HW 5 (I have included my comments in # and my code)

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

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

#Stepwise Regression

#Start with 15 predctors and work its way down. Its starts with 15 and reduces to 8 and uses AIC to do that.

model_1 <-lm(Crime~., data = uscrime)

step(model_1, direction = "backward")

```
Coefficients:
(Intercept)           M          Ed         Po1         M.F          U1          U2        Ineq        Prob
   -6426.10       93.32      180.12      102.65       22.34    -6086.63      187.35       61.33    -3796.03
```

#Use the 8 factors to run a regression.

model_2 <-lm(formula = Crime~ M+Ed+Po1+M.F+U1+U2+Ineq+Prob, data = uscrime)

summary(model_2)

```
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
```

#Remove M.F & U1 since it is insignificant

model_2 <-lm(formula = Crime~ M+Ed+Po1+U2+Ineq+Prob, data = uscrime)

summary(model_2)

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
M             105.02      33.30   3.154  0.00305 ** 
Ed            196.47      44.75   4.390 8.07e-05 ***
Po1           115.02      13.75   8.363 2.56e-10 ***
U2             89.37      40.91   2.185  0.03483 *  
Ineq           67.65      13.94   4.855 1.88e-05 ***
Prob        -3801.84    1528.10  -2.488  0.01711 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 200.7 on 40 degrees of freedom
Multiple R-squared:  0.7659,	Adjusted R-squared:  0.7307 
F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

install.packages("glmnet") # for LASSO and Elastic net
library(glmnet)
set.seed(42)
datascale<-scale(uscrime)
#Lasso
model_lasso <- cv.glmnet(x=as.matrix(uscrime[,-16]), y=as.matrix(uscrime[,16]), alpha = 1, nfolds = 5, type.measure = "mse", family = "gaussian")
model_lasso

```
$lambda.min
[1] 17.71724

$lambda.1se
[1] 49.29927

attr(,"class")
[1] "cv.glmnet"
```

model_lasso$lambda #lambda is the t values or budgets.We want to pick t value that gives lowest error.
model_lasso$cvm

```
> model_lasso$lambda #lambda is the t values or budgets.We want to pick t value that gives lowest error.
 [1] 263.0953966 239.7227267 218.4264204 199.0220193 181.3414516 165.2315769 150.5528590 137.1781579 124.9916285 113.8877167
[11] 103.7702459  94.5515832  86.1518812  78.4983855  71.5248053  65.1707387  59.3811499  54.1058922  49.2992739  44.9196623
[21]  40.9291233  37.2930928  33.9800772  30.9613808  28.2108571  25.7046823  23.4211492  21.3404788  19.4446495  17.7172404
[31]  16.1432896  14.7091643  13.4024427  12.2118066  11.1269433  10.1384564   9.2377838   8.4171246   7.6693704   6.9880447
[41]   6.3672461   5.8015975   5.2861996   4.8165882   4.3886957   3.9988161   3.6435723   3.3198874   3.0249577   2.7562288
[51]   2.5113731   2.2882696   2.0849860   1.8997616   1.7309920   1.5772155   1.4371000   1.3094320   1.1931057   1.0871134
[61]   0.9905373   0.9025407   0.8223615   0.7493051   0.6827389   0.6220863   0.5668219   0.5164670   0.4705855   0.4287799
> model_lasso$cvm
 [1] 146403.92 140712.28 135089.27 128984.80 121931.30 116091.21 111257.10 107256.90 103947.86 101211.55  99056.47  97443.86
[13]  96159.04  94984.45  93391.12  91357.34  88477.78  84222.76  79891.03  76096.80  73026.20  70760.11  69371.69  68695.59
[25]  68337.95  68270.79  68340.70  68161.82  67682.72  67481.51  67707.48  68263.88  68747.49  69116.26  69787.49  70733.14
[37]  71710.68  72718.18  73796.45  74940.75  76101.91  77328.42  78578.87  79768.80  80896.15  81926.74  82855.69  83845.20
[49]  84866.33  85884.60  86902.47  87842.83  88717.25  89537.06  90297.60  91102.65  92089.00  93167.84  94199.94  95169.05
[61]  96102.33  97024.02  97908.63  98754.59  99550.90 100291.01 100975.68 101610.07 102143.30 102609.97
```

coef(model_lasso, s= model_lasso$lambda.min) #this will have the lowest mean sqaure errors

```
(Intercept) -3828.8353017
M              56.1008808
So             30.7597658
Ed             70.8167194
Po1           103.2100909
Po2             .
LF              .
M.F            16.7898439
Pop             .
NW              0.3226147
U1              .
U2             24.9099830
Wealth          .
Ineq           37.7315902
Prob        -3179.3760049
Time            .
```

model_3 <-lm(formula = Crime~ M+So+Ed+Po1+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob, data = uscrime)
summary(model_3)

```
(Intercept) -6.393e+03  1.413e+03  -4.524 7.05e-05 ***
M            8.968e+01  3.927e+01   2.284  0.02876 *
So           2.289e+01  1.253e+02   0.183  0.85621
Ed           1.749e+02  5.627e+01   3.109  0.00378 **
Po1          9.865e+01  2.187e+01   4.511 7.32e-05 ***
M.F          1.660e+01  1.633e+01   1.017  0.31656
Pop         -8.734e-01  1.199e+00  -0.729  0.47113
NW           1.863e+00  5.613e+00   0.332  0.74195
U1          -4.979e+03  3.643e+03  -1.367  0.18069
U2           1.667e+02  7.906e+01   2.108  0.04245 *
Wealth       8.633e-02  9.900e-02   0.872  0.38932
Ineq         7.163e+01  2.135e+01   3.355  0.00196 **
Prob        -4.079e+03  1.809e+03  -2.255  0.03065 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 202.6 on 34 degrees of freedom
Multiple R-squared:  0.7971,	Adjusted R-squared:  0.7255 
F-statistic: 11.13 on 12 and 34 DF,  p-value: 1.52e-08
```

#Elastic

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

model_Elastic

```
$lambda.min
[1] 11.60319

$lambda.1se
[1] 42.68096

attr(,"class")
[1] "cv.glmnet"
```

model_Elastic$lambda #lambda is the t values or budgets.We want to pick t value that gives lowest error.

model_Elastic$cvm

```
> model_Elastic$lambda #lambda is the t values or budgets.We want to pick t value that gives lowest error.
 [1] 526.1907933 479.4454535 436.8528408 398.0440385 362.6829032 330.4631537 301.1057180 274.3563159 249.9832570 227.7754334
[11] 207.5404917 189.1031664 172.3037623 156.9967710 143.0496106 130.3414774 118.7622998 108.2117845  98.5985478  89.8393245
[21]  81.8582466  74.5861856  67.9601544  61.9227616  56.4217141  51.4093646  46.8422983  42.6809576  38.8892990  35.4344809
[31]  32.2865792  29.4183285  26.8048853  24.4236132  22.2538867  20.2769127  18.4755677  16.8342492  15.3387409  13.9760894
[41]  12.7344922  11.6031950  10.5723991   9.6331763   8.7773915   7.9976322   7.2871446   6.6397748   6.0499155   5.5124577
[51]   5.0227461   4.5765392   4.1699721   3.7995232   3.4619841   3.1544309   2.8742000   2.6188640   2.3862113   2.1742269
[61]   1.9810746   1.8050814   1.6447229   1.4986103   1.3654779   1.2441726   1.1336437   1.0329339   0.9411709   0.8575599
[71]   0.7813766   0.7119613   0.6487126   0.5910828   0.5385726   0.4907273   0.4471324   0.4074104
> model_Elastic$cvm
 [1] 149625.72 146235.01 141410.17 134461.08 128124.68 122588.08 117824.93 113745.75 110354.42 107793.67 105983.63 104743.87
[13] 103797.11 103126.45 102736.90 102287.92 100683.61  98511.04  96376.61  93940.52  91400.46  88893.98  86330.24  83963.75
[25]  81468.88  79086.75  76742.70  74249.60  72124.56  70391.48  68848.49  67510.54  66409.80  65511.68  64741.12  64029.00
[37]  63443.45  62950.03  62577.11  62333.41  62185.51  62127.22  62171.64  62238.60  62409.74  62671.18  62948.48  63235.78
[49]  63552.84  63910.79  64322.79  64743.39  65168.30  65578.49  65962.82  66329.33  66672.59  67011.38  67372.25  67689.96
[61]  67990.89  68250.50  68409.98  68546.12  68684.21  68824.65  68966.41  69109.63  69256.49  69505.09  69693.50  69835.96
[73]  69975.43  70110.07  70238.25  70353.65  70466.05  70573.11
```

coef(model_Elastic, s= model_Elastic$lambda.min) #this will have the lowest mean sqaure errors

```
                        1
(Intercept) -5.448218e+03
M            7.317505e+01
So           4.726900e+01
Ed           1.285266e+02
Po1          8.286051e+01
Po2          1.379667e+01
LF           2.007735e+01
M.F          2.139471e+01
Pop         -2.114354e-03
NW           1.599082e+00
U1          -3.038527e+03
U2           1.078048e+02
Wealth       3.039998e-02
Ineq         4.969286e+01
Prob        -3.823196e+03
Time         .
```

model_4 <-lm(formula = Crime~ M+So+Ed+Po1+Po2+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob, data = uscrime)

summary(model_4)

```
M            8.743e+01  3.964e+01   2.205 0.034514 *
So           3.440e+01  1.271e+02   0.271 0.788398
Ed           1.809e+02  5.721e+01   3.163 0.003346 **
Po1          1.688e+02  9.667e+01   1.746 0.090115 .
Po2         -7.692e+01  1.032e+02  -0.745 0.461484
M.F          1.474e+01  1.663e+01   0.887 0.381622
Pop         -9.510e-01  1.211e+00  -0.785 0.437837
NW           2.422e+00  5.699e+00   0.425 0.673604
U1          -4.805e+03  3.674e+03  -1.308 0.200017
U2           1.622e+02  7.982e+01   2.032 0.050269 .
Wealth       8.501e-02  9.967e-02   0.853 0.399833
Ineq         6.912e+01  2.175e+01   3.177 0.003219 **
Prob        -4.185e+03  1.826e+03  -2.292 0.028430 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 204 on 33 degrees of freedom
Multiple R-squared:  0.8005,    Adjusted R-squared:  0.7219
F-statistic: 10.19 on 13 and 33 DF,  p-value: 4.088e-08
```

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.

For example, lets say that I want to introduce a new organic vegan cereal product to the market. I try to create different kinds of packaging for my cereal to get an idea of which packaging would be appealing to the consumers. I can use DOE to understand the effect of packaging on a set of consumers to understand their buying behavior.

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.

#Need two inputs (nruns = 16 fictious houses and nfactors =10 features). Below is the result of the fractional factorial design.

FrF2(16, 10)

```
> FrF2(16, 10)
    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
```

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 : The binomial is a type of distribution that has two possible outcomes. Data on students overall grade in a course explained as either a pass or a fail.

b. Geometric: you ask people outside a polling station who they voted for until you find someone that voted for the independent candidate in a local election. The geometric distribution would represent the number of people who you had to poll before you found someone who voted independent.

c. Poisson: Given the number of diners in a certain restaurant every day, if the average number of diners for seven days is 500, you can predict the probability of a certain day having more customers.

d. Exponential: Let’s say a Poisson distribution models the number of births in a given time period. The time in between each birth can be modeled with an exponential distribution

e. Weibull: How long will it take for a TV to become defective since the time it has been switched on.