Week 5 Homework

loading packages and setting seed for homework

```
library('tidyverse')
library('glmnet')
library('MASS')
library('FrF2')



#Set seed for reproducibility
set.seed(156)
```

Question 1:

Using the crime data set from Homework 3, 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.

```
#Crime data file
crimeData <- read.table("http://www.statsci.org/data/general/uscrime.txt", header = TRUE)

#serparting out dataset
x <- as.matrix(crimeData[,1:15])
y <- as.double(as.matrix(crimeData[,16]))
```

Question 1.1

creating stepwise regression model. Chose a backward direction to start with all variables and reduce. Stepwise regression does not require scaling, therefore did not transform the data. Stepwise model is eliminating variables based on AIC value. A lower AIC is better.

```
#1
#creating stepwise regression model. Chose a backward direction to start with all variables and reduce.
#stepwise regression
stepwiseMod <- step(lm(Crime~.,data=crimeData),direction="backward")
```

```
## Start:  AIC=514.65
## Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob + Time
##
##          Df Sum of Sq     RSS    AIC
## - So      1        29 1354974 512.65
## - LF      1      8917 1363862 512.96
## - Time    1     10304 1365250 513.00
## - Pop     1     14122 1369068 513.14
## - NW      1     18395 1373341 513.28
```

```
## - M.F     1     31967 1386913 513.74
## - Wealth  1     37613 1392558 513.94
## - Po2     1     37919 1392865 513.95
## <none>                1354946 514.65
## - U1      1     83722 1438668 515.47
## - Po1     1    144306 1499252 517.41
## - U2      1    181536 1536482 518.56
## - M       1    193770 1548716 518.93
## - Prob    1    199538 1554484 519.11
## - Ed      1    402117 1757063 524.86
## - Ineq    1    423031 1777977 525.42
##
## Step:  AIC=512.65
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob + Time
##
##          Df Sum of Sq     RSS    AIC
## - Time    1     10341 1365315 511.01
## - LF      1     10878 1365852 511.03
## - Pop     1     14127 1369101 511.14
## - NW      1     21626 1376600 511.39
## - M.F     1     32449 1387423 511.76
## - Po2     1     37954 1392929 511.95
## - Wealth  1     39223 1394197 511.99
## <none>                1354974 512.65
## - U1      1     96420 1451395 513.88
## - Po1     1    144302 1499277 515.41
## - U2      1    189859 1544834 516.81
## - M       1    195084 1550059 516.97
## - Prob    1    204463 1559437 517.26
## - Ed      1    403140 1758114 522.89
## - Ineq    1    488834 1843808 525.13
##
## Step:  AIC=511.01
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - LF      1     10533 1375848 509.37
## - NW      1     15482 1380797 509.54
## - Pop     1     21846 1387161 509.75
## - Po2     1     28932 1394247 509.99
## - Wealth  1     36070 1401385 510.23
## - M.F     1     41784 1407099 510.42
## <none>                1365315 511.01
## - U1      1     91420 1456735 512.05
## - Po1     1    134137 1499452 513.41
## - U2      1    184143 1549458 514.95
## - M       1    186110 1551425 515.01
## - Prob    1    237493 1602808 516.54
## - Ed      1    409448 1774763 521.33
## - Ineq    1    502909 1868224 523.75
##
## Step:  AIC=509.37
```

```
## Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + NW + U1 + U2 + Wealth +
##     Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - NW      1     11675 1387523 507.77
## - Po2     1     21418 1397266 508.09
## - Pop     1     27803 1403651 508.31
## - M.F     1     31252 1407100 508.42
## - Wealth  1     35035 1410883 508.55
## <none>                1375848 509.37
## - U1      1     80954 1456802 510.06
## - Po1     1    123896 1499744 511.42
## - U2      1    190746 1566594 513.47
## - M       1    217716 1593564 514.27
## - Prob    1    226971 1602819 514.54
## - Ed      1    413254 1789103 519.71
## - Ineq    1    500944 1876792 521.96
##
## Step:  AIC=507.77
## Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + U1 + U2 + Wealth + Ineq +
##     Prob
##
##          Df Sum of Sq     RSS    AIC
## - Po2     1     16706 1404229 506.33
## - Pop     1     25793 1413315 506.63
## - M.F     1     26785 1414308 506.66
## - Wealth  1     31551 1419073 506.82
## <none>                1387523 507.77
## - U1      1     83881 1471404 508.52
## - Po1     1    118348 1505871 509.61
## - U2      1    201453 1588976 512.14
## - Prob    1    216760 1604282 512.59
## - M       1    309214 1696737 515.22
## - Ed      1    402754 1790276 517.74
## - Ineq    1    589736 1977259 522.41
##
## Step:  AIC=506.33
## Crime ~ M + Ed + Po1 + M.F + Pop + U1 + U2 + Wealth + Ineq +
##     Prob
##
##          Df Sum of Sq     RSS    AIC
## - Pop     1     22345 1426575 505.07
## - Wealth  1     32142 1436371 505.39
## - M.F     1     36808 1441037 505.54
## <none>                1404229 506.33
## - U1      1     86373 1490602 507.13
## - U2      1    205814 1610043 510.76
## - Prob    1    218607 1622836 511.13
## - M       1    307001 1711230 513.62
## - Ed      1    389502 1793731 515.83
## - Ineq    1    608627 2012856 521.25
## - Po1     1   1050202 2454432 530.57
##
## Step:  AIC=505.07
```

```
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Wealth + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - Wealth  1     26493 1453068 503.93
## <none>                1426575 505.07
## - M.F     1     84491 1511065 505.77
## - U1      1     99463 1526037 506.24
## - Prob    1    198571 1625145 509.20
## - U2      1    208880 1635455 509.49
## - M       1    320926 1747501 512.61
## - Ed      1    386773 1813348 514.35
## - Ineq    1    594779 2021354 519.45
## - Po1     1   1127277 2553852 530.44
##
## Step:  AIC=503.93
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob
##
##        Df Sum of Sq     RSS    AIC
## <none>              1453068 503.93
## - M.F   1    103159 1556227 505.16
## - U1    1    127044 1580112 505.87
## - Prob  1    247978 1701046 509.34
## - U2    1    255443 1708511 509.55
## - M     1    296790 1749858 510.67
## - Ed    1    445788 1898855 514.51
## - Ineq  1    738244 2191312 521.24
## - Po1   1   1672038 3125105 537.93
```

```
#comparison of models. AIC on initial model with all variables was 515, ended up at 504.
stepwiseMod$anova
```

```
##       Step Df    Deviance Resid. Df Resid. Dev      AIC
## 1           NA          NA        31    1354946 514.6488
## 2     - So  1    28.57405        32    1354974 512.6498
## 3   - Time  1 10340.66984        33    1365315 511.0072
## 4     - LF  1 10533.15902        34    1375848 509.3684
## 5     - NW  1 11674.63991        35    1387523 507.7655
## 6    - Po2  1 16706.34095        36    1404229 506.3280
## 7    - Pop  1 22345.36638        37    1426575 505.0700
## 8 - Wealth  1 26493.24677        38    1453068 503.9349
```

```
# The final formula and coefficients with the optimal AIC is:
stepwiseMod$coefficients
```

```
## (Intercept)           M          Ed         Po1         M.F          U1
## -6426.10102    93.32155   180.12011   102.65316    22.33975 -6086.63315
##          U2        Ineq        Prob
##   187.34512    61.33494 -3796.03183
```

```
#creating test dataframe to compare prediction results to actuals
stepwiseTest <- as.data.frame(crimeData[,16])
stepwiseTest$pred <- predict(stepwiseMod)


summary(stepwiseMod)
```

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

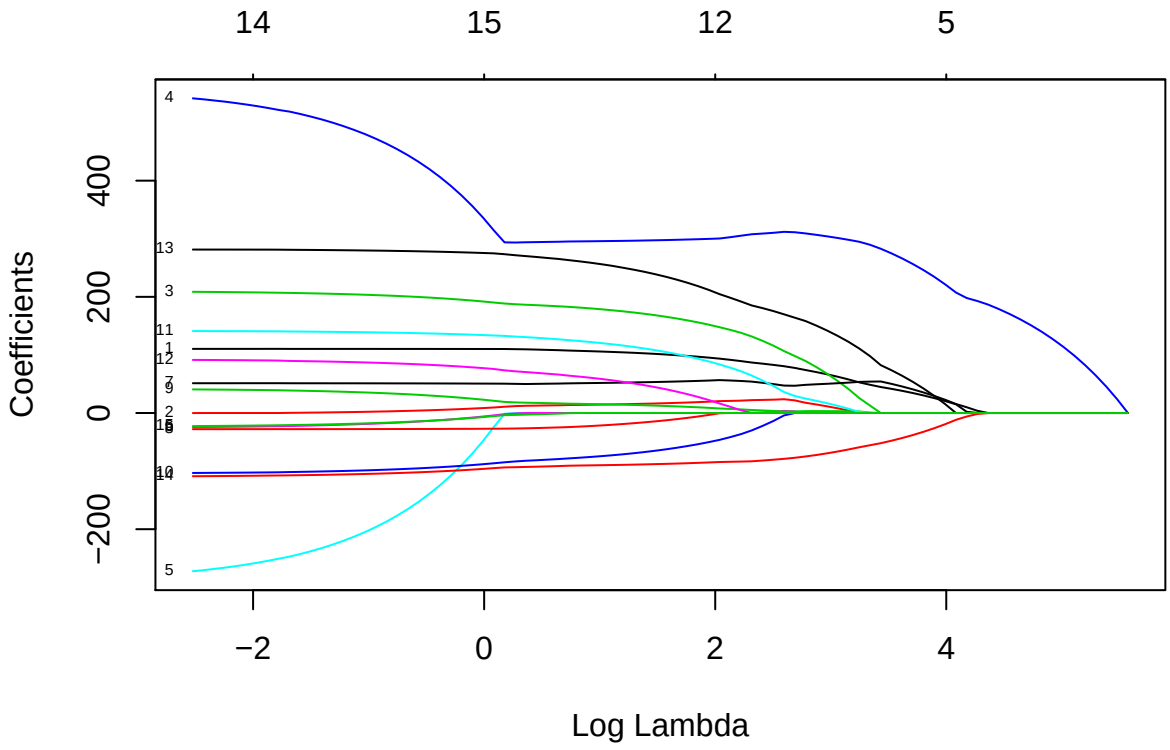
Question 1.2

Lasso Model

```
#scaling data
xScaled <- scale(x, center = TRUE, scale = TRUE)

Mod.Lasso <- glmnet(xScaled,y,family='gaussian', standardize = TRUE, alpha=1)

#plotting model
plot(Mod.Lasso,xvar="lambda",label=TRUE)
```




```
#summary
summary(Mod.Lasso)
```

```
##           Length Class     Mode
## a0           88   -none-    numeric
## beta       1320   dgCMatrix S4
## df           88   -none-    numeric
## dim           2   -none-    numeric
## lambda       88   -none-    numeric
## dev.ratio    88   -none-    numeric
## nulldev       1   -none-    numeric
## npasses       1   -none-    numeric
## jerr          1   -none-    numeric
## offset        1   -none-    logical
## call          6   -none-    call
## nobs          1   -none-    numeric
```

```
#creating prediction
yhat  <- predict(Mod.Lasso,newx <- xScaled, s <- Mod.Lasso$lambda.1se)
mse <- mean((y - yhat)^2)
mse
```

```
## [1] 46936.6
```

```
#sum of squares
sst <- sum((y - mean(y))^2)

#sum of Errors
```

```r
sse <- sum((yhat - y)^2)

#r squared
r <- 1 - sst / sse
r
```

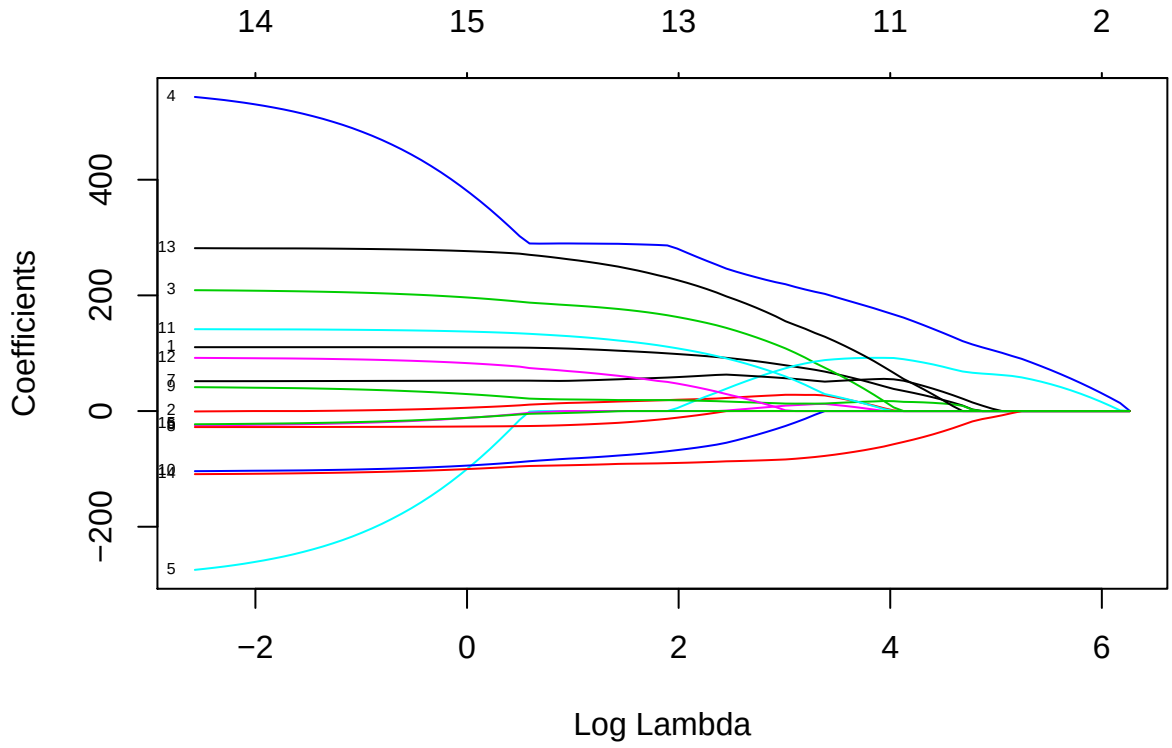
```
## [1] 0.964555
```

Question 1.3

elastic net model

```r
Mod.Elnet <- glmnet(xScaled,y,family='gaussian', standardize = TRUE, alpha=.5)

#plotting model
plot(Mod.Elnet,xvar="lambda",label=TRUE)
```




```r
summary(Mod.Elnet)
```

```
##           Length Class     Mode
## a0          96   -none-    numeric
## beta      1440   dgCMatrix S4
## df          96   -none-    numeric
## dim          2   -none-    numeric
## lambda      96   -none-    numeric
## dev.ratio   96   -none-    numeric
```

```
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         6   -none-    call
## nobs         1   -none-    numeric
```

```
#creating prediction
yhat2  <- predict(Mod.Elnet,newx2 <- xScaled, s2 <- Mod.Elnet$lambda)

#mean square error
mse2 <- mean((y - yhat2)^2)
mse2
```

```
## [1] 47395.16
```

```
#sum of squares
sst2 <- sum((y - mean(y))^2)

#sum of Errors
sse2 <- sum((yhat2 - y)^2)

#r squared
r2 <- 1 - sst2 / sse2
r2
```

```
## [1] 0.9678231
```

Question 2

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

A DOE approach would be appropriate in my job to determine how fraud rates vary based on screen resolution of the device the customer is using to initiate an order. Screen resolution is a data point available to us on each order. As the ordering process is the same for all customers regardless of being fraud or not the experiment would already be controlled. We could further control the experiment by comparing customers from a certain country or type of customer such as average spend amount. The eventual outcome would be to see if there was a relationship to screen resolution and fraud rate.

Question 3

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?

```
#Fractional Factorial design for house
house <- FrF2(16, 10,default.levels = c("no", "yes"))
house
```

```
##      A   B   C   D   E   F   G   H   J   K
## 1  yes  no  no  no  no  no yes  no  no  no
## 2  yes yes  no  no yes  no  no  no yes yes
## 3  yes yes yes yes yes yes yes yes yes yes
```

```
## 4  yes  no  no yes  no  no yes yes yes yes
## 5  yes yes yes  no yes yes yes  no  no  no
## 6   no yes yes yes  no  no yes  no yes  no
## 7   no  no yes  no yes  no  no yes yes  no
## 8   no  no  no  no yes yes yes yes  no yes
## 9  yes yes  no yes yes  no  no yes  no  no
## 10  no yes  no  no  no yes  no yes yes  no
## 11 yes  no yes yes  no yes  no yes  no  no
## 12  no yes  no yes  no yes  no  no  no yes
## 13  no yes yes  no  no  no yes yes  no yes
## 14  no  no yes yes yes  no  no  no  no yes
## 15  no  no  no yes yes yes yes  no yes  no
## 16 yes  no yes  no  no yes  no  no yes yes
## class=design, type= FrF2
```

Question 4

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

Binomial

A binomial distribution is n independent experiments with a boolean outcome such as true/false, 1/0. An example of this type of distrubution would be a survey question asking a set N number of people if they voted for Donald Trump. This would be a yes/no outcome, therefore binomial.

Geometric

A geometric distribution is showing how many trials before we get to what we are looking for. An example of this would be how many licks it takes to get to the center of a tootsie pop. Each lick would carry a certain probability of actaully hitting the center, which each lick we would be closer to the eventual outcome.

Poisson

A Poisson distribution is N number of occurances that are completely independent of each other and the frequency of occurance is known and the number of occurances in the time frequencies can be counted. A good example of this would be how many calls a call center agent receives in a 1 hour time frame.

Exponential

An Exponential distribution describes the arrival time of a randomly recurring independent event sequence. An example of this is if the mean time of a phone call for a customer service agent is 10 minutes, an exponential distrubution can be used to determine the probability of a phone call that lasts only 8 minutes.

Weibull

The Weibull distribution is very useful for modeling the amount of time it takes something to fail, specifically the time between failures. I am going to go back to my geometric example as it also applies in a similar way. Weibull could determine the amount of time it takes to lick your way to the center of a tootsie pop.