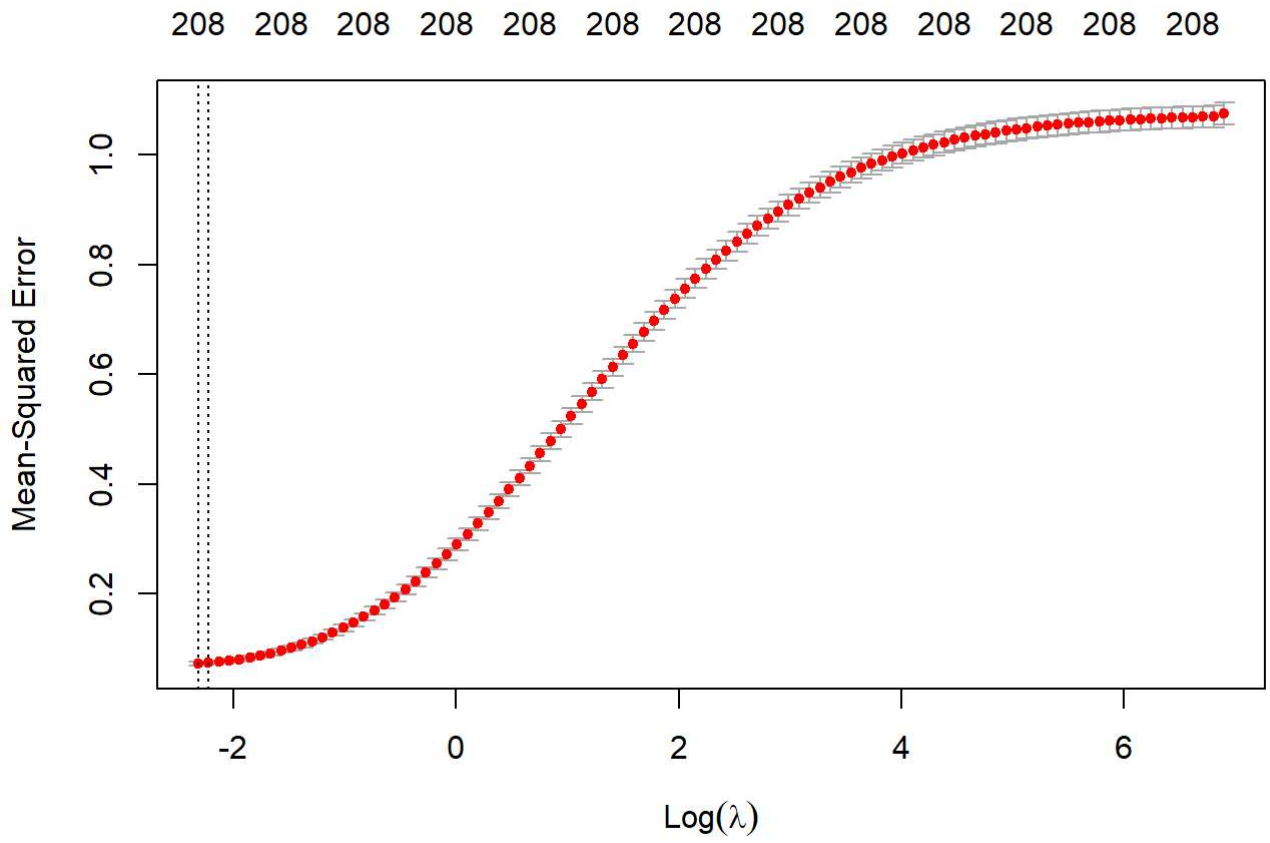
We can get the predictions out via various 'predict' functions, but instead lets make it more transparent and use the estimated coefficients and multiply them by the test feature matrix. In the following few sections, you should end up with a betaHatMETHOD that is the estimate coefficient vectors. Then, we will get the predictions on the test set and compare the test error.

Ridge regression at lambda.min

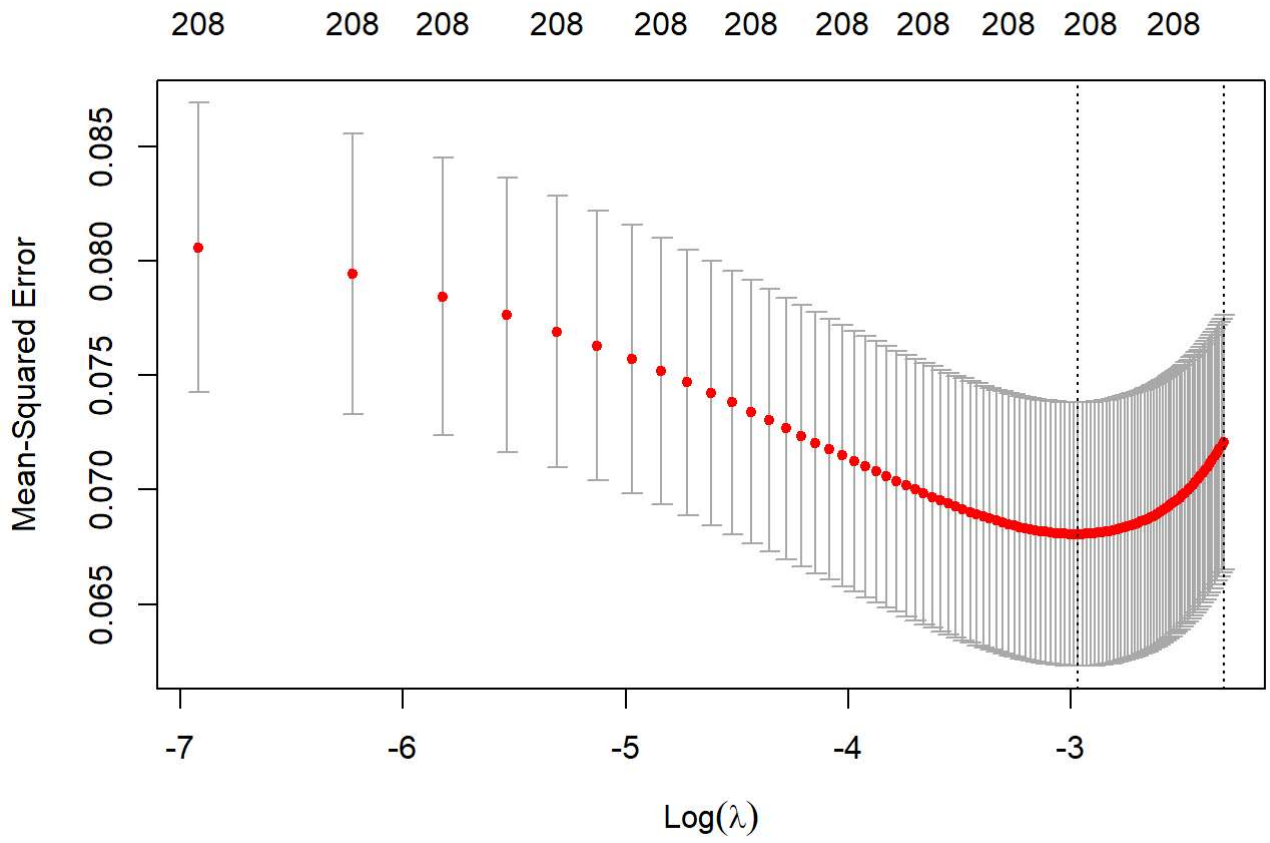
Now that we are looking at prediction, we can use ridge regression (which mainly is used for prediction). Using the package glmnet, let's plot the CV curve over the grid of λ values and indicate the minimum, and finally report the CV estimate of the risk for $\hat{\beta}_{\text{ridge}}(\hat{\lambda})$. There is no need to report the p coefficient estimates from the ridge solution.

In this instance, glmnet has a grid problem. The automatically allocated grid by glmnet has a minimum value that is too large and hence we get a `boundary' solution. Let's make two plots, one that shows the CV plot with and one without this boundary issue

```
ridgeOut = cv.glmnet(X,Y,alpha=0)
plot(ridgeOut) #This has the boundary issue
```




```
minLambda = min(ridgeOut$lambda)
lambdaNew = seq(minLambda, minLambda*0.01,length=100)
ridgeOut  = cv.glmnet(x = X, y = Y, alpha = 0,lambda = lambdaNew)
plot(ridgeOut)
```




```
betaHatRidge = coef(ridgeOut, s = 'lambda.min')
```

Why is a boundary solution for minimizing CV an issue?

CV can be viewed as function of lambda. $\hat{\lambda} = argmin_{\lambda \geq 0} CV_K(\lambda)$ If the possible range for lambda is set too high initially (in this case, could also be set too low), then the boundary of this range will result in a lambda value that does not minimize CV. Here, the penalty is set too high in glmnet and we wind up with a higher CV and worse predictions than we would get with a lower lambda (lower penalty).

Forward selection with Cp

We will change the criterion to Cp for forward selection as this should pick a model that is better at predictions than BIC would. You can use the previous `sumForward` object, just use `cp` instead of `bic`)

```
cpmodelout     = sumForward$which[which.min(sumForward$cp),] ####
Sforward       = cpmodelout[-1]   # Redefine Sforward since it's reused again below
lmForward      = lm(Y ~ X[,Sforward])
betaHatForward = coef(lmForward)
```

Lasso

We can use the previously computed lasso object to get the coefficients (just repeat the code here so we can keep everything together, but this time keep the intercept)

```
betaHatLasso = coef(lassoOut, s='lambda.min')
```

Refitted lasso

Get the refitted lasso solution by fitting the unregularized least squares solution on `Srefitted`

```
lmLassoRefit    = lm(Y ~ X[,Srefitted])
betaHatRefitted = coef(lmLassoRefit)
```

Answer 5.2.5. Getting the predictions and test error

```
# Get predictions on test set:
YhatTestForward  = X_0[,Sforward] %*% betaHatForward[-1] + betaHatForward[1]
YhatTestRidge    = X_0 %*% betaHatRidge[-1] + betaHatRidge[1]
YhatTestLasso    = X_0 %*% betaHatLasso[-1] + betaHatLasso[1]
YhatTestRefitted = X_0[,Srefitted] %*% betaHatRefitted[-1] + betaHatRefitted[1]

# Get estimate of prediction risk via the test set error
testErrorForward  = mean((YhatTestForward-Y_0)^2)
testErrorRidge    = mean((YhatTestRidge-Y_0)^2)
testErrorLasso    = mean((YhatTestLasso-Y_0)^2)
testErrorRefitted = mean((YhatTestRefitted-Y_0)^2)
```

- The prediction error from forward selection + Cp is 0.0893173
- The prediction error from ridge + lambda.min is 0.0970505
- The prediction error from lasso + lambda.min is 0.0655641
- The prediction error from refitted lasso is 0.0692691