Intro to Analytics Modeling HW 5

2024-06-19

Importing Libraries

```r
library(FrF2)
```

```
## Loading required package: DoE.base

## Loading required package: grid

## Loading required package: conf.design

## Registered S3 method overwritten by 'DoE.base':
##   method           from
##   factorize.factor conf.design

##
## Attaching package: 'DoE.base'

## The following objects are masked from 'package:stats':
##
##     aov, lm

## The following object is masked from 'package:graphics':
##
##     plot.design

## The following object is masked from 'package:base':
##
##     lengths
```

```r
library(glmnet)  # for efficient procedures for fitting the entire lasso or elastic-net
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```r
library(MASS)  # for computing stepwise regression
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice

## Registered S3 method overwritten by 'lava':
##   method            from
##   print.equivalence partitions
```

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.

Read Data

```
crime_data <- read.table("~/Desktop/ISYE-6501/week 5 Homework-Summer24/uscrime.txt", stringsAsFactors =
crime_data
```

```
##       M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1  15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2  14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3  14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4  13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5  14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6  12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
## 7  12.7  1 11.1  8.2  7.9 0.519  98.2   4 13.9 0.097 3.8   6200 16.8 0.042100
## 8  13.1  1 10.9 11.5 10.9 0.542  96.9  50 17.9 0.079 3.5   4720 20.6 0.040099
## 9  15.7  1  9.0  6.5  6.2 0.553  95.5  39 28.6 0.081 2.8   4210 23.9 0.071697
## 10 14.0  0 11.8  7.1  6.8 0.632 102.9   7  1.5 0.100 2.4   5260 17.4 0.044498
## 11 12.4  0 10.5 12.1 11.6 0.580  96.6 101 10.6 0.077 3.5   6570 17.0 0.016201
## 12 13.4  0 10.8  7.5  7.1 0.595  97.2  47  5.9 0.083 3.1   5800 17.2 0.031201
## 13 12.8  0 11.3  6.7  6.0 0.624  97.2  28  1.0 0.077 2.5   5070 20.6 0.045302
## 14 13.5  0 11.7  6.2  6.1 0.595  98.6  22  4.6 0.077 2.7   5290 19.0 0.053200
## 15 15.2  1  8.7  5.7  5.3 0.530  98.6  30  7.2 0.092 4.3   4050 26.4 0.069100
## 16 14.2  1  8.8  8.1  7.7 0.497  95.6  33 32.1 0.116 4.7   4270 24.7 0.052099
## 17 14.3  0 11.0  6.6  6.3 0.537  97.7  10  0.6 0.114 3.5   4870 16.6 0.076299
## 18 13.5  1 10.4 12.3 11.5 0.537  97.8  31 17.0 0.089 3.4   6310 16.5 0.119804
## 19 13.0  0 11.6 12.8 12.8 0.536  93.4  51  2.4 0.078 3.4   6270 13.5 0.019099
## 20 12.5  0 10.8 11.3 10.5 0.567  98.5  78  9.4 0.130 5.8   6260 16.6 0.034801
## 21 12.6  0 10.8  7.4  6.7 0.602  98.4  34  1.2 0.102 3.3   5570 19.5 0.022800
## 22 15.7  1  8.9  4.7  4.4 0.512  96.2  22 42.3 0.097 3.4   2880 27.6 0.089502
## 23 13.2  0  9.6  8.7  8.3 0.564  95.3  43  9.2 0.083 3.2   5130 22.7 0.030700
## 24 13.1  0 11.6  7.8  7.3 0.574 103.8   7  3.6 0.142 4.2   5400 17.6 0.041598
## 25 13.0  0 11.6  6.3  5.7 0.641  98.4  14  2.6 0.070 2.1   4860 19.6 0.069197
## 26 13.1  0 12.1 16.0 14.3 0.631 107.1   3  7.7 0.102 4.1   6740 15.2 0.041698
## 27 13.5  0 10.9  6.9  7.1 0.540  96.5   6  0.4 0.080 2.2   5640 13.9 0.036099
## 28 15.2  0 11.2  8.2  7.6 0.571 101.8  10  7.9 0.103 2.8   5370 21.5 0.038201
## 29 11.9  0 10.7 16.6 15.7 0.521  93.8 168  8.9 0.092 3.6   6370 15.4 0.023400
## 30 16.6  1  8.9  5.8  5.4 0.521  97.3  46 25.4 0.072 2.6   3960 23.7 0.075298
## 31 14.0  0  9.3  5.5  5.4 0.535 104.5   6  2.0 0.135 4.0   4530 20.0 0.041999
## 32 12.5  0 10.9  9.0  8.1 0.586  96.4  97  8.2 0.105 4.3   6170 16.3 0.042698
```

```
## 33 14.7  1 10.4  6.3  6.4 0.560  97.2  23  9.5 0.076 2.4   4620 23.3 0.049499
## 34 12.6  0 11.8  9.7  9.7 0.542  99.0  18  2.1 0.102 3.5   5890 16.6 0.040799
## 35 12.3  0 10.2  9.7  8.7 0.526  94.8 113  7.6 0.124 5.0   5720 15.8 0.020700
## 36 15.0  0 10.0 10.9  9.8 0.531  96.4   9  2.4 0.087 3.8   5590 15.3 0.006900
## 37 17.7  1  8.7  5.8  5.6 0.638  97.4  24 34.9 0.076 2.8   3820 25.4 0.045198
## 38 13.3  0 10.4  5.1  4.7 0.599 102.4   7  4.0 0.099 2.7   4250 22.5 0.053998
## 39 14.9  1  8.8  6.1  5.4 0.515  95.3  36 16.5 0.086 3.5   3950 25.1 0.047099
## 40 14.5  1 10.4  8.2  7.4 0.560  98.1  96 12.6 0.088 3.1   4880 22.8 0.038801
## 41 14.8  0 12.2  7.2  6.6 0.601  99.8   9  1.9 0.084 2.0   5900 14.4 0.025100
## 42 14.1  0 10.9  5.6  5.4 0.523  96.8   4  0.2 0.107 3.7   4890 17.0 0.088904
## 43 16.2  1  9.9  7.5  7.0 0.522  99.6  40 20.8 0.073 2.7   4960 22.4 0.054902
## 44 13.6  0 12.1  9.5  9.6 0.574 101.2  29  3.6 0.111 3.7   6220 16.2 0.028100
## 45 13.9  1  8.8  4.6  4.1 0.480  96.8  19  4.9 0.135 5.3   4570 24.9 0.056202
## 46 12.6  0 10.4 10.6  9.7 0.599  98.9  40  2.4 0.078 2.5   5930 17.1 0.046598
## 47 13.0  0 12.1  9.0  9.1 0.623 104.9   3  2.2 0.113 4.0   5880 16.0 0.052802
##       Time Crime
## 1  26.2011   791
## 2  25.2999  1635
## 3  24.3006   578
## 4  29.9012  1969
## 5  21.2998  1234
## 6  20.9995   682
## 7  20.6993   963
## 8  24.5988  1555
## 9  29.4001   856
## 10 19.5994   705
## 11 41.6000  1674
## 12 34.2984   849
## 13 36.2993   511
## 14 21.5010   664
## 15 22.7008   798
## 16 26.0991   946
## 17 19.1002   539
## 18 18.1996   929
## 19 24.9008   750
## 20 26.4010  1225
## 21 37.5998   742
## 22 37.0994   439
## 23 25.1989  1216
## 24 17.6000   968
## 25 21.9003   523
## 26 22.1005  1993
## 27 28.4999   342
## 28 25.8006  1216
## 29 36.7009  1043
## 30 28.3011   696
## 31 21.7998   373
## 32 30.9014   754
## 33 25.5005  1072
## 34 21.6997   923
## 35 37.4011   653
## 36 44.0004  1272
## 37 31.6995   831
## 38 16.6999   566
```

```
## 39 27.3004   826
## 40 29.3004  1151
## 41 30.0001   880
## 42 12.1996   542
## 43 31.9989   823
## 44 30.0001  1030
## 45 32.5996   455
## 46 16.6999   508
## 47 16.0997   849
```

Scaling the data

```
crime_data_scaled = as.data.frame(scale(crime_data[,c(1,3,4,5,6,7,8,9,10,11,12,13,14,15)]))
 # Add column 2 and response variable back in
crime_data_scaled <- cbind(crime_data[,2],crime_data_scaled,crime_data[,16])
colnames(crime_data_scaled)[1] <- "So"
colnames(crime_data_scaled)[16] <- "Crime"
```

Stepwise Regression

```
set.seed(1)
# Set up repeated 5-fold cross-validation
train.control <- trainControl(method = "cv", number = 5, p=0.7)
# Train the model
step.model <- train(
                Crime ~., data = crime_data_scaled,
                method = "lmStepAIC",
                trControl = train.control,
              )
```

```
## Start:  AIC=418.66
## .outcome ~ So + M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob + Time
##
##          Df Sum of Sq     RSS    AIC
## - So      1      1417  998651 416.71
## - Wealth  1      1472  998706 416.71
## - Po2     1      8395 1005629 416.97
## - Pop     1     10678 1007912 417.06
## - LF      1     27061 1024295 417.67
## - Time    1     43750 1040984 418.29
## <none>                 997234 418.66
## - U1      1     55541 1052775 418.72
## - Po1     1     62591 1059825 418.97
## - NW      1     64969 1062203 419.05
## - M       1     68316 1065550 419.17
## - Prob    1     89802 1087036 419.93
## - M.F     1    113567 1110801 420.75
## - U2      1    116645 1113879 420.86
## - Ineq    1    217901 1215135 424.17
```

```
## Step:  AIC=503.93
## .outcome ~ 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
```

```r
summary(step.model)
```

```
##
## Call:
## lm.default(formula = .outcome ~ M + Ed + Po1 + M.F + U1 + U2 +
##     Ineq + Prob, data = dat)
##
## 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)   905.09      28.52  31.731  < 2e-16 ***
## M             117.28      42.10   2.786  0.00828 **
## Ed            201.50      59.02   3.414  0.00153 **
## Po1           305.07      46.14   6.613 8.26e-08 ***
## M.F            65.83      40.08   1.642  0.10874
## U1           -109.73      60.20  -1.823  0.07622 .
## U2            158.22      61.22   2.585  0.01371 *
## Ineq          244.70      55.69   4.394 8.63e-05 ***
## Prob          -86.31      33.89  -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
```

Observations: Applying Stepwise Regression and Cross Validation, it down-selected 8 variables and obtained an Adjusted R-Squared value of 0.7444. Now that the p-values of M.F and U1 are not so small (both are greater than 0.05), let's see if we can down-select more variables.

```r
#Fitting a new model with the 6 variables, excluding M.F and U1
set.seed(1)
step.model.simpler = lm(Crime ~ Prob+U2+M+Ed+Ineq+Po1, data = crime_data_scaled)
summary(step.model.simpler)
```

```
##
```

```
## Call:
## lm.default(formula = Crime ~ Prob + U2 + M + Ed + Ineq + Po1,
##     data = crime_data_scaled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      29.27  30.918  < 2e-16 ***
## Prob          -86.44      34.74  -2.488  0.01711 *
## U2             75.47      34.55   2.185  0.03483 *
## M             131.98      41.85   3.154  0.00305 **
## Ed            219.79      50.07   4.390 8.07e-05 ***
## Ineq          269.91      55.60   4.855 1.88e-05 ***
## Po1           341.84      40.87   8.363 2.56e-10 ***
## ---
## 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
```

Observations: These two models (one with 8 variables and the other with 6 variables) produce a very similar adjusted R-squared. In order to verify the model performance better, I need to run a cross validation for these models, respectively.

Cross Validation for model comparison between the one with 8 variables and the other with 6 variables

```r
set.seed(1)
# R-squared (Cross Validation) for the model with 8 variables
TSS <- sum((crime_data_scaled$Crime - mean(crime_data_scaled$Crime))^2)
RSS <- 0
for(i in 1:nrow(crime_data_scaled)) {
  mod_Step_i = lm(Crime ~ M.F+U1+Prob+U2+M+Ed+Ineq+Po1, data = crime_data_scaled[-i,])
  pred_i <- predict(mod_Step_i,newdata=crime_data_scaled[i,])
  RSS <- RSS + ((pred_i - crime_data_scaled[i,16])^2)
}
cv_R2 <- 1 - RSS/TSS
cv_R2
```

```
##        1
## 0.667621
```

```r
set.seed(1)
# R-squared (Cross Validation) for the model with 6 variables
TSS <- sum((crime_data_scaled$Crime - mean(crime_data_scaled$Crime))^2)
RSS <- 0
for(i in 1:nrow(crime_data_scaled)) {
  mod_Step_i = lm(Crime ~ Prob+U2+M+Ed+Ineq+Po1, data = crime_data_scaled[-i,])
  pred_i <- predict(mod_Step_i,newdata=crime_data_scaled[i,])
```

```r
  RSS <- RSS + ((pred_i - crime_data_scaled[i,16])^2)
}
cv_R2 <- 1 - RSS/TSS
cv_R2
```

```
##         1
## 0.6661638
```

Observations: Even if the second model has less number of variables, the R-squared by the cross validation produces a very similar to the one with two more variables. This indicates that the variables M.F and U1 are not significant.

LASSO

```r
set.seed(1)
# alpha = 1 for LASSO
lasso_model <- cv.glmnet(x=as.matrix(crime_data_scaled[,-16]),y=as.matrix(crime_data_scaled$Crime),alpha
coef(lasso_model, s=lasso_model$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept) 889.648600
## So           45.344736
## M            77.635868
## Ed           98.182711
## Po1         311.196426
## Po2           .
## LF            2.887732
## M.F          46.955007
## Pop           .
## NW            2.654767
## U1            .
## U2           29.302470
## Wealth        .
## Ineq        164.140537
## Prob        -77.051224
## Time          .
```

```r
set.seed(1)
# Fitting a new model with 11 variables
lasso_model = lm(Crime ~So+M+Ed+Po1+LF+M.F+NW+U1+U2+Ineq+Prob, data = crime_data_scaled)
summary(lasso_model)
```

```
##
## Call:
## lm.default(formula = Crime ~ So + M + Ed + Po1 + LF + M.F + NW +
##     U1 + U2 + Ineq + Prob, data = crime_data_scaled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -443.2 -101.4    4.1  120.5  486.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   892.63      55.99  15.943  < 2e-16 ***
## So             36.57     139.62   0.262  0.79489
## M             106.61      49.29   2.163  0.03747 *
## Ed            209.15      65.00   3.218  0.00278 **
## Po1           295.60      54.50   5.424 4.44e-06 ***
## LF            -10.69      54.11  -0.198  0.84447
## M.F            74.96      51.13   1.466  0.15159
## NW             13.01      59.46   0.219  0.82814
## U1           -109.08      71.71  -1.521  0.13725
## U2            151.47      65.99   2.295  0.02783 *
## Ineq          233.00      67.67   3.443  0.00151 **
## Prob          -96.00      39.58  -2.425  0.02059 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202.9 on 35 degrees of freedom
## Multiple R-squared:  0.7906, Adjusted R-squared:  0.7248
## F-statistic: 12.01 on 11 and 35 DF,  p-value: 6.965e-09
```

Observation: I obtained Adjusted R-SQuared of ~0.725 for the model with the 11 variables using LASSO and Cross Validation. But there are several variables don't appear to be significant. I would like to rebuild a model by removing these variables.

```r
set.seed(1)
# Fitting a new model with the selected 6 variables.
lasso_model = lm(Crime ~M+Ed+Po1+U2+Ineq+Prob, data = crime_data_scaled)
summary(lasso_model)
```

```
##
## Call:
## lm.default(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob,
##     data = crime_data_scaled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      29.27  30.918  < 2e-16 ***
## M             131.98      41.85   3.154  0.00305 **
## Ed            219.79      50.07   4.390 8.07e-05 ***
## Po1           341.84      40.87   8.363 2.56e-10 ***
## U2             75.47      34.55   2.185  0.03483 *
## Ineq          269.91      55.60   4.855 1.88e-05 ***
## Prob          -86.44      34.74  -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
```

Observation: The model with less variables by removing not-significant 5 variables gives a slightly higher Adjusted R-squared value, which is 0.7307. Now it's time to run a leave-one-out cross-validation to check how good this model is actually.

```r
set.seed(1)
# R-squared (Cross Validation) for the model with 6 variables
TSS <- sum((crime_data_scaled$Crime - mean(crime_data_scaled$Crime))^2)
RSS <- 0
for(i in 1:nrow(crime_data_scaled)) {
  mod_Step_i = lm(Crime ~ M+Ed+Po1+U2+Ineq+Prob, data = crime_data_scaled[-i,])
  pred_i <- predict(mod_Step_i,newdata=crime_data_scaled[i,])
  RSS <- RSS + ((pred_i - crime_data_scaled[i,16])^2)
}
cv_R2 <- 1 - RSS/TSS
cv_R2
```

Cross Validation

```
##         1
## 0.6661638
```

Observations: Somehow, the selected set of variables is the same as the simpler model derived by Stepwise Regression as above; therefore, the R-squared using a leave-one-out cross-validation is the same.

Elastic Net

```r
set.seed(1)
# By varying alpha value between 0 and 1, we can
r_sqrd <- c()
alphas <- c()
for (i in 0:10) {
  alpha <- i / 10
  temp_EN_model <- cv.glmnet(x=as.matrix(crime_data_scaled[,-16]),y=as.matrix(crime_data_scaled$Crime),a

  r_sqrd <- c(r_sqrd, temp_EN_model$glmnet.fit$dev.ratio[which(temp_EN_model$glmnet.fit$lambda == temp_l
  alphas <- c(alphas, alpha)
}

alpha_r2 <- cbind(alphas, r_sqrd)
alpha_r2
```

```
##       alphas    r_sqrd
##  [1,]    0.0 0.7493364
##  [2,]    0.1 0.7535141
```

```
##  [3,]    0.2 0.7386241
##  [4,]    0.3 0.7254203
##  [5,]    0.4 0.7723366
##  [6,]    0.5 0.7921574
##  [7,]    0.6 0.7942149
##  [8,]    0.7 0.7730448
##  [9,]    0.8 0.7776139
## [10,]    0.9 0.7935339
## [11,]    1.0 0.6982766
```

```r
best_alpha <- (which.max(r_sqrd) - 1) / 10
best_alpha
```

```
## [1] 0.6
```

```r
EN_model <- cv.glmnet(
  x=as.matrix(crime_data_scaled[,-16]),
  y=as.matrix(crime_data_scaled$Crime),
  alpha=best_alpha,
  nfolds = 5,
  type.measure="mse",
  family="gaussian"
)

#Output the coefficients of the variables selected by Elastic Net
coef(EN_model, s=EN_model$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept) 894.344910
## So           31.549328
## M           106.159345
## Ed          179.973055
## Po1         291.061285
## Po2           .
## LF            .
## M.F          53.165375
## Pop         -22.378005
## NW           18.449918
## U1          -78.619726
## U2          124.856961
## Wealth       63.742853
## Ineq        256.573258
## Prob        -91.802322
## Time         -1.002722
```

```r
EN_model <- lm(Crime ~ So+M+Ed+Po1+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob+Time, data = crime_data_scaled[-i,]
summary(EN_model)
```

```
##
## Call:
## lm.default(formula = Crime ~ So + M + Ed + Po1 + M.F + Pop +
```

```
##     NW + U1 + U2 + Wealth + Ineq + Prob + Time, data = crime_data_scaled[-i,
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -439.61 -113.98   15.62  116.99  479.87
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  898.717     53.820  16.699  < 2e-16 ***
## So            22.346    130.796   0.171 0.865422
## M            114.239     51.713   2.209 0.034445 *
## Ed           194.909     65.189   2.990 0.005330 **
## Po1          290.075     68.464   4.237 0.000179 ***
## M.F           49.042     50.866   0.964 0.342208
## Pop          -29.890     48.717  -0.614 0.543857
## NW            21.616     61.060   0.354 0.725646
## U1           -89.334     68.575  -1.303 0.201969
## U2           139.306     70.155   1.986 0.055692 .
## Wealth        82.667     99.313   0.832 0.411358
## Ineq         283.436     88.278   3.211 0.003011 **
## Prob         -98.120     49.740  -1.973 0.057228 .
## Time          -8.706     47.428  -0.184 0.855506
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 208.5 on 32 degrees of freedom
## Multiple R-squared:  0.7965, Adjusted R-squared:  0.7139
## F-statistic: 9.637 on 13 and 32 DF,  p-value: 1.063e-07
```

Observations: There are 6 variables appear to be significant. While the adjusted R-squared of 0.714 doesn't look bad, this suggests that we might want to rebuild a model using only these 6 variables.

```
# Rebuild a model using 6 variables
EN_model <- lm(Crime ~ M+Ed+Po1+U2+Ineq+Prob, data = crime_data_scaled[-i,])
summary(EN_model)
```

```
##
## Call:
## lm.default(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob,
##     data = crime_data_scaled[-i, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -472.64  -75.59   -5.77  135.33  557.55
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   907.00      29.92  30.313  < 2e-16 ***
## M             132.80      42.33   3.137  0.00324 **
## Ed            222.74      51.06   4.362 9.14e-05 ***
## Po1           339.71      41.60   8.166 5.70e-10 ***
## U2             74.38      35.00   2.125  0.03996 *
```

```
## Ineq          269.46      56.19   4.796 2.38e-05 ***
## Prob          -86.67      35.11  -2.469  0.01805 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202.8 on 39 degrees of freedom
## Multiple R-squared:  0.7655, Adjusted R-squared:  0.7295
## F-statistic: 21.22 on 6 and 39 DF,  p-value: 6.919e-11
```

Observations: After dropping non-significant variables, the new model's adjusted R-squared improved a bit. Now, it's time to check how good the model is using a leave-one-out cross-validation.

```r
set.seed(1)
# R-squared (Cross Validation) for the model with 6 variables
TSS <- sum((crime_data_scaled$Crime - mean(crime_data_scaled$Crime))^2)
RSS <- 0
for(i in 1:nrow(crime_data_scaled)) {
  mod_Step_i = lm(Crime ~ M+Ed+Po1+U2+Ineq+Prob, data = crime_data_scaled[-i,])
  pred_i <- predict(mod_Step_i,newdata=crime_data_scaled[i,])
  RSS <- RSS + ((pred_i - crime_data_scaled[i,16])^2)
}
cv_R2 <- 1 - RSS/TSS
cv_R2
```

Cross Validation

```
##         1
## 0.6661638
```

Observations: After running Stepwise Regression, LASSO, and Elastic Net for the variable selection, there is one very interesting outcome. Even though the initial outcomes (=Adjusted R-squared values) after running each algorithm for the variable selection are varying throughout algorithms, the final outcomes after dropping non-significant variables are all the same for all algorithms.

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.

Answer: In my current company, we conduct research and development of lithium metal batteries. When designing a new battery, we need to test its performance using various metrics such as rate capability, cycle life, and degradation modes. These metrics are essential for comparing different battery designs. Given the high cost of producing batteries and the extensive time required for cycling (charging and discharging a battery to test its performance), we must optimize our testing process.

A design of experiments (DOE) approach would be appropriate here. By systematically varying key factors such as electrode materials, electrolyte compositions, and manufacturing conditions, we can efficiently determine the optimal combination of variables that enhance battery performance. DOE allows us to minimize the number of experiments needed while still gaining comprehensive insights into how different design parameters influence the battery's outcomes. This approach not only saves time and resources but also ensures that we obtain reliable and statistically significant results.

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.

```
set.seed(1)
ff_design<-FrF2(nruns = 16,nfactors = 10, default.levels = c(-1, 1))
ff_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 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 number of goals scored by Messi in a penalty shootout. Messi has his own penalty conversion rate. So then, each penalty kick can be considered a Bernoulli trial and the binomial distribution gives the number of goals (successes) in a fixed number of trials.

b. Geometric

- The number of YouTube videos that I click until I finally find one that both my wife and myself like.

c. Poisson

- The number of chocolate chips on my chocolate chip cookies.

d. Exponential

- Recently, I got a lot of scam calls. I am curious the time until my next scam call.

e. Weibull

- The time until a battery hits 80% capacity retension. (Batteries degrade over time due to usage and environmental factors. To understand the lifetime of the battery, using Weibull distribution is common)