Question 11.1 Using the crime data set uscrime.txt from Questions 8.2, 9.1, and 10.1, build a regression model using: 1. Stepwise regression 2. Lasso 3. Elastic net

1. **Stepwise Regression**

```r
#perform backward elimination
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")

#test out the model
bestmodel <- step(model_back,
                  direction = "backward",
                  trace = 0)
```

Here, I built both the forward and backward elimination variable selection methods. I then looked for which version had a lower AIC to determine which was the better model. The backwards model had an AIC of 503.93 while the forward method had an AIC of 504.79. Thus, I used the backward method to evaluate the fit of the model.

```r
#test out the model
bestmodel <- step(model_back,
                  direction = "backward",
                  trace = 0)
summary(bestmodel)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -6426.10    1194.61  -5.379 4.04e-06 ***
M              93.32      33.50   2.786  0.00828 **
Ed            180.12      52.75   3.414  0.00153 **
Po1           102.65      15.52   6.613 8.26e-08 ***
M.F            22.34      13.60   1.642  0.10874
U1          -6086.63    3339.27  -1.823  0.07622 .
U2            187.35      72.48   2.585  0.01371 *
Ineq           61.33      13.96   4.394 8.63e-05 ***
Prob        -3796.03    1490.65  -2.547  0.01505 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’

Residual standard error: 195.5 on 38 degrees of freedom
Multiple R-squared:  0.7888,	Adjusted R-squared:  0.7444
F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

The output tells me that the r-squared is .78, so the model is explaining about 78% of the variability, which is pretty good! Also, the p-value is quite low and the p-value of each coefficient is quite low, so the results of this model are strong

2. Lasso

```
#perform LASSO
install.packages("glmnet")
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)
```

```
Measure: Mean-Squared Error

    Lambda Index Measure    SE Nonzero
min   8.84     8   62393 11474      11
1se  23.31     6   69651 13921       9
```

For the lasso method, we use the glmnet function and set alpha to be 1. This ensure that the parabolic part of the equation is zeroed out. The function does 8 fold cross validation testing out 20 different values of lambda and trying to find the value of lambda that gives the optimal Mean Squared Error. In this case, the optimal Lambda is 8.84 and the MSE for that lambda is 62,393. Importantly, we standardized the data, so the values of the coefficients will not make sense unless we unscale them.

3. Elastic Net

```
#perform elastic net
install.packages("caret")
library(caret)
set.seed(123)
y=as.matrix(uscrime[,16])
folds = as.matrix(createFolds(y, k = 8))

model_elasticnet <- cv.glmnet(x=as.matrix(uscrime[,-16]),
                              y,
                              alpha =.5,
                              nfolds = 8,
                              nlambda=20,
                              type.measure="mse",
                              family="gaussian",
                              standardize=TRUE)
```

```
Measure: Mean-Squared Error

    Lambda Index Measure   SE Nonzero
min  17.68     8   61369 8427      13
1se  46.61     6   69445 5279      11
>
```

I tried to use foldid for this function since testing out different values of alpha would benefit from keeping the same set of folds. However, i kept getting an error when i plugged in foldid = folds, so i took that part of the code out and ran the elastic net regression with alpha = .5. I ended up getting the optimal lambda at 17.68, with more non-zero coefficients. The MSE was 61,369 so it performed better than the lasso regression. The elastic net will shrink the value of the coefficients which leads to some bias, but reduces the variance of the estimates of the coefficients.

I also ran the elastic net with different values of alpha between 0 and 1, but because i couldn't figure out the foldid argument, it was hard to evaluate against each other.