11.1.1)

To view R code, please reference Appendix section 11.1.

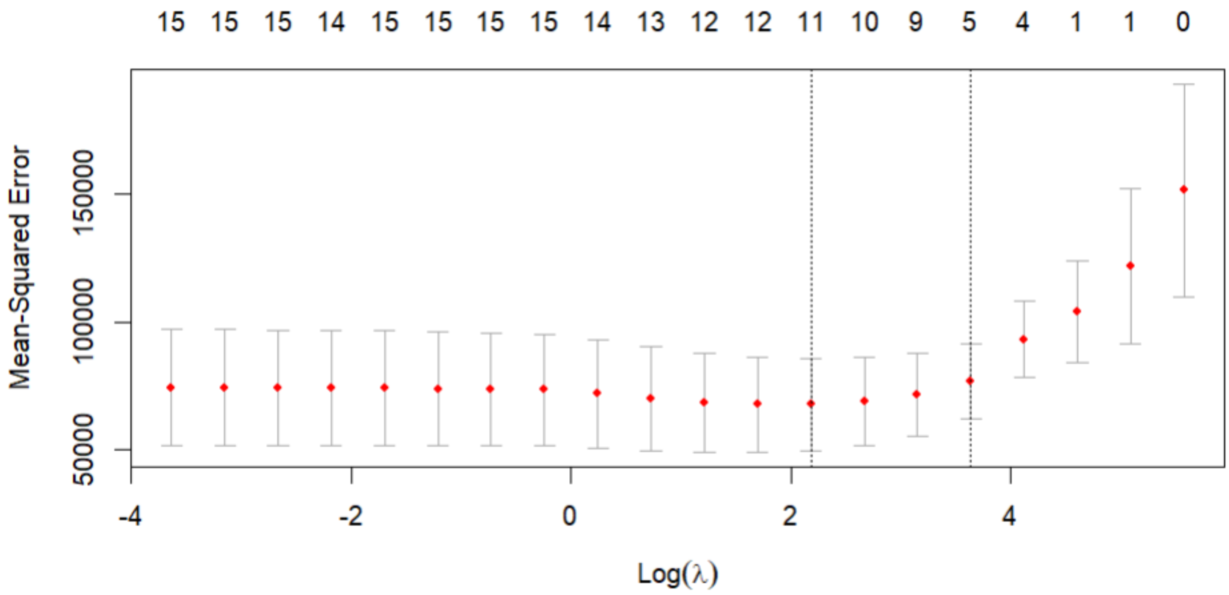
Using the “step” function in R, a stepwise regression model was created for the data in “uscrime.txt”. The details of this model are shown below.

```
Call:
lm.default(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq +
    Prob, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-444.70 -111.07    3.03  122.15  483.30

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 ‘ ’ 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
```

11.1.2)

To view R code, please reference Appendix section 11.1.

Using the “glmnet” function in R, a Lasso regression model was created for the data in “uscrime.txt”. Twenty different values for Lambda were tried, and the Mean-Squared Error for each Lambda is plotted below.



The value of Lambda with the minimum MSE was 8.839527. This Lambda value was selected for the final model. The details of the final model are shown below. The final model had an MSE of 74223.75

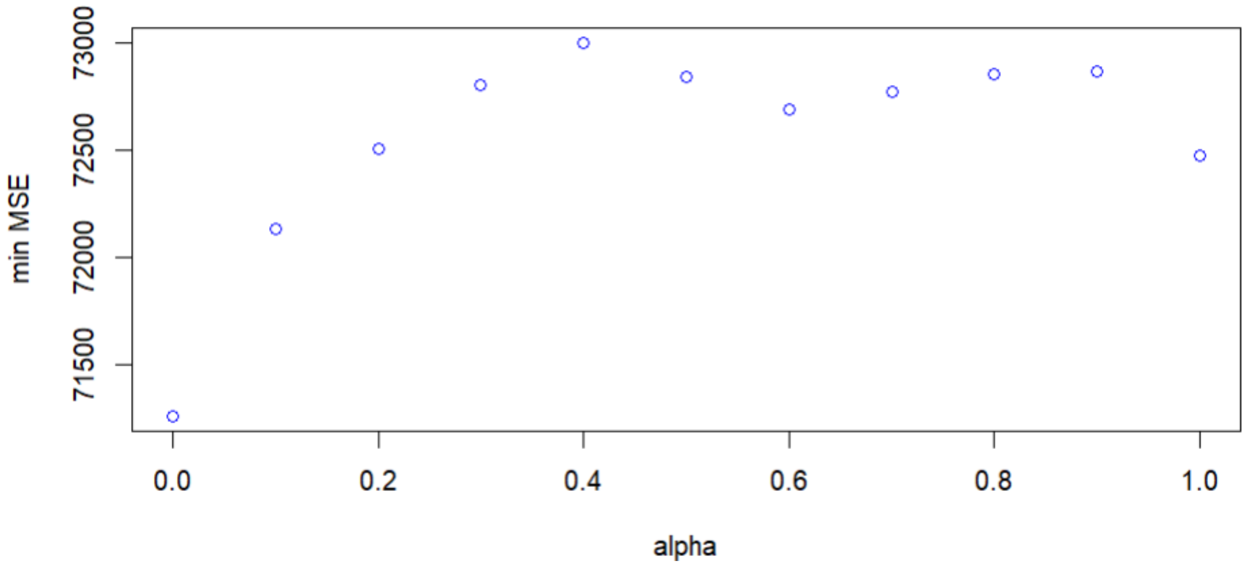
```
Call:  cv.glmnet(x = x, y = y, type.measure = "mse", nfolds = 8, alpha = 1, 
nlambda = 20, family = "gaussian", standardize = TRUE) 

Measure: Mean-Squared Error 

    Lambda Index Measure    SE Nonzero
min   8.84     8   67599 18042      11
1se  37.84     5   76654 14761       5


(Intercept) -5.072255e+03
M            7.184295e+01
So           4.466407e+01
Ed           1.253875e+02
Po1          1.023402e+02
Po2          .           
LF           .           
M.F          1.888147e+01
Pop          .           
NW           6.315089e-01
U1          -2.143645e+03
U2           8.835503e+01
Wealth       7.715072e-03
Ineq         4.882548e+01
Prob        -3.688177e+03
```

```
Time         .
```

11.1.3)

To view R code, please reference Appendix section 11.1.

Using the “glmnet” function in R, an Elastic Net regression model was created for the data in “uscrime.txt”. Various values for alpha were tried between 0 (Ridge regression) and 1 (Lasso regression). The min MSE for each alpha is plotted below. Note that the “foldid” argument was used during cross validation to ensure that the performance of the model trained with different alpha values was compared using the same cross validation data splits.



Because alpha = 0 had the lowest MSE, this was selected for the final regression model. The details of the final model are shown below. This model, which is a Ridge regression model, had an MSE of 66761.38, which is lower than the Lasso model. However, it did not exclude any variables as the Lasso regression model did.

```
Call:  cv.glmnet(x = x, y = y, type.measure = "mse", nfolds = 8, alpha = 0, 
nlambda = 20, family = "gaussian", standardize = TRUE) 

Measure: Mean-Squared Error 

    Lambda Index Measure    SE Nonzero
min  69.37    18   66761 10554      15
1se 182.90    16   71829 10821      15


(Intercept) -4.770560e+03
M            5.614442e+01
So           9.164839e+01
Ed           7.611284e+01
Po1          4.474142e+01
Po2          3.877345e+01
LF           6.300239e+02
M.F          2.355699e+01
Pop          1.861779e-01
NW           3.539245e+00
```

```
U1          -2.143101e+03
U2           8.378942e+01
Wealth       3.510454e-02
Ineq         2.781670e+01
Prob        -3.572986e+03
Time         8.716273e-01
```

12.1)

An example of a situation for which a design of experiments might be appropriate could be a research study on the impacts of lifestyle on health and longevity. A team of scientists using mice to study the impact of lifestyle on health might have too many variables to conduct every possible combination of various diets, exercise patterns, sleep, etc. on individual mice (i.e. a full factorial design). They might select a fractional factorial design of experiments to reduce the number of combinations of factors down to a level that can be achieved in their study.

12.2)

To view R code, please reference Appendix section 12.2.

The function “FrF2” was used in R to find a fractional factorial design for the real estate agent to show 16 fictitious houses with different combinations of 10 features (A through K).

The design of experiments is shown below, where a 1 indicates to include this feature, and a -1 indicates to exclude this feature.

	Feature										
Fictitious House #		A	B	C	D	E	F	G	H	J	K
	1	1	-1	-1	-1	-1	-1	1	-1	-1	-1
	2	-1	1	-1	1	-1	1	-1	-1	-1	1
	3	-1	1	-1	-1	-1	1	-1	1	1	-1
	4	-1	-1	-1	-1	1	1	1	1	-1	1
	5	-1	-1	1	-1	1	-1	-1	1	1	-1
	6	1	1	-1	1	1	-1	-1	1	-1	-1
	7	1	-1	-1	1	-1	-1	1	1	1	1
	8	1	-1	1	-1	-1	1	-1	-1	1	1
	9	-1	-1	-1	1	1	1	1	-1	1	-1
	10	1	1	1	1	1	1	1	1	1	1
	11	-1	-1	1	1	1	-1	-1	-1	-1	1
	12	-1	1	1	1	-1	-1	1	-1	1	-1
	13	1	1	1	-1	1	1	1	-1	-1	-1
	14	1	1	-1	-1	1	-1	-1	-1	1	1
	15	1	-1	1	1	-1	1	-1	1	-1	-1
	16	-1	1	1	-1	-1	-1	1	1	-1	1

13.1)

a) Binomial Distribution: Someone in a football sports betting "pick 'em" league selects who they think will win each football game every week. Assuming P is the probability of picking any individual game correctly, the number of correct picks each week should follow a binomial distribution.

b) Geometric Distribution: Geographical areas experiencing a severe drought might be interested in estimating the probability that they have to go X more days without rain. If they know the probability that it rains on any given day is P, then the probability of going X more days without rain will follow a geometric distribution.

c) Poisson Distribution: The number of orders at an online retailer for a particular item could follow a Poisson distribution, where Lambda is the average rate of orders placed for the item, and the Poisson distribution is modeling the probability that X orders are placed in a given time period. The retailer might be interested in studying this probability distribution so that they can better manage their inventory for this item.

d) Exponential Distribution: Using the example above, an exponential distribution could be used to model the time between orders.

e) **Weibull Distribution**: The Weibull distribution is a commonly used distribution in the field of reliability engineering. As an aerospace engineer, this is the distribution I am most familiar with. If a dataset contains the number of operating hours for a particular aircraft component (such as a turbine blade) prior to failure, a Weibull distribution can be fit to this data. Once this fit has been performed, various useful reliability metrics can be estimated for this component. The Weibull distribution can effectively incorporate features such as infant mortality (parts with a failure rate that decreases after an initial period) and wear out failures (parts with a failure rate that increases over time)

Appendix

11.1)

```
#Good practices (reference GA Tech office hours 5/20/24)
rm(list = ls())
set.seed(1)

data = read.table("uscrime.txt", header=T)

# Perform stepwise regression (Reference GA Tech office hours 6/17/24)
model <- lm(Crime~., data = data)

model_step <- step(model,direction = "both")

summary(model_step)

# Calculate MSE
predictions <- predict(model_step, data)
mse <- mean((predictions - data[,16])^2)


#Define x and y data
x <- as.matrix(data[,-16])
y <- as.matrix(data[,16])

#Perform lasso regression (Reference GA Tech office hours 6/17/24)
library(glmnet)
model_lasso <- cv.glmnet(x=x,
                         y=y,
                         alpha=1,
                         nfolds=8,
                         nlambda=20,
                         type.measure="mse",
                         family="gaussian",
                         standardize=TRUE)

model_lasso
#Plot cross validation MSE for each value of lambda
plot(model_lasso)
#Get the value of lambda with the smallest MSE
model_lasso$lambda.min
#Get the model coeffs at lambda = lambda min
coef(model_lasso,s=model_lasso$lambda.min)
#Get the minimum MSE
min(model_lasso$cvm)


#Elastic Net (Reference: https://glmnet.stanford.edu/articles/glmnet.html)
foldid <- sample(1:10, size = length(y), replace = TRUE)

#Create vector of alpha vals
alpha_vect <- seq(0, 1, by = 0.1)

#Create empty matrix for results vector
result <- numeric(length(alpha_vect))
```

```
#Perform cross-validation for each alpha val
for (i in 1:length(alpha_vect))
{
  cv_model <- cv.glmnet(x, y, alpha = alpha_vect[i], foldid = foldid)
  result[i] <- min(cv_model$cvm)
}

#Plot MSE vs alpha
plot(alpha_vect, result,
     xlab="alpha",
     ylab="min MSE",
     type="p",
     col="blue")

#Elastic net with alpha = 0 (ridge regression)
model_ridge <- cv.glmnet(x=x,
                         y=y,
                         alpha=0,
                         nfolds=8,
                         nlambda=20,
                         type.measure="mse",
                         family="gaussian",
                         standardize=TRUE)

model_ridge
#Plot cross validation MSE for each value of lambda
plot(model_ridge)
#Get the value of lambda with the smallest MSE
model_ridge$lambda.min
#Get the model coeffs at lambda = lambda min
coef(model_ridge,s=model_ridge$lambda.min)
#Get the minimum MSE
min(model_ridge$cvm)
```

12.2)

```
#Good practices (reference GA Tech office hours 5/20/24)
rm(list = ls())
set.seed(1)

require(FrF2)

FrF2(16, 10)
```