Homework 5

6/17/2020

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.

```
set.seed(8010)
uscrimes <- read.delim('uscrime.txt')
```

Stepwise Regression

To compute stepwise regression, I'll use the stepAIC function in the MASS package and use AIC to define if a feature is good enough to keep in the model. I've also created a full model using all the predictors to compare the models to.

```
# The Full Model
full_model <- lm(Crime ~., data = uscrimes)

#Stepwise Regression Model
stepwise_model <- stepAIC(full_model, direction = "both")
```

```
## 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
## + So      1        29 1354946 514.65
## - 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
## + Time    1     10341 1354974 512.65
## + So      1        65 1365250 513.00
## - 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
## + LF      1     10533 1365315 511.01
## + Time    1      9996 1365852 511.03
## + So      1      3046 1372802 511.26
## - 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
## + NW      1     11675 1375848 509.37
## + So      1      7207 1380316 509.52
## + LF      1      6726 1380797 509.54
## + Time    1      4534 1382989 509.61
## - 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
## + Po2     1     16706 1387523 507.77
## + NW      1      6963 1397266 508.09
## + So      1      3807 1400422 508.20
## + LF      1      1986 1402243 508.26
## + Time    1       575 1403654 508.31
## - 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
## + Pop     1     22345 1404229 506.33
## + Po2     1     13259 1413315 506.63
## + NW      1      5927 1420648 506.87
## + So      1      5724 1420851 506.88
## + LF      1      5176 1421398 506.90
## + Time    1      3913 1422661 506.94
## - 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
## + Wealth  1     26493 1426575 505.07
## - M.F     1    103159 1556227 505.16
## + Pop     1     16697 1436371 505.39
## + Po2     1     14148 1438919 505.47
## + So      1      9329 1443739 505.63
## + LF      1      4374 1448694 505.79
## + NW      1      3799 1449269 505.81
## + Time    1      2293 1450775 505.86
## - 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
```

The last model generated the lowest AIC of 503.93 has M, Ed, Po1, M.F, U1, U2, Ineq and Prob as predcictors.

Let's see what it's r squared value is. For comparison, I'm also showing summary metrics for the full model with all the predictors.

```
final_step_model <- lm(Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob, data = uscrimes)
summary(final_step_model)
```

```
##
```

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

```
summary(full_model)
```

```
##
## Call:
## lm.default(formula = Crime ~ ., data = uscrimes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

The reduced model with predictors chosen through stepwise regression has a really good adjusted r squared value compared to the full model. As we've noticed in previous homework assigments, these models are likely to be overfit because of the small amount of data available.

Interestingly, I used regsubsets function in the leaps package in a previous homework assignment to find the best predictors and came up with the exact same list of predictors that stepAIC came up with.

LASSO

Before using the LASSO method, I will scale the data and split it into training and test sets.

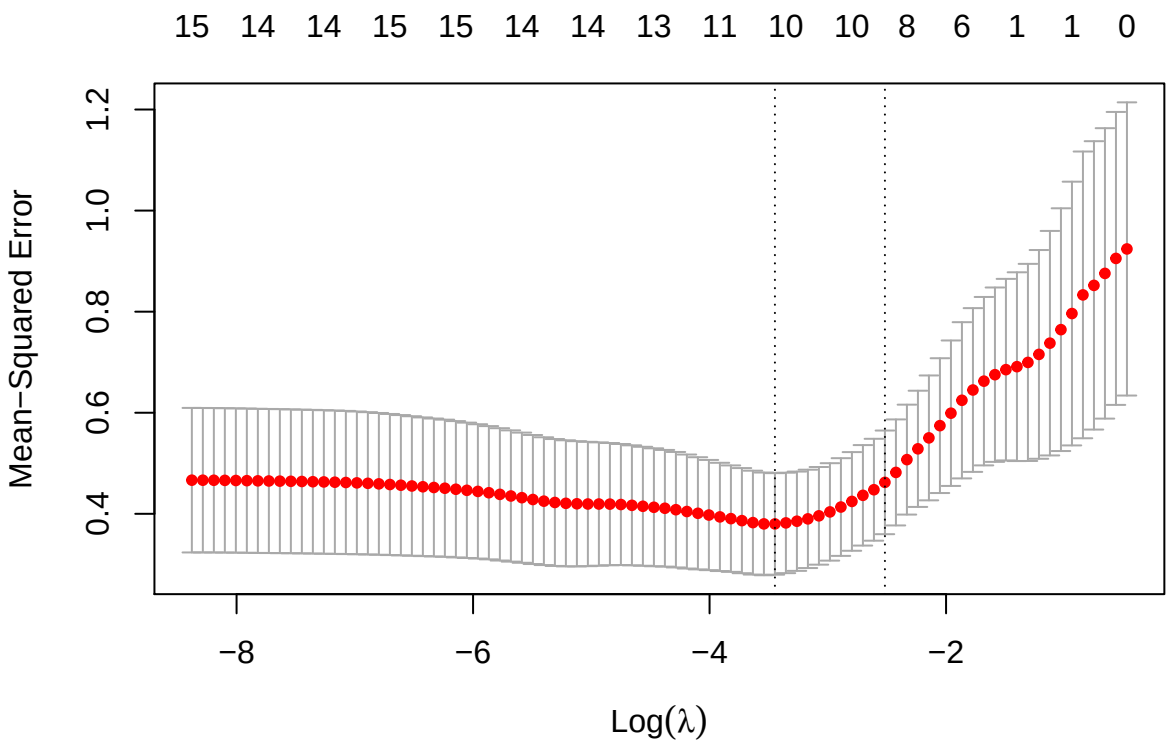
```
set.seed(2184)
uscrimes_scaled <-  scale(uscrimes)
train_sample <- createDataPartition(y = uscrimes_scaled[,16], p = 0.7, list = FALSE)
training <- uscrimes_scaled[train_sample, ]
test <- uscrimes_scaled[-train_sample, ]

x_train <- as.matrix(training[, 1:15])
y_train <- as.matrix(training[, 16])
x_test <- as.matrix(test[, 1:15])
y_test <- as.matrix(test[, 16])
```

Generating the LASSO model:

```
lasso_model <- glmnet(x = x_train, y = y_train, alpha = 1, family = "mgaussian")
```

Now I'll use cross validation to determine what value of lambda to use. Ideally, I want the smallest value of lambda as it gives the minimum cross validated error.

```
lasso_model_cv = cv.glmnet(x_train, y_train, alpha = 1)
plot(lasso_model_cv)
```




```
lasso_model_cv$lambda.min
```

```
## [1] 0.03182821
```

The optimal lambda is 0.03182821. Now I'll get the model's coefficients using lambda = 0.03182821 as the penalty parameter.

```
coef(lasso_model_cv, s = lasso_model_cv$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept)  0.08566297
## M            0.20015315
## So           0.05760064
## Ed           0.37270393
## Po1          0.77936943
## Po2          .
## LF           .
## M.F          0.16837753
## Pop          .
## NW           0.27606796
## U1           .
## U2           0.07870377
## Wealth       .
```

```
## Ineq          0.41827876
## Prob         -0.14756115
## Time          0.11388116
```

Looking at these results, it looks like Po1, Ineq and Ed are among the top significant predictors and Po2, LF, Pop, U1 and Wealth are dropped by the model.

Let's use this model on our test data and calculate r squared to see how well it performs:

```
yhat_lasso <- predict(lasso_model_cv, s = lasso_model_cv$lambda.min,  newx = x_test)
ss_res = sum((yhat_lasso - y_test)^2)
ss_total <- sum((y_test- mean(y_test))^2)
R2 = 1 - ss_res/ss_total
R2
```

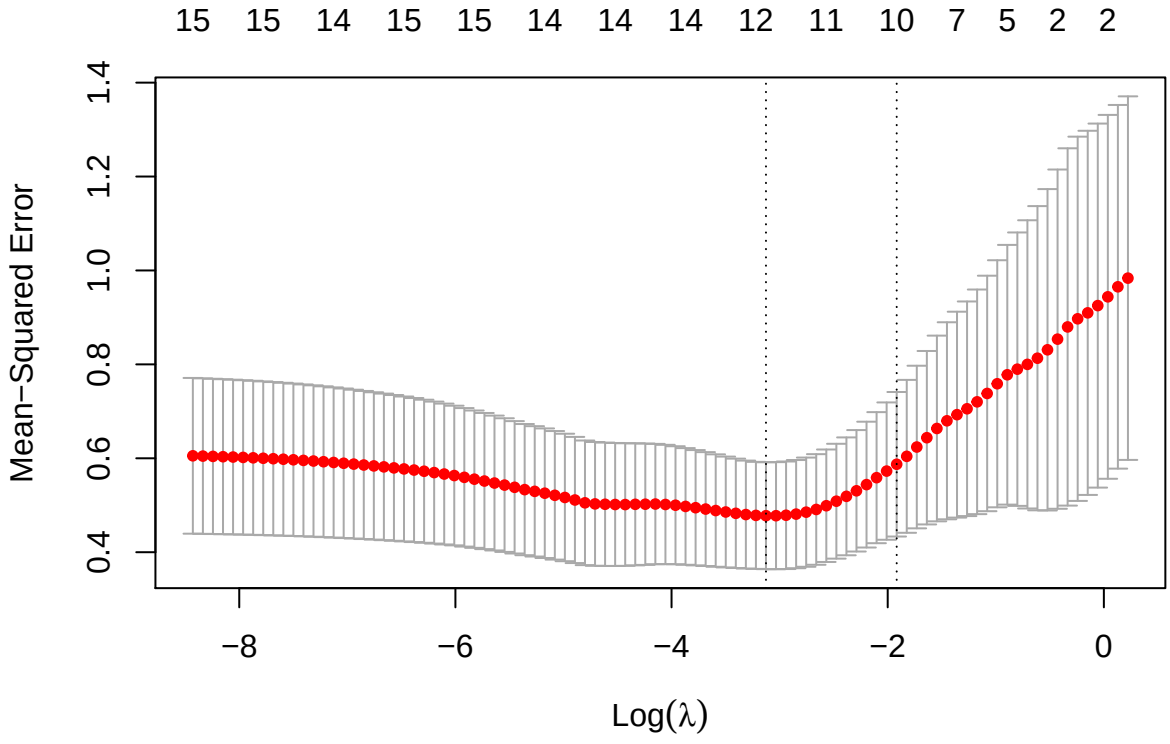
```
## [1] 0.4884978
```

Elastic Net

For this, I'll use the same training and test data I defined for LASSO.

Generating the Elastic Net model using the same function as LASSO but setting alpha = 0.5. I'll also do cross validation to determine the optimal value of lambda.

```
set.seed(2184)
elnet <- glmnet(x = x_train, y = y_train, alpha = 0.5, family = "mgaussian")
elnet_cv <- cv.glmnet(x = x_train, y_train, alpha = 0.5)
plot(elnet_cv)
```



```
elnet_cv$lambda.min
```

```
## [1] 0.0438759
```

The optimal lambda here is 0.0438759, larger than that of LASSO. Let's see what predictors this model chooses based on this lambda:

```
coef(elnet_cv, s = elnet_cv$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept)  0.0886884808
## M            0.2101027257
## So           0.0839415703
## Ed           0.3853152197
## Po1          0.5859162090
## Po2          0.1829065106
## LF           .
## M.F          0.1855985539
## Pop          .
## NW           0.2738954285
## U1          -0.0001587682
## U2           0.0912171136
## Wealth       .
## Ineq         0.4064895971
## Prob        -0.1501238497
## Time         0.1352608863
```

Interesting! This model kept more predictors than the LASSO model and excluded the LF, Pop and Wealth like the LASSO model.

Let's calculate r squared to compare to the LASSO model:

```
yhat_lasso <- predict(elnet_cv, s = lasso_model_cv$lambda.min,  newx = x_test)
ss_res = sum((yhat_lasso - y_test)^2)
ss_total <- sum((y_test- mean(y_test))^2)
R2 = 1 - ss_res/ss_total
R2
```

```
## [1] 0.4660338
```

The two r squared values are pretty close. However, even with more predictors, it looks like the elastic net doesn't perform as well as the LASSO function.

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.

I regularly use A/B testing in my current position as an analyst for a loyalty program. We're always testing different subject lines and/or creative versions in emails to see which combination will generate higher open and click rates. The data science team at my company has actually built multi armed bandit models for some of our participating retailers to help them manage their CRM strategy more effectively. Ideally, we're sending the right creative and offer for the right member at the right time in the right channel using the learnings from the model.

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.

Using FrF2, the columns would be the features and rows would be the houses.

```
set.seed(8010)
fac_design <- FrF2(nruns = 16, nfactors = 10)
fac_design
```

```
##     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 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 probability that a car sales person will sell a car or not. Let p = probability of selling a car and n be the number of clients the sales person talks to in a work day.
b. Geometric - the probability of a car sales person not selling a car x times before selling a car.
c. Poisson - the probability that x number of test drives will be done today, given the dealership's average number of daily test drives.
d. Exponential - estimating the amount of time between each test drive so that the dealership can clean and refuel the car.
e. Weibull - estimating the amount of time it takes to sell a car.