

#Q 11.1

```
rm(list = ls())
uscrime <- read.table("uscrime.txt", stringsAsFactors = FALSE, header = TRUE)

#Perform backward elimination. Can use the trace argument to suppress the output.
model_back <- lm(Crime~., data = uscrime)
step(model_back, direction = "backward")
step(model_back, direction = "backward", trace = 0)

#Perform forward selection
model_forward <- lm(Crime~1, data = uscrime)
step(model_forward, scope = formula(lm(Crime~.,data=uscrime)), direction = "forward")

#Perform stepwise regression
model_both <- lm(Crime~., data = uscrime)
step(model_both, scope = list(lower = formula(lm(Crime~1,data = uscrime)),upper=
formula(lm(Crime~.,data=uscrime))),direction = "both")

#Perform LASSO. This is the same as the elastic net model with a lambda/alpha of 1.
library(glmnet)
set.seed(42)
model_lasso <- cv.glmnet(x=as.matrix(uscrime[,-16]),
y=as.matrix(uscrime[,16]),alpha=1,nfolds=8,nlambda=20,type.measure = "mse",family =
"gaussian", standardize = TRUE)

#Print some outputs, including a plot of MSE with confidence intervals for each lambda,
the lambda that corresponds to the lowest MSE, and the coefficients for the model
corresponding to the best lambda
model_lasso
plot(model_lasso)
model_lasso$lambda.min
cbind(model_lasso$lambda, model_lasso$cvm, model_lasso$nzzero)
coef(model_lasso,s=model_lasso$lambda.min)

#Perform elastic net. For loop to find lowest standard error
for (i in 1:10) {
  set.seed(42)
  model_elasticnet <- cv.glmnet(x=as.matrix(uscrime[,-16]),
y=as.matrix(uscrime[,16]),alpha=i*.1,nfolds=8,nlambda=20,type.measure = "mse",family =
"gaussian", standardize = TRUE)
  print(i*.1)
  print(min(model_elasticnet$cvstd))
}

#Perform elastic net just weighing LASSO and Ridge Regression equally
model_elasticnet <- cv.glmnet(x=as.matrix(uscrime[,-16]),
y=as.matrix(uscrime[,16]),alpha=.5,nfolds=8,nlambda=20,type.measure = "mse",family =
"gaussian", standardize = TRUE)
```