STAT656: Homework 4

Hussein Mansour Mohamed Mansour UIN: 633003079

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.

Problem 1 (10 pts)

```
load("hiv.rda")

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

geneLabels = colnames(X)
```

Problem 1.1

What are n and p in this problem?

Answer 1.1.1.

```
p = ncol(X)
n = nrow(X)
```

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

Answer 1.1.2.

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?

features are 208 types of gens. Observations are each virus. Supervisor is the susceptibility of the virus to get treatmented. larger values for the supervisor indicate that the larger the value the less susceptible the virus is.

Problem 2 (10 pts)

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

```
table(X)
```

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

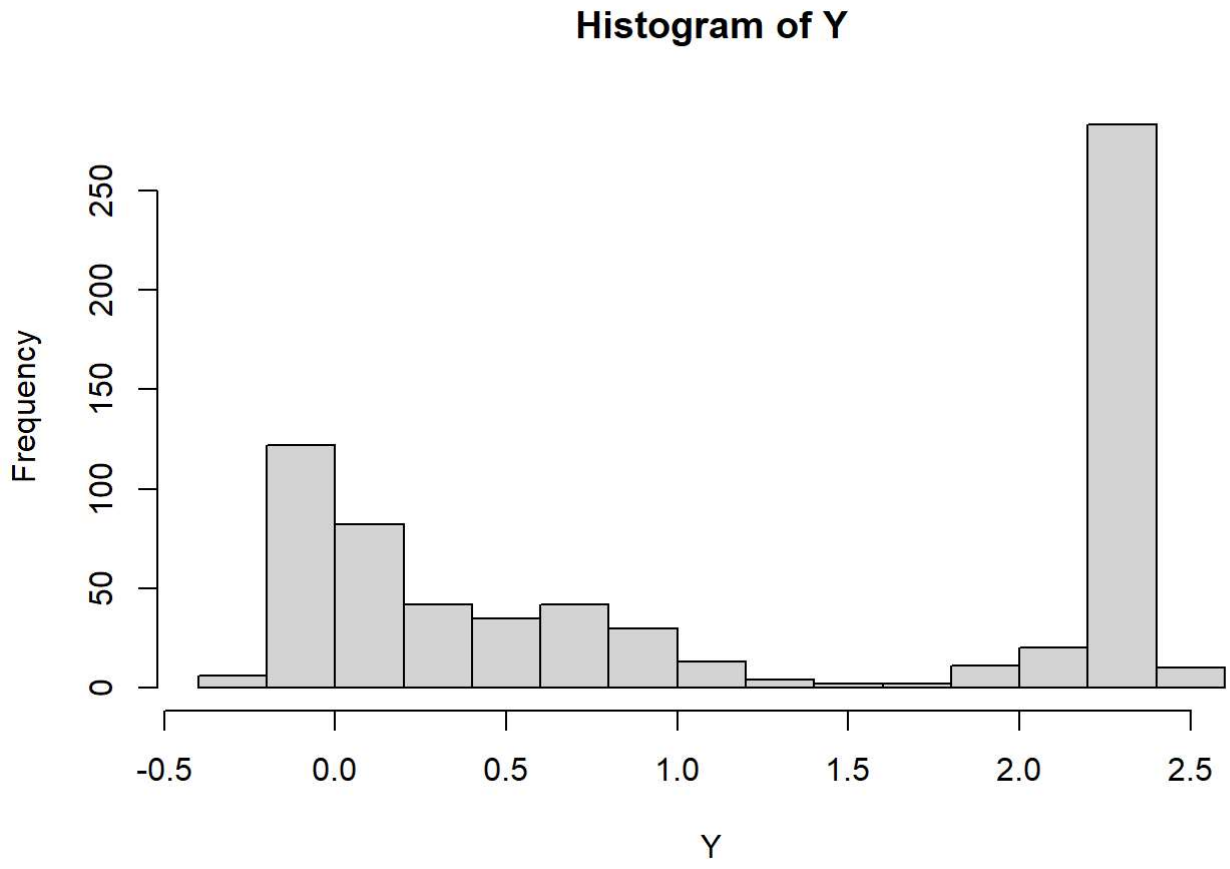
Answer 2.1

What do these results indicate? It means that the only values in the feature matrix either 0 or 1 and we have 10843 genetic mutation and 135589 non-mutation across the given data.

Problem 3 (10 pts)

The supervisor is the log transformed susceptibility of a virus to the considered treatment, with large values indicating the virus is relatively more resistant (that is, not susceptible).

```
hist(Y)
```



Answer 3.1

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)

The histogram is clearly shows that the supervisors (Y) has a peak at higher values so we can say that many viruses are more resistance to treatment. the graph looks skewed and around half of the supervisors have a value grater than 2.

Problem 4 (70 pts)

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

Problem 4.1. Inference

Find the estimated coefficient vectors for the following procedures

- lasso
- refitted lasso

Problem 4.1.1. Lasso

Now, find the CV minimizing lasso solution

Answer 4.1.1.

```
require(glmnet)
```

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

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

```
## Loaded glmnet 4.1-6
```

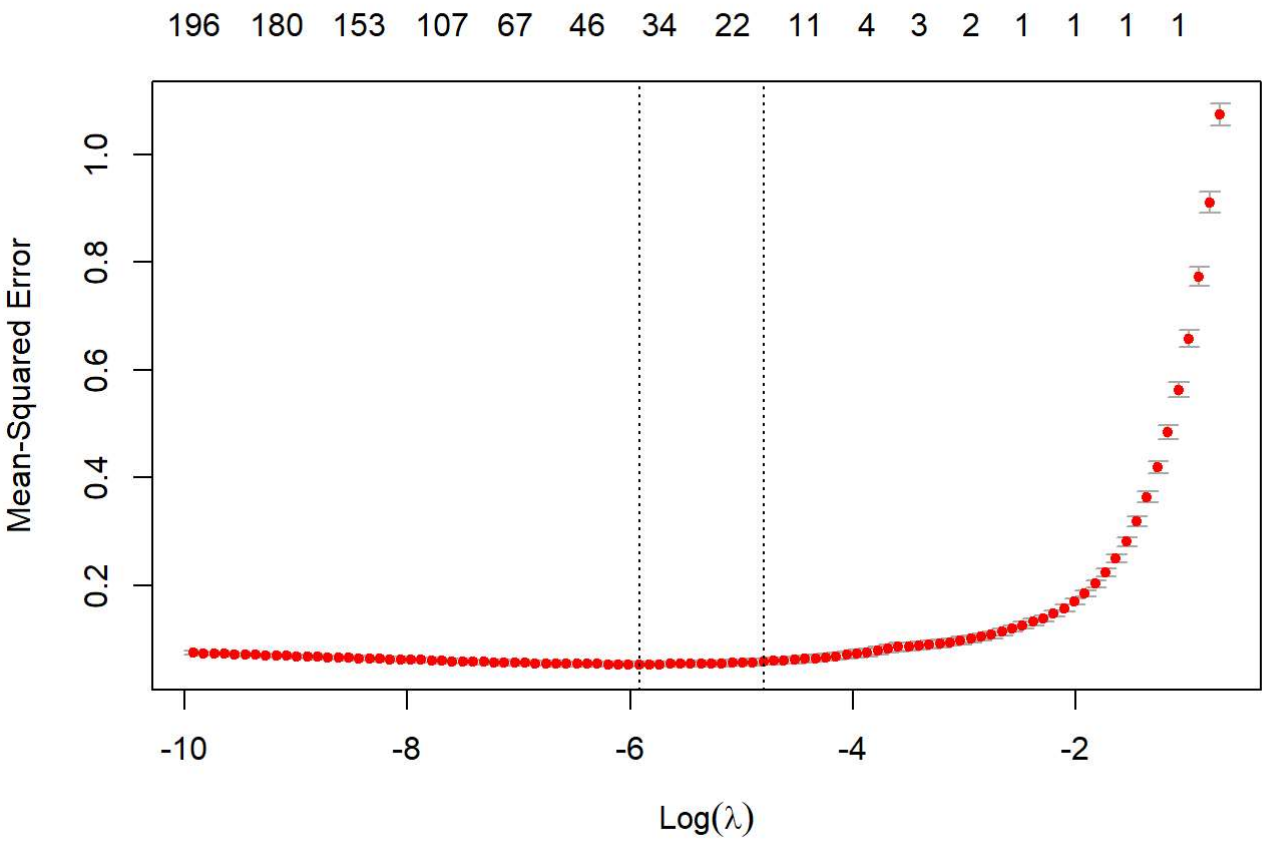
```
lassoOut     = cv.glmnet(X ,Y, alpha=1, standardize = FALSE) #Consider why we wouldn't standardi
ze
best_lambda  = lassoOut$lambda.min
betaHatLasso = coef(lassoOut,s=best_lambda)
Slasso       = which(abs(betaHatLasso) > 1e-16)
```

Problem 4.1.2. Refitted lasso

Now, find the refitted lasso using the '1 standard error rule' lambda and refitting with least squares. I've included a plot of CV so that you can see the lambda.1se solution.

Answer 4.1.2.

```
plot(lassoOut)
```




```
betaHatTemp     = coef(lassoOut,s='lambda.1se')[-1]
Srefitted       = which(abs(betaHatTemp) > 1e-16)
Xdf             = as.data.frame(X[,Srefitted])
refittedOut     = lm(Y~ ., data = Xdf)
betaHatRefitted = refittedOut$coefficients
betaHatRefitted
```

```
## (Intercept)         p21         p41         p65         p67         p69
##  0.09857841  0.06585942  0.01407802  0.66636702  0.10217962  0.07058462
##         p75         p118        p151        p162        p181        p184
##  0.11712646  0.02704918  0.31028150  0.04527725  0.05544007  1.89152262
##        p200         p210        p215        p219        p228
## -0.04322196  0.05059270  0.19717664  0.02808258  0.02857999
```

Answer 4.1.3

What are the genes selected by the refitted lasso (that is, what are the 'geneLabels' (the feature names) that correspond to the nonzero coefficients in the coefficient vector)?

```
cat('The selected genes from refitted lasso are: \n',
     geneLabels[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
```

Problem 4.1.4

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

Answer 4.1.4

```
cat('The gene is :', names(betaHatRefitted[which.max(betaHatRefitted)]) , 'The susceptibility is
: ', max(betaHatRefitted), '\n')
```

```
## The gene is : p184 The susceptibility is :  1.891523
```

Answer 4.1.5

Interpret this estimated coefficient within the context of the problem

'A change from no mutation to mutation for the above gene (P184) is associated with decrease in viral susceptibility by 1.8915226 on average holding other features constant.

Problem 4.2. Prediction

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

```
Xtest = hiv.test$x
Ytest = hiv.test$y
```

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

- ridge
- lasso
- refitted lasso

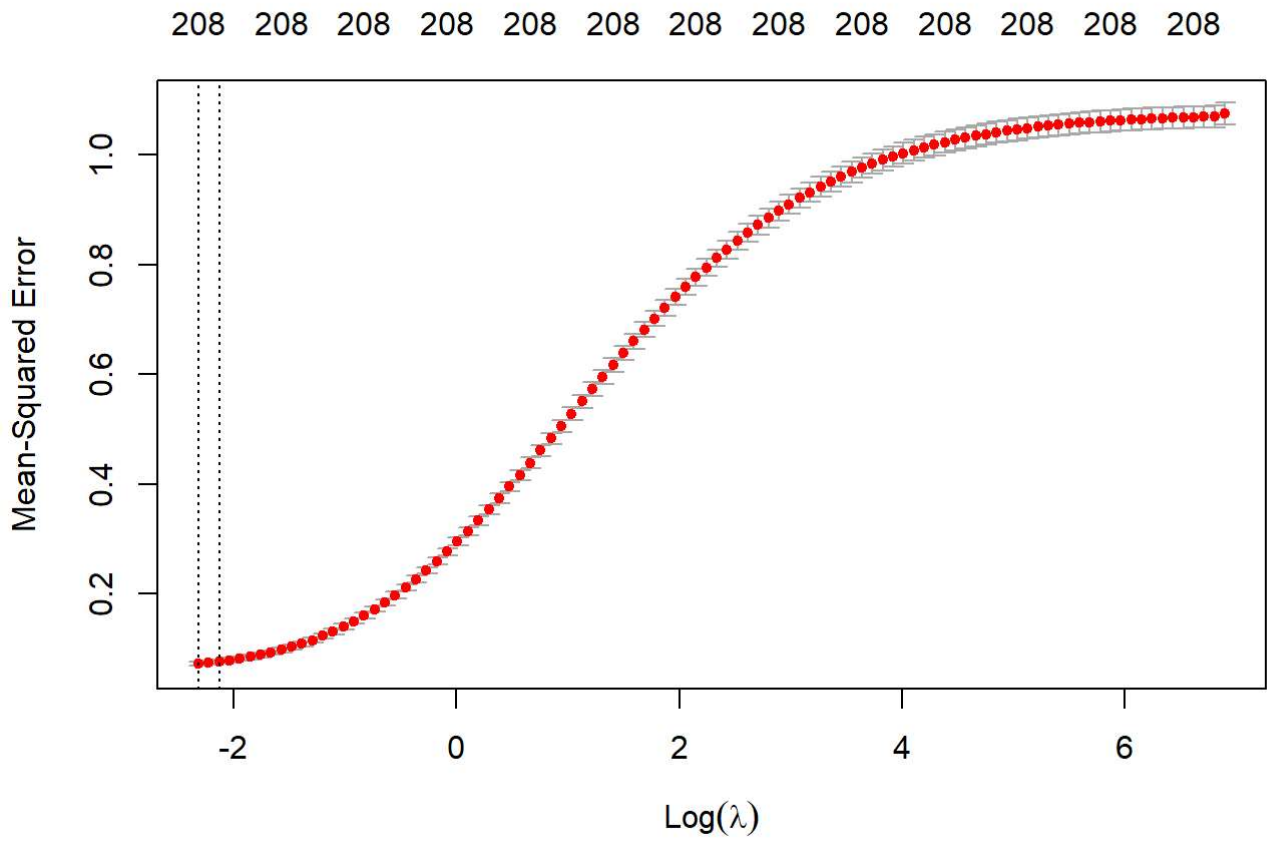
We can get the predictions out via various 'predict' functions

Problem 4.2.1. Ridge regression at lambda.min

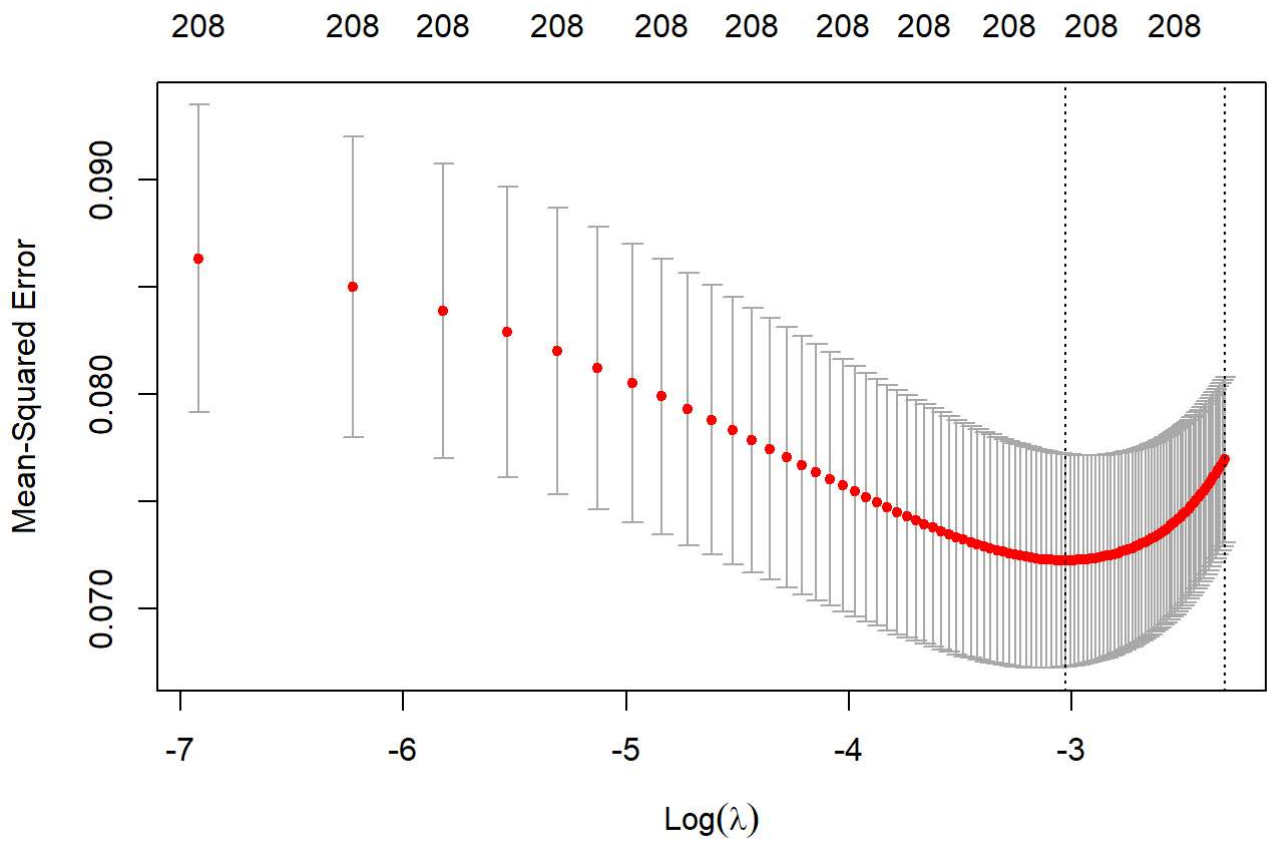
Now that we are looking at prediction, we can use ridge regression (which mainly is used for prediction). Using the package glmnet, let's plot the CV curve over the grid of lambda values and indicate the minimum, and finally report the CV estimate of the test error for ridge at each lambda.

There is no need to report the p coefficient estimates from the ridge solution. Also, glmnet has a grid problem. The automatically allocated grid by glmnet has a minimum value that is too small 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)
```




```
YhatTestRidge = predict(ridgeOut, Xtest, s = 'lambda.min')
```

Answer 4.2.1.

Why is a boundary solution for minimizing CV an issue?

the left-most dotted vertical line occurs at the CV minimum and the right-most dotted vertical line is the largest value of lambda such that CV(lambda) is within one standard error of CV(lamda-hat) which is in this case 0.0988485.Also The issue here is that the automatic lambda grid values allocated by glmnet is not good because sometimes the grid values are too far apart near the minimum or the grid doesn't allow small/large enough lambda values.so we don't know what the solution would have been if the grid contained smaller values.

Problem 4.2.2. Lasso

We can use the previously computed lasso object to get the predictions

```
YhatTestLasso = predict(lassoOut, Xtest, s = 'lambda.min')
```

Answer 4.2.3. Refitted lasso

Get the predictions for the refitted lasso. Remember, choose lambda via the 1se rule and then fit the least squares solution on the selected features.

```
betaHatTemp      = coef(lassoOut,s='lambda.1se')[-1]
Srefitted        = which(abs(betaHatTemp) > 1e-16)
XtestDF          = as.data.frame(Xtest[,Srefitted])
refittedOut      = lm(Y~ ., data = Xdf)
YhatTestRefitted = predict(refittedOut , XtestDF)
```

Answer 4.2.4. Getting the test errors

```
# Get the test error
testErrorRidge    = mean((Ytest-YhatTestRidge)**2)
testErrorLasso    = mean((Ytest-YhatTestLasso)**2)
testErrorRefitted = mean((Ytest-YhatTestRefitted)**2)
```

- The test error from ridge w/ lambda chosen as lambda.min is 0.0971249
- The test error from lasso w/ lambda chosen as is 0.0652027
- The test error from refitted lasso is 0.0692691