Patrick Rogan
UNI: psr2125

STAT4702 Homework 11

6.8.1

a. Best subset selection will have the smallest training RSS (up to a tie with one of the other methods). As best subset selection contains all subsets including forward stepwise and backward stepwise, it must then contain the subset with the lowest RSS which may well have been found by one of the other methods.

b. It is impossible to say. Best subset selection may have picked the lowest training RSS, but depending on the situation this may have been found by chance from looking at such a large set of models which could result in poor test RSS. Both backward subset selection and forward selection may miss the best model.

c.

   i. True, by the algorithm design the (k+1) variable model is the k variable model plus one additional variable.

   ii. True, once the (k+1) variable model has been selected, only subsets of this model are considered.

   iii. False, say backward selection removes predictor X_1 at the first iteration and forward selection adds X_1 in the first iteration. Forward selection will then always have that predictor and backward selection will never have it.

   iv. False, there is no guarantee forward selection will produce the same result as backward selection of (k+1) up to a predictor. Table 6.1 provides an example if we consider backwards selection instead of best subset selection. It may be the case that a variable that provides the best single predictor model is not part of a multivariable model.

   v. False, best subset selection is allowed to choose whatever subsets produce the lowest error for k predictors, regardless of the (k+1) set. Table 6.1 shows a case where a smaller subset is not in a larger subset. However, it is possible that the predictors in the k^{th} subset are in the (k+1) model.

6.8.2

a. iii is correct. Depending on the situation, Lasso can completely remove predictors from the model, which will decrease model variance at the expense of adding some bias. Depending on the true relationship between observations and predictors, this could increase prediction accuracy.

b. iii is correct. Much like Lasso, ridge regression will decrease model variance by regularizing the parameter estimates. If this variance is smaller than the bias added, prediction will be improved depending on the true relationship between observations and predictors.

c. ii is correct. Turning to nonlinear models will increase variance, and potentially decrease bias if nonlinear predictors are more indicative of the underlying behavior of the data.

6.8.3

a. iv. Steadily decrease. As the model is allowed to become more flexible, the bias will decrease and the training RSS will decrease.

b. ii. Decrease initially, and then eventually start increasing in a U shape. Initially the model will better fit the data and the bias will decrease, but at some point, by making the model more flexible the variance will increase to the point where the test error will begin to increase (overfitting). Note this is generally true, but could be incorrect in different situations.

c. iii. Steadily increase. The model variance will increase as more predictors and larger coefficients are allowed.

d. iv. Steadily decrease. The squared bias will decrease as the model becomes more flexible which is the case as the constraints on the coefficients are relaxed.

e. v. Remain constant. By the definition of irreducible error no matter what we do to the model we will not change this quantity.

6.8.4

a. iii. Steadily increase. The training error will increase as the model flexibility is decreased.

b. ii. Decrease initially, and then eventually start increasing in a U shape. The model will produce a lower test RSS due to a decrease in variance that outweighs an increase in bias, however eventually the model will lose flexibility and the test error will increase. This is generally true.

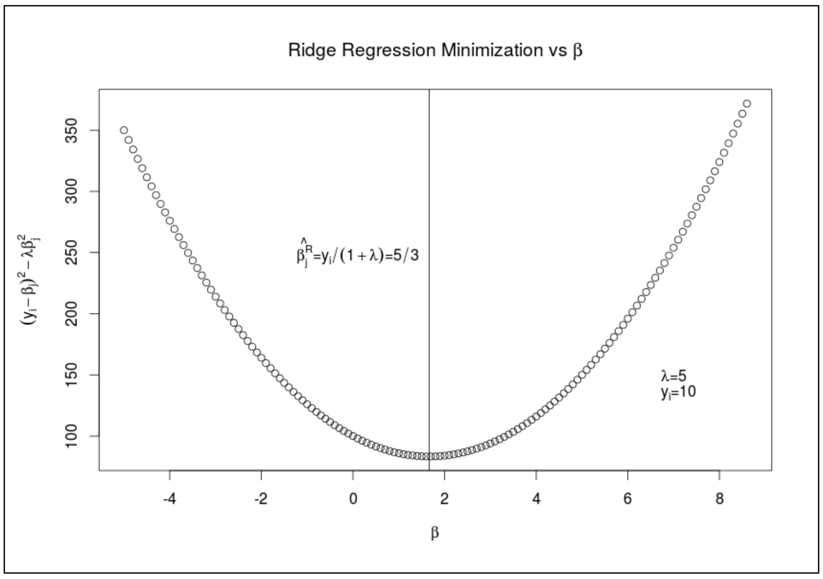
c. iv. Steadily decrease. As λ increases we expect several of the estimated coefficients to decrease, reducing the variance of the model.

d. iii. Steadily increase. As λ increases the model will become less flexible and the squared bias will increase.

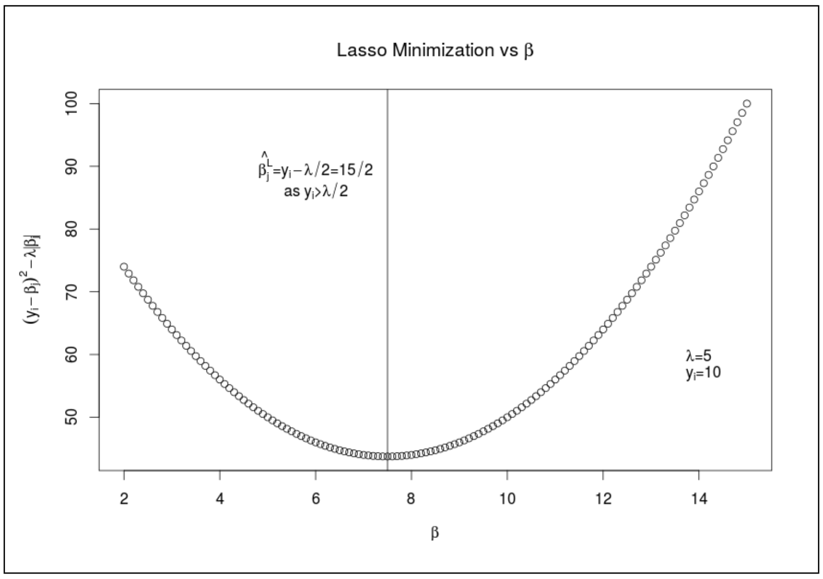
e. v. Remain constant. For the same reason ad 6.8.3 part e.

6.8.5 See handwritten section.
6.8.6

a.



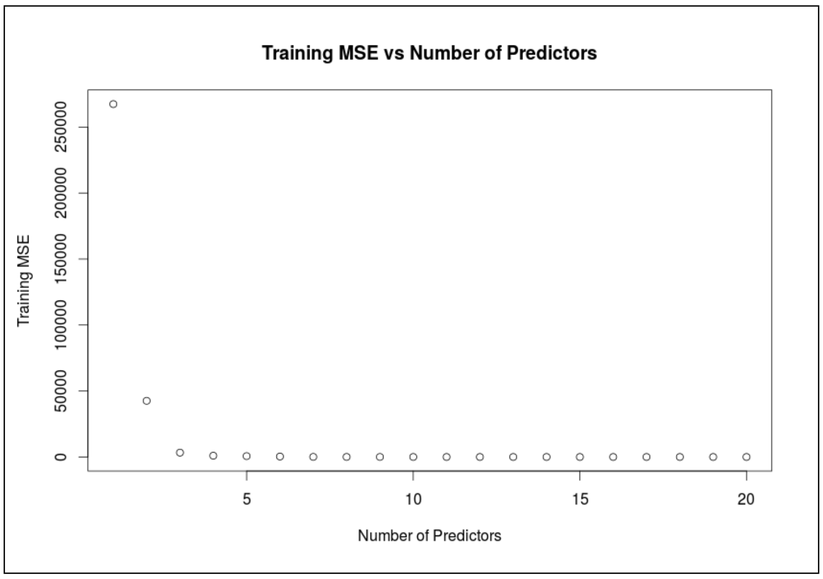
b.



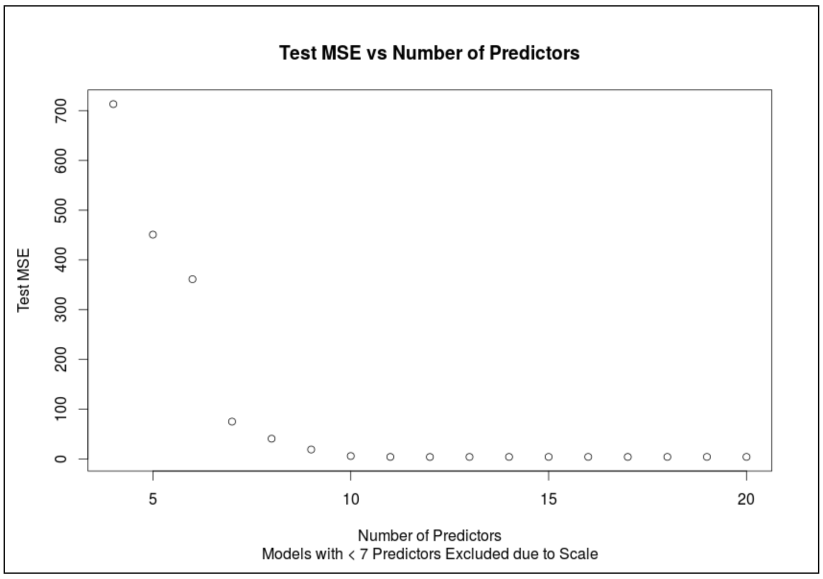
6.8.7 See handwritten section.
6.8.10

a,b. See supporting code.

c.



d.



e. A scenario was created for which the number of predictors that minimizes the test MSE is 12. Several different data sets were used before settling on this one. Given the underlying behavior of the data is linear and the model of choice is linear, the test MSE drops rapidly once relevant predictors have been added to the model. We can create situations with much more pronounced 'U' shaped test MSE behavior if we generate non-linear data.

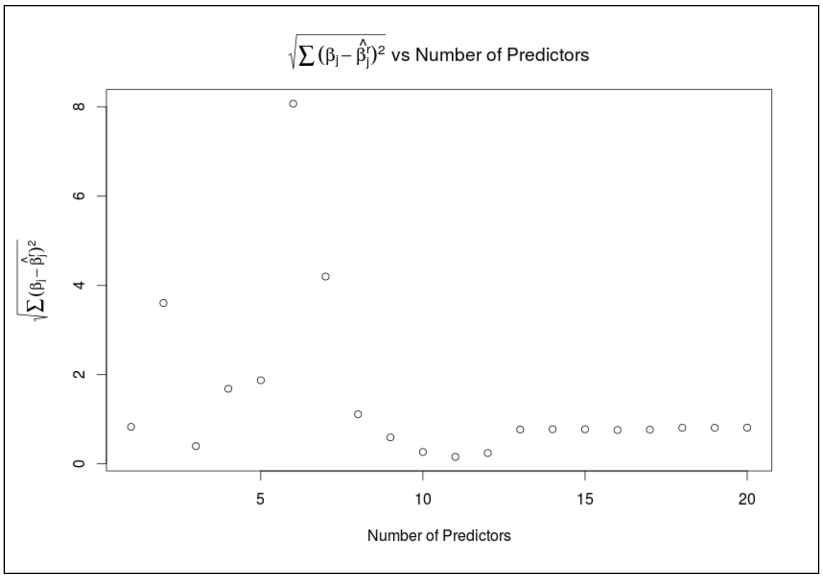
f. The model does a very good job of estimating the coefficient values. This is expected given the linear model and linear behavior.

```
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 1.893e+02  1.509e+02    1.255     0.21
X2          1.160e+00  1.038e-01   11.173   <2e-16 ***
X3          3.991e+00  7.246e-02   55.079   <2e-16 ***
X5          9.993e+01  1.522e-01  656.558   <2e-16 ***
X7          6.015e+00  9.354e-02   64.299   <2e-16 ***
X8          1.701e+02  7.417e-01  229.386   <2e-16 ***
X10         7.003e+01  2.644e-02 2649.141   <2e-16 ***
X12         1.000e+02  1.552e-02 6444.079   <2e-16 ***
X13         3.911e+00  7.487e-02   52.238   <2e-16 ***
X14         2.011e+00  8.775e-03  229.193   <2e-16 ***
```

```
X15         2.000e+02  2.492e-02 8024.447   <2e-16 ***
X16         1.057e+00  6.845e-02   15.445   <2e-16 ***
X17         6.976e+00  3.031e-02  230.189   <2e-16 ***

True Values
beta2 = 1
beta3 = 4
beta5 = 100
beta7 = 6
beta8 = 170
beta10 = 70
beta12 = 100
beta13 = 4
beta14 = 2
beta15 = 200
beta16 = 1
beta17 = 7
```

g. This plot is somewhat similar to what was observed for the minimum MSE. There is a minimum in the 11-13 predictor range followed by a levelling off in performance as more predictors are added.



```
################################################################################################
###################################### Supporting Code ######################################
################################################################################################
# STAT4702 Homework 11
# Patrick Rogan
# UNI: psr2125

#  6.8.6 ------------------------------------------------------------------
beta = seq(-5,8.66,0.1)
lambda = 5
y_i = 10

# Part A
par(mar=c(5.1,4.7,4.1,2.1))
plot(beta, (y_i - beta)^2 + lambda*beta^2, xlab = expression(beta),
     ylab = expression((y[i]-beta[j])^2-lambda*beta[j]^2),
     text(0.1,250,expression(paste(hat(beta[j]^R) ,"=", y[i]/(1+lambda),"=", 5/3))),
     main = expression(paste("Ridge Regression Minimization vs ", beta)))
text(7,150,expression(paste(lambda,"=",5)))
```

```r
text(7.1,135,expression(paste(y[i],"=",10)))
abline(v = y_i/(1 + lambda))

# Part B
beta = seq(2,15,0.1)
par(mar=c(5.1,4.7,4.1,2.1))
plot(beta, (y_i - beta)^2 + lambda*abs(beta), xlab = expression(beta),
     ylab = expression((y[i]-beta[j])^2-lambda*abs(beta[j])),
     text(6,90,expression(paste(hat(beta[j]^L) ,"=", y[i]-lambda/2,"=", 15/2))),
     main = expression(paste("Lasso Minimization vs ", beta)))
text(6,86,expression(paste("as ",y[i], ">",lambda/2)))
text(14,60,expression(paste(lambda,"=",5)))
text(14.1,57,expression(paste(y[i],"=",10)))
abline(v = y_i - lambda/2)

# 6.8.10 -----------------------------------------------------------------
set.seed(1)
X = matrix(0,nrow = 1000, ncol = 20)
meanVec = c(0,120,2,0.25,100,14,1,0,10,99,2,100,2000,3,180,18,0,0,0,0)
sdVec = c(1,.7,1,5,.5,.2,.8,.1,.2,3,2,5,1,9,3,1.1,2.4,2.1,3,0.2)

eps = rnorm(1000, mean = 0, sd = 2.0)
betaVect = c(0,1,4,0,100,1,6,170,1,70,0,100,4,2,200,1,7,0,0,0)
beta0 = 3
Y = beta0 + eps
for (i in 1:20){
  X[,i] = rnorm(1000, mean = meanVec[i], sd = sdVec[i])
  Y = Y + betaVect[i]*X[,i]
}

simDat = data.frame(Y,X)
colnames(simDat) = c("Y","X1","X2","X3","X4","X5","X6","X7","X8","X9","X10","X11","X12",
                     "X13","X14","X15","X16","X17","X18","X19","X20")

# Part B
library(caret)

train.ix = as.integer(createDataPartition(simDat[,1], p = .9, list = FALSE, times = 1))
train = rep(FALSE,length(simDat[,1]))
train[train.ix] = TRUE
simDat.test = simDat[!train,]

# Part C
library(leaps)
attach(simDat)
regfit.full = regsubsets(Y~., simDat, subset = train,nvmax = 20)
summary(regfit.full)
trainMSE = rep(0,20)
testMSE = rep(0,20)
delBeta = rep(0,20)

for (i in c(1:20)){
  tempFrame = simDat[colnames(simDat[,-1])[summary(regfit.full)$which[i,c(2:21)]]]
  tfz = data.frame(tempFrame[train,])
  tfz2 = data.frame(tempFrame[!train,])
  tempFrame = data.frame(simDat[,1],tempFrame)
  tempFrame = tempFrame[train,]
  colnames(tempFrame)[1] = "Y"
  attach(tempFrame)
  lm.fit = lm(Y ~., tempFrame, subset = train)
  betav= unname(lm.fit$coefficients[-1]) - betaVect[summary(regfit.full)$which[i,c(2:21)]]
  delBeta[i] = sqrt(betav%*%betav)

  lm.p = unname(predict(lm.fit, tfz))
  lm.p2 = unname(predict(lm.fit,tfz2))

  trainMSE[i] =  mean((simDat$Y[train]-lm.p)^2)
  testMSE[i] =  mean((simDat$Y[!train]-lm.p2)^2)
}

plot(c(1:20),trainMSE, xlab = "Number of Predictors", ylab = "Training MSE",
     main = "Training MSE vs Number of Predictors")
plot(c(4:20),testMSE[4:20], xlab = "Number of Predictors", ylab = "Test MSE",
     main = "Test MSE vs Number of Predictors",
     sub = "Models with < 7 Predictors Excluded due to Scale")
par(mar=c(5.1,5.1,4.1,2.1))
plot(c(1:20),delBeta[1:20], xlab = "Number of Predictors",
     ylab = expression(sqrt(sum((beta[j]-hat(beta[j]^r)))^2)),
     main = expression(paste(sqrt(sum((beta[j]-hat(beta[j]^r)))^2), " vs Number of Predictors")))
```