Homework 3

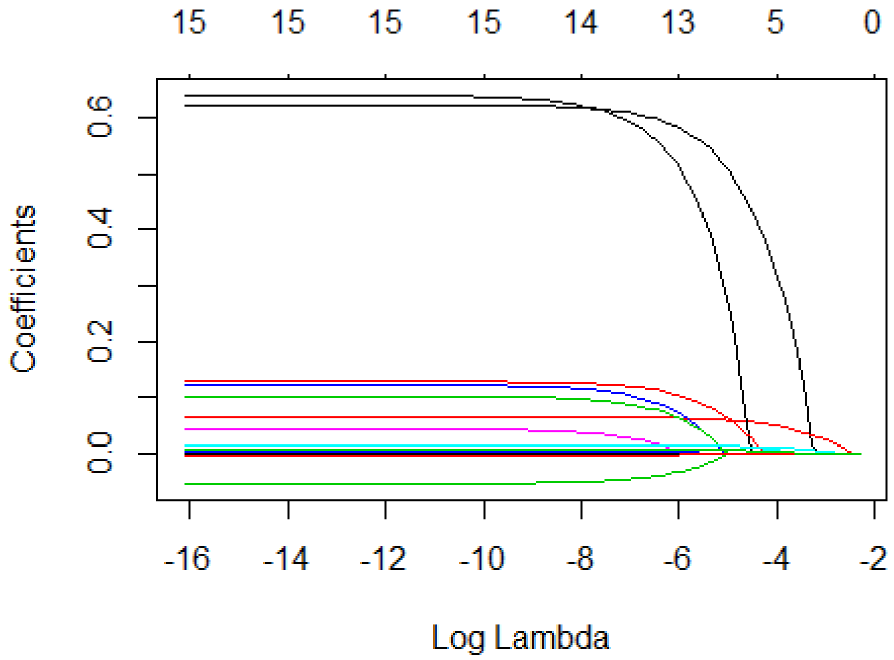
Lydia Strebe

March 13, 2019

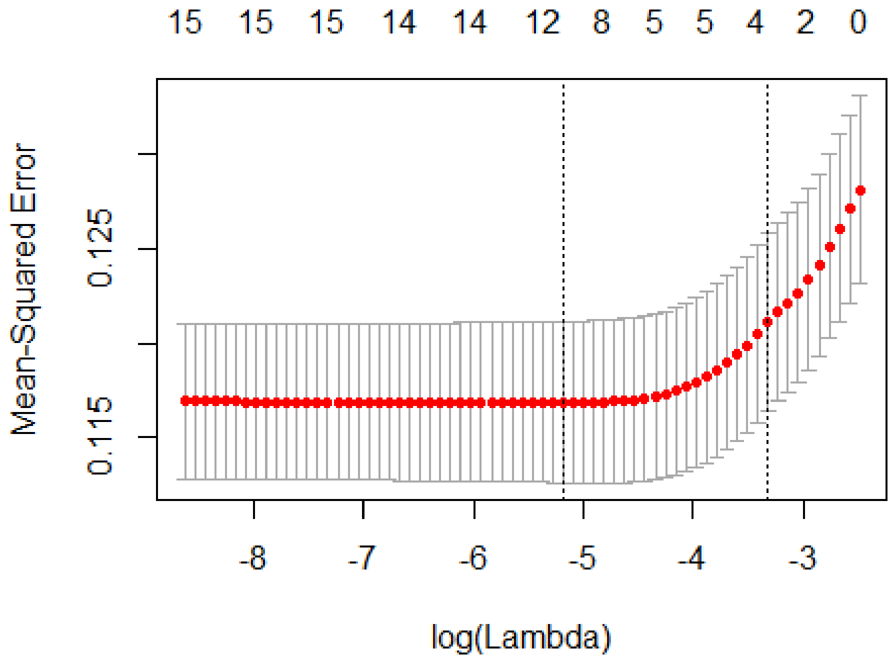
Problem 2

The following code builds a logistic regression model using LASSO to predict whether or not certain individuals have chronic heart disease.

```
set.seed(1)

library(glmnet)

## Warning: package 'glmnet' was built under R version 3.5.2

## Loading required package: Matrix

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.5.2

## Loaded glmnet 2.0-16

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.2

#Download the framingham.csv file from canvas.
og_framingham = read.csv("C:/Users/lydia/Documents/Framingham.csv", header =
TRUE)

#Delete missing entries.
framingham=na.omit(og_framingham)

#Split it into a training set and a test set.
test_rows=sample(1:nrow(framingham),size=(nrow(framingham)/4),replace=FALSE)
framingham_train=framingham[-test_rows,]
framingham_test=framingham[test_rows,]

#Create lambdas
grid=10^seq(-7,-1,length=100)

#Build logistic regression with LASSO
x=model.matrix(TenYearCHD~.,framingham_train)[,-1]
y=framingham_train$TenYearCHD
lasso.mod=glmnet(x,y,alpha=1,lambda=grid,family="binomial")
plot(lasso.mod,xvar='lambda')
```



```
#Perform cross-validation on the training data to find the best value of
lambda
cv.out=cv.glmnet(x,y,alpha=1)
plot(cv.out)
```



```
bestlam=cv.out$lambda.min
bestlam
```

```
## [1] 0.005601457
```

```
#Fit a logistic LASSO on the entire training set using the best value of
lambda.
train_log_lasso = glmnet(x,y,alpha=1,lambda=bestlam,family = "binomial")
coef(train_log_lasso)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)     -8.021002404
## male             0.526628183
## age              0.060884798
## education       -0.008792854
## currentSmoker    0.011352682
## cigsPerDay       0.014450852
## BPMeds           .
## prevalentStroke  0.334861147
## prevalentHyp     0.073002862
## diabetes         0.014158948
## totChol          0.001719403
## sysBP            0.013224158
## diaBP            .
## BMI              .
## heartRate         .
## glucose           0.006498471
```

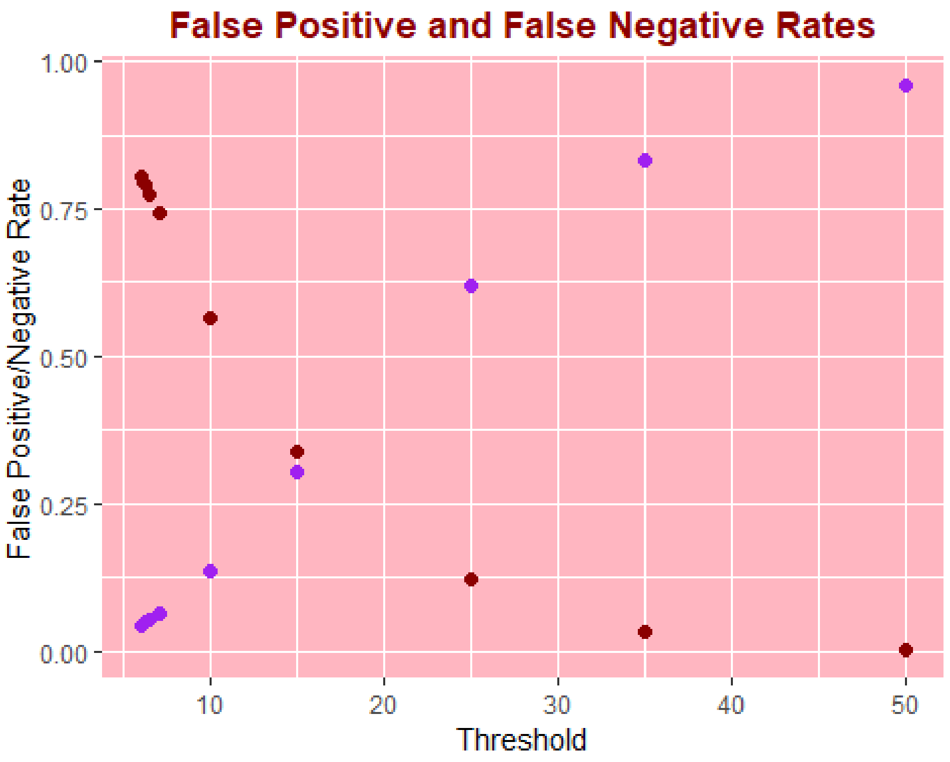
The regressors that seem the most important to me are age, cigarettes per day, total cholesteral, and the prevelence of hypertension and stroke.

Predictions

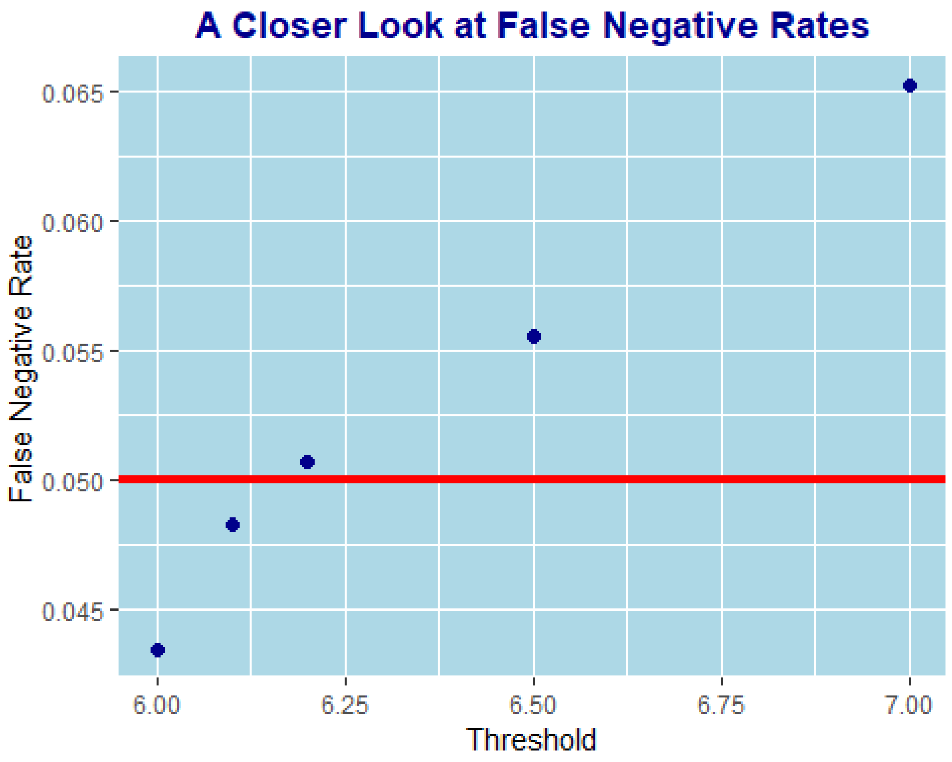
The following code uses our model to predict whether or not an individual in our test set has chronic heart disease, and tries to find a probability threshold that yields a false negative rate of 5% or lower.

```
CHD_prob=predict(train_log_lasso,x,type = "response")

#False negative and positive rates with various thresholds
CHD_pred_50=ifelse(CHD_prob>0.5,"Yes","No")
table_50=table(CHD_pred_50,framingham_train$TenYearCHD)
CHD_fn_50=table_50[1,2]/(table_50[2,2]+table_50[1,2])
CHD_fp_50=table_50[2,1]/(table_50[2,1]+table_50[1,1])

CHD_pred_35=ifelse(CHD_prob>0.35,"Yes","No")
table_35=table(CHD_pred_35,framingham_train$TenYearCHD)
CHD_fn_35=table_35[1,2]/(table_35[2,2]+table_35[1,2])
CHD_fp_35=table_35[2,1]/(table_35[2,1]+table_35[1,1])

CHD_pred_25=ifelse(CHD_prob>0.25,"Yes","No")
table_25=table(CHD_pred_25,framingham_train$TenYearCHD)
CHD_fn_25=table_25[1,2]/(table_25[2,2]+table_25[1,2])
CHD_fp_25=table_25[2,1]/(table_25[2,1]+table_25[1,1])

CHD_pred_15=ifelse(CHD_prob>0.15,"Yes","No")
table_15=table(CHD_pred_15,framingham_train$TenYearCHD)
CHD_fn_15=table_15[1,2]/(table_15[2,2]+table_15[1,2])
CHD_fp_15=table_15[2,1]/(table_15[2,1]+table_15[1,1])

CHD_pred_10=ifelse(CHD_prob>0.1,"Yes","No")
table_10=table(CHD_pred_10,framingham_train$TenYearCHD)
CHD_fn_10=table_10[1,2]/(table_10[2,2]+table_10[1,2])
CHD_fp_10=table_10[2,1]/(table_10[2,1]+table_10[1,1])

CHD_pred_06=ifelse(CHD_prob>0.06,"Yes","No")
table_06=table(CHD_pred_06,framingham_train$TenYearCHD)
CHD_fn_06=table_06[1,2]/(table_06[2,2]+table_06[1,2])
CHD_fp_06=table_06[2,1]/(table_06[2,1]+table_06[1,1])

CHD_pred_07=ifelse(CHD_prob>0.07,"Yes","No")
table_07=table(CHD_pred_07,framingham_train$TenYearCHD)
CHD_fn_07=table_07[1,2]/(table_07[2,2]+table_07[1,2])
CHD_fp_07=table_07[2,1]/(table_07[2,1]+table_07[1,1])
```

```r
CHD_pred_065=ifelse(CHD_prob>0.065,"Yes","No")
table_065=table(CHD_pred_065,framingham_train$TenYearCHD)
CHD_fn_065=table_065[1,2]/(table_065[2,2]+table_065[1,2])
CHD_fp_065=table_065[2,1]/(table_065[2,1]+table_065[1,1])

CHD_pred_061=ifelse(CHD_prob>0.061,"Yes","No")
table_061=table(CHD_pred_061,framingham_train$TenYearCHD)
CHD_fn_061=table_061[1,2]/(table_061[2,2]+table_061[1,2])
CHD_fp_061=table_061[2,1]/(table_061[2,1]+table_061[1,1])

CHD_pred_062=ifelse(CHD_prob>0.062,"Yes","No")
table_062=table(CHD_pred_062,framingham_train$TenYearCHD)
CHD_fn_062=table_062[1,2]/(table_062[2,2]+table_062[1,2])
CHD_fp_062=table_062[2,1]/(table_062[2,1]+table_062[1,1])

#Create data frams for plots
threshold=c(50,35,25,15,10,7,6.5,6.2,6.1,6)
fn_rates=c(CHD_fn_50,CHD_fn_35,CHD_fn_25,CHD_fn_15,CHD_fn_10,CHD_fn_07,CHD_fn_065,CHD_fn_062,CHD_fn_061,CHD_fn_06)
fp_rates=c(CHD_fp_50,CHD_fp_35,CHD_fp_25,CHD_fp_15,CHD_fp_10,CHD_fp_07,CHD_fp_065,CHD_fp_062,CHD_fp_061,CHD_fp_06)

fp_data=data.frame(threshold,fp_rates)

fn_data1=data.frame(threshold,fn_rates)

fn_data2=data.frame(
  "Threshold"=c(7,6.5,6.2,6.1,6),
  "FN Rate"=c(CHD_fn_07,CHD_fn_065,CHD_fn_062,CHD_fn_061,CHD_fn_06))

#Plot false negative and false positive rates
ggplot()+geom_point(data=fp_data,aes(y=fp_rates,x=threshold),size=2,colour="darkred")->p1
p1+geom_point(data=fn_data1,aes(y=fn_rates,x=threshold),size=2,colour="purple")->p2
p2+labs(title="False Positive and False Negative Rates")->p3
p3+labs(x="Threshold",y='False Positive/Negative Rate')->p4
p4+theme(panel.background=element_rect(fill="lightpink"))->p5
p5+theme(plot.title=element_text(hjust=0.5,face="bold",colour="darkred"))
```



```
#Plot false negative rates around 5%
ggplot(data=fn_data2,aes(y=FN.Rate,x=Threshold))+geom_point(size=2,colour="da
rkblue")->g1
g1+labs(title="A Closer Look at False Negative Rates")->g2
g2+labs(x="Threshold",y='False Negative Rate')->g3
g3+theme(panel.background=element_rect(fill="lightblue"))->g4
g4+theme(plot.title=element_text(hjust=0.5,face="bold",colour="darkblue"))-
>g5
g5+geom_hline(yintercept=0.05,color="red",size=1.5)
```



```
CHD_fn_061

## [1] 0.04830918

CHD_fp_061

## [1] 0.7961373
```

The threshold that yields a false negative rate closest to (but not exceeding) 5% is 6.1%. We will use this on the test data.

```
#False negative and false positive rate on the test data using the threshold
chosen above
xtest=model.matrix(TenYearCHD~.,framingham_test)[,-1]
CHD_prob_test=predict(train_log_lasso,xtest,type = "response")
CHD_pred_test=ifelse(CHD_prob_test>0.061,"Yes","No")
table_test=table(CHD_pred_test,framingham_test$TenYearCHD)
table_test

##
## CHD_pred_test   0   1
##           No  157   6
##           Yes 614 137

CHD_fn_test=table_test[1,2]/(table_test[2,2]+table_test[1,2])
CHD_fn_test

## [1] 0.04195804
```

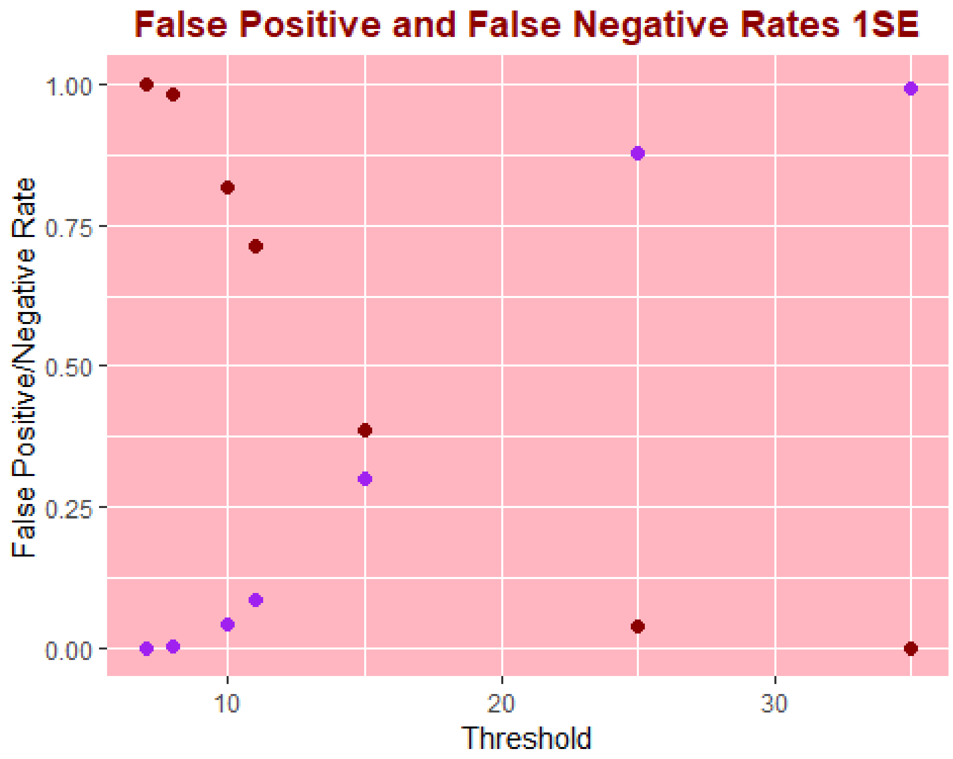
```
CHD_fp_test=table_test[2,1]/(table_test[2,1]+table_test[1,1])
CHD_fp_test

## [1] 0.7963684
```

The following code repeats the steps above using the 1se value of lambda.

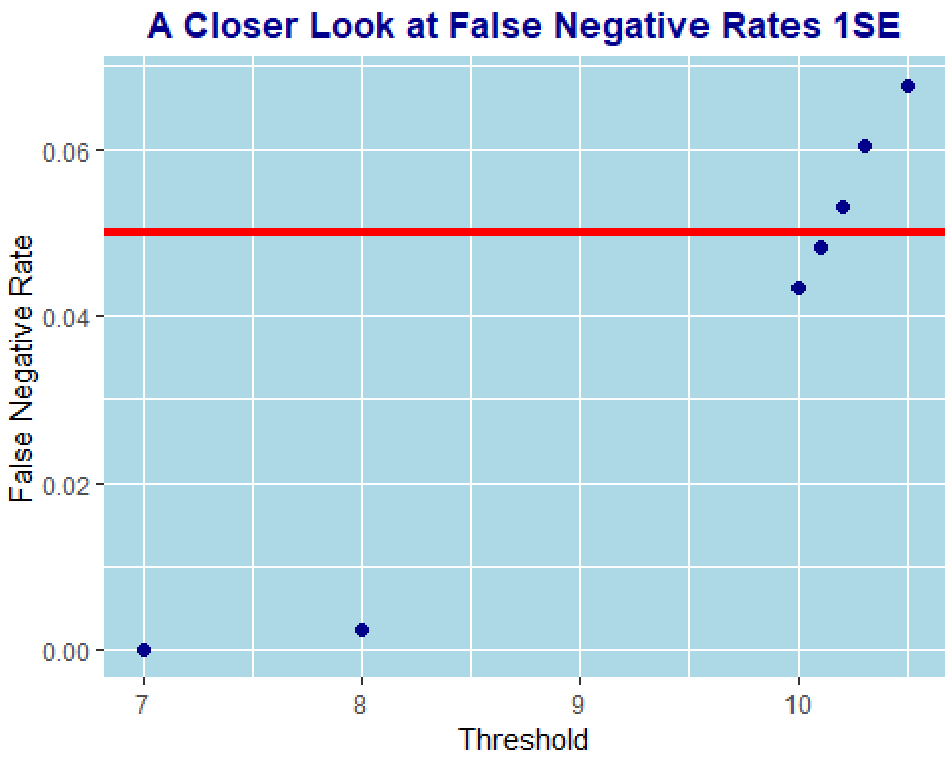
```
#Select lambda
lam_1se=cv.out$lambda.1se #lambda one standard deviation out
train_lasso_1se = glmnet(x,y,alpha=1,lambda=lam_1se,family = "binomial")
coef(train_lasso_1se)

## 16 x 1 sparse Matrix of class "dgCMatrix"
##                            s0
## (Intercept)     -4.5722950630
## male             0.0437117553
## age              0.0355934242
## education        .           
## currentSmoker    .           
## cigsPerDay       .           
## BPMeds           .           
## prevalentStroke  .           
## prevalentHyp     .           
## diabetes         .           
## totChol          .           
## sysBP            0.0072695018
## diaBP            .           
## BMI              .           
## heartRate        .           
## glucose          0.0006180554

CHD_prob_1se=predict(train_lasso_1se,x,type = "response")

#False negative and positive rates with various thresholds
CHD_pred_07_1se=ifelse(CHD_prob_1se>0.07,"Yes","No")
table_07_1se=table(CHD_pred_07_1se,framingham_train$TenYearCHD)
CHD_fn_07_1se=table_07_1se[1,2]/(table_07_1se[2,2]+table_07_1se[1,2])
CHD_fp_07_1se=table_07_1se[2,1]/(table_07_1se[2,1]+table_07_1se[1,1])

CHD_pred_08_1se=ifelse(CHD_prob_1se>0.08,"Yes","No")
table_08_1se=table(CHD_pred_08_1se,framingham_train$TenYearCHD)
CHD_fn_08_1se=table_08_1se[1,2]/(table_08_1se[2,2]+table_08_1se[1,2])
CHD_fp_08_1se=table_08_1se[2,1]/(table_08_1se[2,1]+table_08_1se[1,1])

CHD_pred_10_1se=ifelse(CHD_prob_1se>0.1,"Yes","No")
table_10_1se=table(CHD_pred_10_1se,framingham_train$TenYearCHD)
CHD_fn_10_1se=table_10_1se[1,2]/(table_10_1se[2,2]+table_10_1se[1,2])
CHD_fp_10_1se=table_10_1se[2,1]/(table_10_1se[2,1]+table_10_1se[1,1])

CHD_pred_11_1se=ifelse(CHD_prob_1se>0.11,"Yes","No")
table_11_1se=table(CHD_pred_11_1se,framingham_train$TenYearCHD)
```

```
CHD_fn_11_1se=table_11_1se[1,2]/(table_11_1se[2,2]+table_11_1se[1,2])
CHD_fp_11_1se=table_11_1se[2,1]/(table_11_1se[2,1]+table_11_1se[1,1])

CHD_pred_105_1se=ifelse(CHD_prob_1se>0.105,"Yes","No")
table_105_1se=table(CHD_pred_105_1se,framingham_train$TenYearCHD)
CHD_fn_105_1se=table_105_1se[1,2]/(table_105_1se[2,2]+table_105_1se[1,2])
CHD_fp_105_1se=table_105_1se[2,1]/(table_105_1se[2,1]+table_105_1se[1,1])

CHD_pred_103_1se=ifelse(CHD_prob_1se>0.103,"Yes","No")
table_103_1se=table(CHD_pred_103_1se,framingham_train$TenYearCHD)
CHD_fn_103_1se=table_103_1se[1,2]/(table_103_1se[2,2]+table_103_1se[1,2])
CHD_fp_103_1se=table_103_1se[2,1]/(table_103_1se[2,1]+table_103_1se[1,1])

CHD_pred_101_1se=ifelse(CHD_prob_1se>0.101,"Yes","No")
table_101_1se=table(CHD_pred_101_1se,framingham_train$TenYearCHD)
CHD_fn_101_1se=table_101_1se[1,2]/(table_101_1se[2,2]+table_101_1se[1,2])
CHD_fp_101_1se=table_101_1se[2,1]/(table_101_1se[2,1]+table_101_1se[1,1])

CHD_pred_102_1se=ifelse(CHD_prob_1se>0.102,"Yes","No")
table_102_1se=table(CHD_pred_102_1se,framingham_train$TenYearCHD)
CHD_fn_102_1se=table_102_1se[1,2]/(table_102_1se[2,2]+table_102_1se[1,2])
CHD_fp_102_1se=table_102_1se[2,1]/(table_102_1se[2,1]+table_102_1se[1,1])

CHD_pred_35_1se=ifelse(CHD_prob_1se>0.35,"Yes","No")
table_35_1se=table(CHD_pred_35_1se,framingham_train$TenYearCHD)
CHD_fn_35_1se=table_35_1se[1,2]/(table_35_1se[2,2]+table_35_1se[1,2])
CHD_fp_35_1se=table_35_1se[2,1]/(table_35_1se[2,1]+table_35_1se[1,1])

CHD_pred_25_1se=ifelse(CHD_prob_1se>0.25,"Yes","No")
table_25_1se=table(CHD_pred_25_1se,framingham_train$TenYearCHD)
CHD_fn_25_1se=table_25_1se[1,2]/(table_25_1se[2,2]+table_25_1se[1,2])
CHD_fp_25_1se=table_25_1se[2,1]/(table_25_1se[2,1]+table_25_1se[1,1])

CHD_pred_15_1se=ifelse(CHD_prob_1se>0.15,"Yes","No")
table_15_1se=table(CHD_pred_15_1se,framingham_train$TenYearCHD)
CHD_fn_15_1se=table_15_1se[1,2]/(table_15_1se[2,2]+table_15_1se[1,2])
CHD_fp_15_1se=table_15_1se[2,1]/(table_15_1se[2,1]+table_15_1se[1,1])

#Create data frams for plots
threshold_1se=c(35,25,15,11,10,8,7)
fn_rates_1se=c(CHD_fn_35_1se,CHD_fn_25_1se,CHD_fn_15_1se,CHD_fn_11_1se,CHD_fn
_10_1se,CHD_fn_08_1se,CHD_fn_07_1se)
fp_rates_1se=c(CHD_fp_35_1se,CHD_fp_25_1se,CHD_fp_15_1se,CHD_fp_11_1se,CHD_fp
_10_1se,CHD_fp_08_1se,CHD_fp_07_1se)

fp_data_1se=data.frame(threshold_1se,fp_rates_1se)

fn_data1_1se=data.frame(threshold_1se,fn_rates_1se)
```

```
fn_data2_1se=data.frame(
  "Threshold"=c(10.5,10.3,10.2,10.1,10,8,7),
  "FN
Rate"=c(CHD_fn_105_1se,CHD_fn_103_1se,CHD_fn_102_1se,CHD_fn_101_1se,CHD_fn_10
_1se,CHD_fn_08_1se,CHD_fn_07_1se))

#Plot false negative and false positive rates
ggplot()+geom_point(data=fp_data_1se,aes(y=fp_rates_1se,x=threshold_1se),size
=2,colour="darkred")->n1
n1+geom_point(data=fn_data1_1se,aes(y=fn_rates_1se,x=threshold_1se),size=2,co
lour="purple")->n2
n2+labs(title="False Positive and False Negative Rates 1SE")->n3
n3+labs(x="Threshold",y='False Positive/Negative Rate')->n4
n4+theme(panel.background=element_rect(fill="lightpink"))->n5
n5+theme(plot.title=element_text(hjust=0.5,face="bold",colour="darkred"))
```



```
#Plot false negative rates around 5%
ggplot(data=fn_data2_1se,aes(y=FN.Rate,x=Threshold))+geom_point(size=2,colour
="darkblue")->m1
m1+labs(title="A Closer Look at False Negative Rates 1SE")->m2
m2+labs(x="Threshold",y='False Negative Rate')->m3
m3+theme(panel.background=element_rect(fill="lightblue"))->m4
m4+theme(plot.title=element_text(hjust=0.5,face="bold",colour="darkblue"))-
>m5
m5+geom_hline(yintercept=0.05,color="red",size=1.5)
```



```
CHD_fn_101_1se

## [1] 0.04830918

CHD_fp_101_1se

## [1] 0.8025751
```

The threshold that yields a false negative rate closest to (but not exceeding) 5% is 10.1%. We will use this on the test data.

```
#False negative and false positive rate on the test data using the threshold
chosen above
prob_test_1se=predict(train_lasso_1se,xtest,type = "response")
pred_test_1se=ifelse(prob_test_1se>0.101,"Yes","No")
table_test_1se=table(pred_test_1se,framingham_test$TenYearCHD)
table_test_1se

##
## pred_test_1se   0   1
##           No  144   5
##           Yes 627 138

fn_test_1se=table_test_1se[1,2]/(table_test_1se[2,2]+table_test_1se[1,2])
fn_test_1se

## [1] 0.03496503
```

```
fp_test_1se=table_test_1se[2,1]/(table_test_1se[2,1]+table_test_1se[1,1])
fp_test_1se

## [1] 0.8132296
```

Although we are not using the “best” lambda, and we are using a higher threshold, the false negative rate is surprisingly lower on our test data.