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

For Parts 2 and 3, remember to scale the data first – otherwise, the regression coefficients will be on different scales and the constraint won't have the desired effect.

For Parts 2 and 3, use the *glmnet* function in R.

Notes on R:

- For the elastic net model, what we called λ in the videos, *glmnet* calls "alpha"; you can get a range of results by varying alpha from 1 (lasso) to 0 (ridge regression) [and, of course, other values of alpha in between].
- In a function call like *glmnet(x,y,family="mgaussian",alpha=1)* the predictors x need to be in R's matrix format, rather than data frame format. You can convert a data frame to a matrix using *as.matrix* – for example, *x <- as.matrix(data[,1:n-1])*
- Rather than specifying a value of T, *glmnet* returns models for a variety of values of T.

11.1 ANSWER

Let's scale the data first.

```
#Read data
uscrime <- read.table("11.1uscrimeSummer2018.txt", stringsAsFactors=FALSE, header=TRUE)

#scale the data first
#do not scale the binary and the last column
uscrime2 <- as.data.frame(scale(uscrime))
uscrime_scaled <- uscrime2[,-2]
uscrime_scaled$So <- uscrime[,2] #original binary colulmn
uscrime_scaled$Crime <- uscrime$Crime #original last column
```

Doing the stepwise regression, we get these columns as factors

```
########################
#STEPWISE Regression
########################
model_l <- lm(Crime~., data=uscrime_scaled)
step(model_l, scope = list(lower = formula(lm(Crime~1,data=uscrime_scaled)),
                           upper = formula(lm(Crime~.,data=uscrime_scaled))),
                           direction = "both")

#The Stepwise Regression (variable selection method) gave me these params to use
model_lm <- lm(formula = Crime~M+Ed+Po1+M.F+U1+U2+Ineq+Prob, data=uscrime)

summary(model_lm)

cv.lm(uscrime, model_lm, m=5, plotit=TRUE)
```

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

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
```

The above result shows R-Squared value of 0.7888 and adjusted R-Squared of 0.7444.

The cross validation shows that the sum of square value is smallest at K=5 in K-fold CV. The resulting analysis shows similarity in the feature significances.

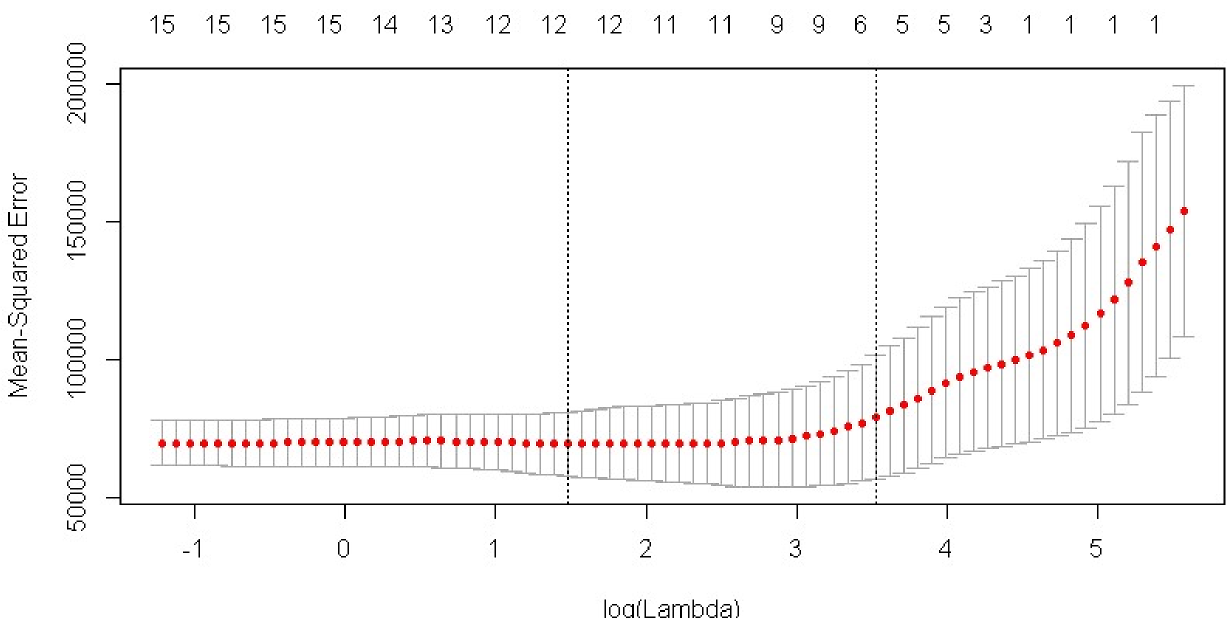
```
Analysis of Variance Table

Response: Crime
          Df  Sum Sq Mean Sq F value  Pr(>F)
M          1   55084   55084    1.44 0.23748
Ed         1  725967  725967   18.99 9.7e-05 ***
Po1        1 3173852 3173852   83.00 4.3e-11 ***
M.F        1  177521  177521    4.64 0.03759 *
U1         1       4       4    0.00 0.99191
U2         1  395014  395014   10.33 0.00267 **
Ineq       1  652440  652440   17.06 0.00019 ***
Prob       1  247978  247978    6.49 0.01505 *
Residuals 38 1453068   38239
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1



fold 1
Observations in test set: 9
              1    4    8   9  18      20   23  32   47
```

```
Predicted   730 1847 1391 686 807 1227.6  927 785 1076
cvpred      631 1789 1312 589 689 1257.6  836 796 1169
Crime       791 1969 1555 856 929 1225.0 1216 754  849
CV residual 160  180  243 267 240  -32.6  380 -42 -320

Sum of squares = 495178    Mean square = 55020    n = 9

fold 2
Observations in test set: 10
               5   13   15  17   25    34  39   40  42
Predicted   1119  754  950 440  628 980.7 798 1130 338
cvpred      1017  866 1063 257  729 978.1 873 1178 188
Crime       1234  511  798 539  523 923.0 826 1151 542
CV residual  217 -355 -265 282 -206 -55.1 -47  -27 354
              46
Predicted    786
cvpred       916
Crime        508
CV residual -408

Sum of squares = 662808    Mean square = 66281    n = 10

fold 3
Observations in test set: 10
               2   3   11    14     16   22      28   31
Predicted   1430 392 1191 780.9 942.97  673 1197.01  450
cvpred      1390 398 1067 740.4 947.44  723 1223.04  477
Crime       1635 578 1674 664.0 946.00  439 1216.00  373
CV residual  245 180  607 -76.4  -1.44 -284   -7.04 -104
              33     38
Predicted    865 577.76
cvpred       839 557.46
Crime       1072 566.00
CV residual  233   8.54

Sum of squares = 611754    Mean square = 61175    n = 10

fold 4
Observations in test set: 9
              19    21   26    27   29    30     36   44
Predicted   1195 759.8 1932 301.9 1381 711.8 1142.0 1163
cvpred      1381 816.2 1863 352.6 1655 655.4 1210.6 1188
Crime        750 742.0 1993 342.0 1043 696.0 1272.0 1030
CV residual -631 -74.2  130 -10.6 -612  40.6   61.4 -158
              45
Predicted    576
cvpred       594
Crime        455
CV residual -139

Sum of squares = 844342    Mean square = 93816    n = 9

fold 5
Observations in test set: 9
                6   7    10  12  24    35   37  41   43
Predicted   724.3 786 772.7 723 850 745.0 1012 772 1091
cvpred      693.3 742 760.1 717 775 678.6 1180 795 1207
```

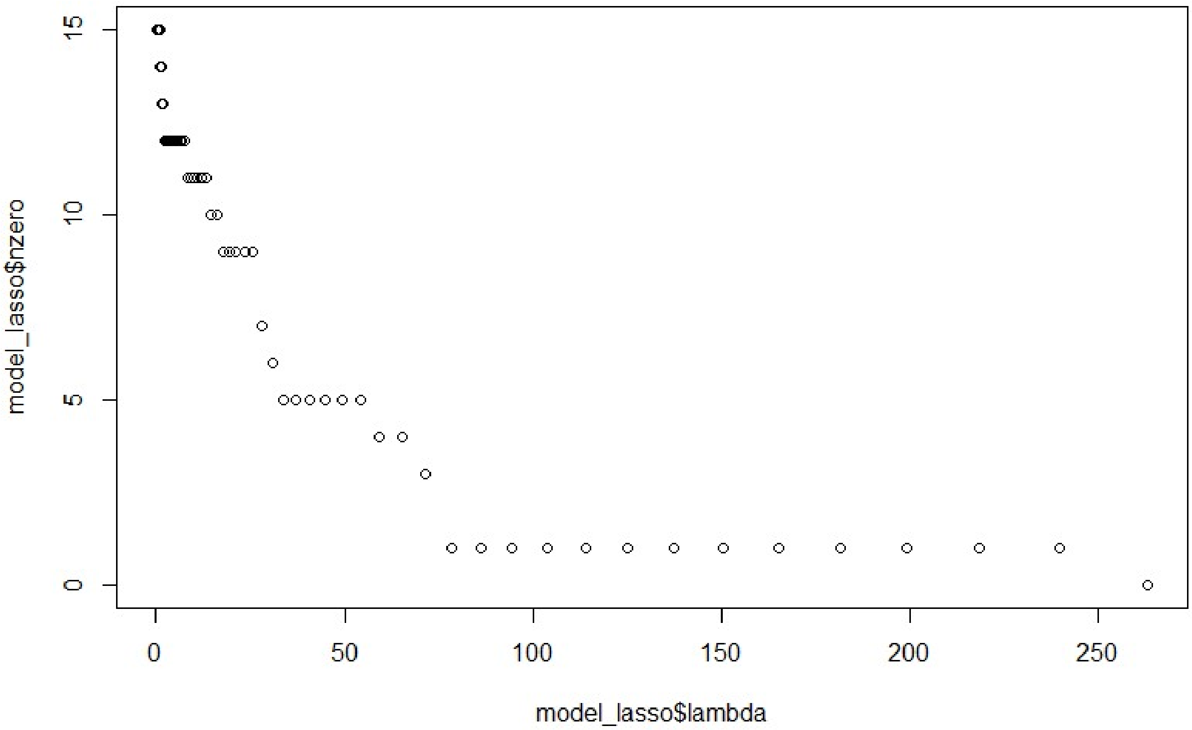
```
Crime        682.0 963 705.0 849 968 653.0  831 880  823
CV residual -11.3 221 -55.1 132 193 -25.6 -349  85 -384

Sum of squares = 383301    Mean square = 42589    n = 9
```

Moving now to LASSO, here is the code for it:

```
########################
#LASSO
########################
library(glmnet)

set.seed(1)

model_lasso <- cv.glmnet(x=as.matrix(uscrime[,-16]),
                         y=as.matrix(uscrime[,16]),
                         alpha=1,
                         nfolds=5,
                         type.measure="mse",
                         family="gaussian")
plot(model_lasso)
```



The above shows that as lambda increases, mean squared error increases. The below graph shows per lambda value how many coefficients are picked. Higher the lambda, less coefficients used.



Let's check out the LASSO model's coefficients that optimizes the lambda.

```
1.  coef(model_lasso, s=model_lasso$lambda.min)
```

```
16 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept) -5.156017e+03
M            7.289034e+01
So           4.358394e+01
Ed           1.279696e+02
Po1          1.018357e+02
Po2          .
LF           .
M.F          1.900270e+01
Pop          .
NW           6.853832e-01
U1          -2.302834e+03
U2           9.240509e+01
Wealth       1.115381e-02
Ineq         4.973944e+01
Prob        -3.692343e+03
Time         .
```

Here is Elastic Net model:

```
1.  ########################
2.  # Elastic Net
```

```
########################

set.seed(1)
count = 0
datalist = data.frame()
for (a in seq(0.1, 1, by = 0.1)){
  model_elasticnet <- cv.glmnet(x=as.matrix(uscrime[,-16]),
                               y=as.matrix(uscrime[,16]),
                               alpha=a,
                               nfolds=5,
                               type.measure="mse",
                               family="gaussian")
  l <- model_elasticnet$glmnet.fit$lambda
  dr <- model_elasticnet$glmnet.fit$dev.ratio

  df <- data.frame(lambda_min = l[88], dev_ratio = dr[88], alpha = a)

  datalist <- rbind(datalist,df)
  count = count + 1
}

datalist
```

	lambda_min	dev_ratio	alpha
1	0.80345533	0.8016304	0.1
2	0.40172767	0.8025140	0.2
3	0.26781844	0.8027642	0.3
4	0.20086383	0.8028711	0.4
5	0.16069107	0.8029270	0.5
6	0.13390922	0.8029610	0.6
7	0.11477933	0.8029827	0.7
8	0.10043192	0.8029979	0.8
9	0.08927281	0.8030073	0.9
10	0.08034553	0.8030156	1.0

Based on the lambda min and dev ratio, I can see the alpha value of 1.0 is the best in producing the model. It is the same as LASSO in this case.

Question 12.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a design of experiments approach would be appropriate.

12.1 ANSWER

Determining what affects the heart palpitation the most from number of patients can be due to several factors, but would be hard to have enough number of patients to continually monitor and check their conditions and environmental factors. It would be good to design of experiments to selectively pick patient groups from various demographics, race, age, etc.

Question 12.2

To determine the value of 10 different yes/no features to the market value of a house (large yard, solar roof, etc.), a real estate agent plans to survey 50 potential buyers, showing a fictitious house with different combinations of features. To reduce the survey size, the agent wants to show just 16 fictitious houses. Use R's *FrF2* function (in the *FrF2* package) to find a fractional factorial design for this experiment: what set of features should each of the 16 fictitious houses have? Note: the output of *FrF2* is "1" (include) or "-1" (don't include) for each feature.

12.2 ANSWER

Using FrF2 function, the below code shows how many features are included or excluded for the 16 fictitious houses:

```
rm(list=ls())
#install.packages("FrF2")
library(FrF2)
set.seed(42)
FrF2(nruns = 16, nfactors=10)
```

The result shows up like this:

```
    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
class=design, type= FrF2
```

The columns denote different features and the rows (from 1 to 16) denote each fictitious house. The values 1 and -1 tells to either include or exclude the feature in the survey.

Question 13.1

For each of the following distributions, give an example of data that you would expect to follow this distribution (besides the examples already discussed in class).

a. Binomial
b. Geometric
c. Poisson

d. Exponential
e. Weibull

13.1 ANSWER

Examples of each would be:

a. Binomial
    a. Marketing team is doing one campaign a month to drive sales up. Each month there are different campaign and each campaign reaches a certain goal. The probability of each month's sales result from the campaign is binomially distributed.
b. Geometric
    a. A hacker tries to hack into different banking systems and the number of trials between failing to breaking in.
c. Poisson
    a. Arrival of orders made at local restaurants during lunch hours are distributed i.i.d.
d. Exponential
    a. Time between the orders made at local restaurant at lunch hour.
e. Weibull
    a. A hacker tries to hack into different banking systems and the time between attempts