HW5 Notebook

ISYE6501x - Introduction to Analytics Modeling

Due: June 21, 2017

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=38),tidy=TRUE,fig.width=5, fig.height=4)
```

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, and 3) elastic net. For Parts 2 and 3, remember to scale the data first. For Parts 2 and 3, use the glmnet function in R.

Part 1) Stepwise regression:

```
rm(list = ls())
setwd("~/Desktop/Edx/Intro to Analytics Modeling/Week 5/WK5 Homework")
uscrime = read.table("11.1uscrimeSummer2018.txt",
    stringsAsFactors = FALSE, header = TRUE)
# head(uscrime)

# Scaling data except So (binary) and
# Crime (response)
uscrime3 = as.data.frame(scale(uscrime))
uscrime3$So = uscrime$So
uscrime3$Crime = uscrime$Crime
# head(uscrime3)

# Backwards stepwise regression
model_backward = lm(Crime ~ ., data = uscrime3)
step(model_backward, 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
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
##     data = uscrime3)
##
## Coefficients:
## (Intercept)            M           Ed          Po1          M.F
##      905.09       117.28       201.50       305.07        65.83
##          U1           U2         Ineq         Prob
##     -109.73       158.22       244.70       -86.31
```

```r
# Forwards stepwise regression
model_forward = lm(Crime ~ 1, data = uscrime3)
step(model_forward, scope = formula(lm(Crime ~
    ., data = uscrime3)), direction = "forward")
```

```
## Start:  AIC=561.02
## Crime ~ 1
##
##          Df Sum of Sq     RSS    AIC
## + Po1     1   3253302 3627626 532.94
## + Po2     1   3058626 3822302 535.39
## + Wealth  1   1340152 5540775 552.84
## + Prob    1   1257075 5623853 553.54
## + Pop     1    783660 6097267 557.34
## + Ed      1    717146 6163781 557.85
## + M.F     1    314867 6566061 560.82
## <none>                6880928 561.02
## + LF      1    245446 6635482 561.32
## + Ineq    1    220530 6660397 561.49
## + U2      1    216354 6664573 561.52
```

```
## + Time    1    154545 6726383 561.96
## + So      1     56527 6824400 562.64
## + M       1     55084 6825844 562.65
## + U1      1     17533 6863395 562.90
## + NW      1      7312 6873615 562.97
##
## Step:  AIC=532.94
## Crime ~ Po1
##
##          Df Sum of Sq     RSS    AIC
## + Ineq    1    739819 2887807 524.22
## + M       1    616741 3010885 526.18
## + M.F     1    250522 3377104 531.57
## + NW      1    232434 3395192 531.82
## + So      1    219098 3408528 532.01
## + Wealth  1    180872 3446754 532.53
## <none>                3627626 532.94
## + Po2     1    146167 3481459 533.00
## + Prob    1     92278 3535348 533.72
## + LF      1     77479 3550147 533.92
## + Time    1     43185 3584441 534.37
## + U2      1     17848 3609778 534.70
## + Pop     1      5666 3621959 534.86
## + U1      1      2878 3624748 534.90
## + Ed      1       767 3626859 534.93
##
## Step:  AIC=524.22
## Crime ~ Po1 + Ineq
##
##          Df Sum of Sq     RSS    AIC
## + Ed      1    587050 2300757 515.53
## + M.F     1    454545 2433262 518.17
## + Prob    1    280690 2607117 521.41
## + LF      1    260571 2627236 521.77
## + Wealth  1    213937 2673871 522.60
## + M       1    181236 2706571 523.17
## + Pop     1    130377 2757430 524.04
## <none>                2887807 524.22
## + NW      1     36439 2851369 525.62
## + So      1     33738 2854069 525.66
## + Po2     1     30673 2857134 525.71
## + U1      1      2309 2885498 526.18
## + Time    1       497 2887310 526.21
## + U2      1       253 2887554 526.21
##
## Step:  AIC=515.53
## Crime ~ Po1 + Ineq + Ed
##
##          Df Sum of Sq     RSS    AIC
## + M       1    239405 2061353 512.37
## + Prob    1    234981 2065776 512.47
## + M.F     1    117026 2183731 515.08
## <none>                2300757 515.53
## + Wealth  1     79540 2221218 515.88
```

```
## + U2      1     62112 2238646 516.25
## + Time    1     61770 2238987 516.26
## + Po2     1     42584 2258174 516.66
## + Pop     1     39319 2261438 516.72
## + U1      1      7365 2293392 517.38
## + LF      1      7254 2293503 517.39
## + NW      1      4210 2296547 517.45
## + So      1      4135 2296622 517.45
##
## Step:  AIC=512.37
## Crime ~ Po1 + Ineq + Ed + M
##
##          Df Sum of Sq     RSS    AIC
## + Prob    1    258063 1803290 508.08
## + U2      1    200988 1860365 509.55
## + Wealth  1    163378 1897975 510.49
## <none>                2061353 512.37
## + M.F     1     74398 1986955 512.64
## + U1      1     50835 2010518 513.20
## + Po2     1     45392 2015961 513.32
## + Time    1     42746 2018607 513.39
## + NW      1     16488 2044865 513.99
## + Pop     1      8101 2053251 514.19
## + So      1      3189 2058164 514.30
## + LF      1      2988 2058365 514.30
##
## Step:  AIC=508.08
## Crime ~ Po1 + Ineq + Ed + M + Prob
##
##          Df Sum of Sq     RSS    AIC
## + U2      1    192233 1611057 504.79
## + Wealth  1     86490 1716801 507.77
## + M.F     1     84509 1718781 507.83
## <none>                1803290 508.08
## + U1      1     52313 1750977 508.70
## + Pop     1     47719 1755571 508.82
## + Po2     1     37967 1765323 509.08
## + So      1     21971 1781320 509.51
## + Time    1     10194 1793096 509.82
## + LF      1       990 1802301 510.06
## + NW      1       797 1802493 510.06
##
## Step:  AIC=504.79
## Crime ~ Po1 + Ineq + Ed + M + Prob + U2
##
##          Df Sum of Sq     RSS    AIC
## <none>                1611057 504.79
## + Wealth  1     59910 1551147 505.00
## + U1      1     54830 1556227 505.16
## + Pop     1     51320 1559737 505.26
## + M.F     1     30945 1580112 505.87
## + Po2     1     25017 1586040 506.05
## + So      1     17958 1593098 506.26
## + LF      1     13179 1597878 506.40
```

```
## + Time     1      7159 1603898 506.58
## + NW       1       359 1610698 506.78
```

```
##
## Call:
## lm(formula = Crime ~ Po1 + Ineq + Ed + M + Prob + U2, data = uscrime3)
##
## Coefficients:
## (Intercept)          Po1         Ineq           Ed            M
##      905.09       341.84       269.91       219.79       131.98
##        Prob           U2
##      -86.44        75.47
```

```r
# Both stepwise regression
model_both = lm(Crime ~ ., data = uscrime3)
step(model_both, scope = list(lower = formula(lm(Crime ~
    1, data = uscrime3)), upper = formula(lm(Crime ~
    ., data = uscrime3))), 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
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
##     data = uscrime3)
##
## Coefficients:
## (Intercept)            M           Ed          Po1          M.F
##      905.09       117.28       201.50       305.07        65.83
##          U1           U2         Ineq         Prob
##     -109.73       158.22       244.70       -86.31
```

Analysis: For the backwards regression, we start with 15 factors and the function keeps eliminating factors as the AIC improves (lowers) and ultimately ends up with 8 final factors (M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob). Conversely, using forwards regression, the final model ended up with only 6 factors (Po1 + Ineq + Ed + M + Prob + U2). Using the both-ways regression, the final model was the same as using the backwards methodology (M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob).

Final stepwise regression model, quality of fit and coefficients:

```r
model_final = lm(formula = Crime ~ M +
    Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
    data = uscrime3)
summary(model_final)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
##     data = uscrime3)
##
## 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
```

```r
# asses quality of fit using
# cross-validation
# install.packages('DAAG')
library(DAAG)
```

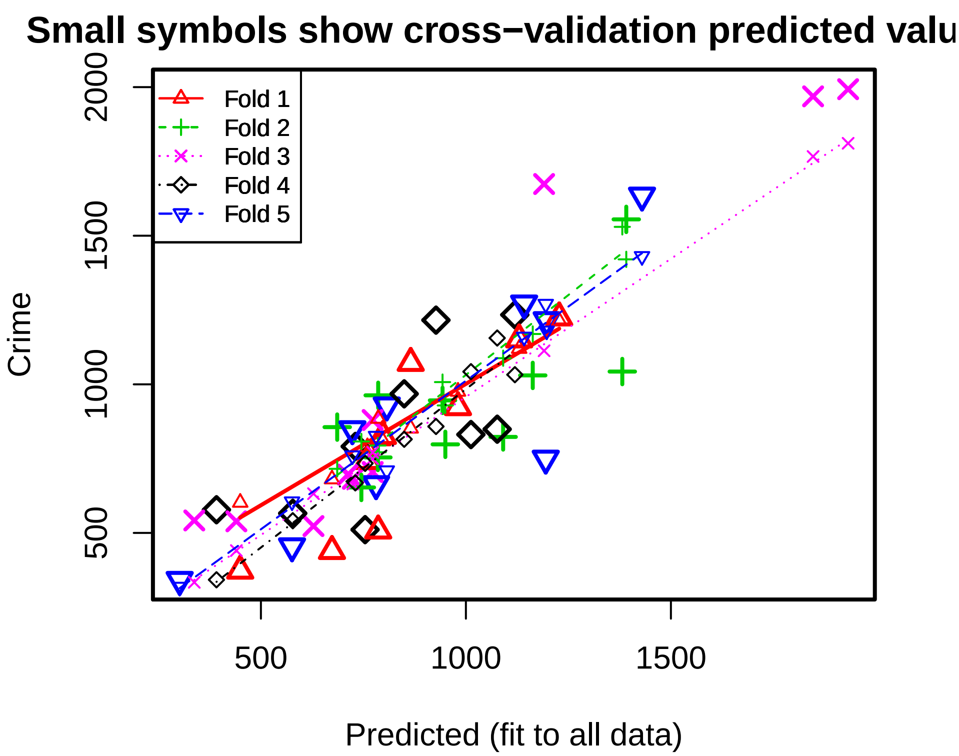
```
## Loading required package: lattice
```

```r
model_final_cv = cv.lm(uscrime3, model_final,
    m = 5, seed = 42)
```

```
## 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
```

```
## Warning in cv.lm(uscrime3, model_final, m = 5, seed = 42):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 9
##                 20    21   22   31   33    34  39    40   46
## Predicted   1227.6 759.8  673  450  865 980.7 798 1129.9  786
## cvpred      1213.5 788.9  681  602  852 975.8 809 1120.1  885
## Crime       1225.0 742.0  439  373 1072 923.0 826 1151.0  508
## CV residual   11.5 -46.9 -242 -229  220 -52.8  17   30.9 -377
##
## Sum of squares = 307812    Mean square = 34201    n = 9
##
## fold 2
## Observations in test set: 10
##                7    8   9   15     16   29    32  35   43   44
## Predicted   786 1391 686  950  943.0 1381 785.3  745 1091 1163
```

```
## cvpred      771 1421 716  929 1007.5 1530 794.1  808 1088 1169
## Crime       963 1555 856  798  946.0 1043 754.0  653  823 1030
## CV residual 192  134 140 -131  -61.5 -487 -40.1 -155 -265 -139
##
## Sum of squares = 447554    Mean square = 44755    n = 10
##
## fold 3
## Observations in test set: 10
##                4     6    10    11    17   25   26     30  41  42
## Predicted   1847 724.3 772.7 1191 440.2  628 1932 711.82 772 338
## cvpred      1767 664.4 771.1 1113 440.5  632 1811 697.22 750 334
## Crime       1969 682.0 705.0 1674 539.0  523 1993 696.00 880 542
## CV residual  202  17.6 -66.1  561  98.5 -109  182  -1.22 130 208
##
## Sum of squares = 475521    Mean square = 47552    n = 10
##
## fold 4
## Observations in test set: 9
##               1   3    5   13   23  24   37  38   47
## Predicted   730 392 1119  754  927 850 1012 578 1076
## cvpred      668 342 1032  734  858 815 1043 541 1156
## Crime       791 578 1234  511 1216 968  831 566  849
## CV residual 123 236  202 -223  358 153 -212  25 -307
##
## Sum of squares = 451853    Mean square = 50206    n = 9
##
## fold 5
## Observations in test set: 9
##                2    12   14  18   19    27     28   36   45
## Predicted   1430 723.1  781 807 1195 301.9 1197.0 1142  576
## cvpred      1431 760.4  826 710 1270 318.1 1180.3 1160  605
## Crime       1635 849.0  664 929  750 342.0 1216.0 1272  455
## CV residual  204  88.6 -162 219 -520  23.9   35.7  112 -150
##
## Sum of squares = 431372    Mean square = 47930    n = 9
##
## Overall (Sum over all 9 folds)
##    ms
## 44981
```

```r
# CV R^2 = 1 - SSR/SST SST = sum of
# squared totals SST = sum of (Crime
# value - mean(uscrime3$Crime))^2 over
# all data points
mean_crime = mean(uscrime3$Crime)
# mean_crime
sq_tot = (uscrime3$Crime - mean_crime)^2
# sq_tot
SST = sum(sq_tot)
# SST Extract MSE below
MSE = attr(model_final_cv, "ms")
# MSE SSR = sum of squared residuals
# for regular R^2 SSR = MSE*N for CV.
# R^2
```

```r
SSR = MSE * 47
# SSR
cv_r2 = 1 - (SSR/SST)
cv_r2
```

```
## [1] 0.693
```

```r
# R^2 adj = 1 - (1-R^2)(N-1)/(N-K-1) K
# = number of predictors
cv_r2_adj = 1 - (1 - cv_r2) * (47 - 1)/(47 -
    8 - 1)
cv_r2_adj
```

```
## [1] 0.628
```

```r
# Coefficients
model_final$coefficients
```

```
## (Intercept)           M          Ed         Po1         M.F          U1
##       905.1       117.3       201.5       305.1        65.8      -109.7
##          U2        Ineq        Prob
##       158.2       244.7       -86.3
```

Analysis: as you can see from the summary output of the final chosen model above (M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob), the p-values would indicate that we could make a judgement call to elminate M.F. However, in this case, as the p-value is very close to being significant at the .10 level, we will keep it in. The final calculated R^2 is 69.3%, and Adjusted R^2 is 62.8%, indicating a fairly good quality of fit. The coefficients of the model are seen in the final output line above.

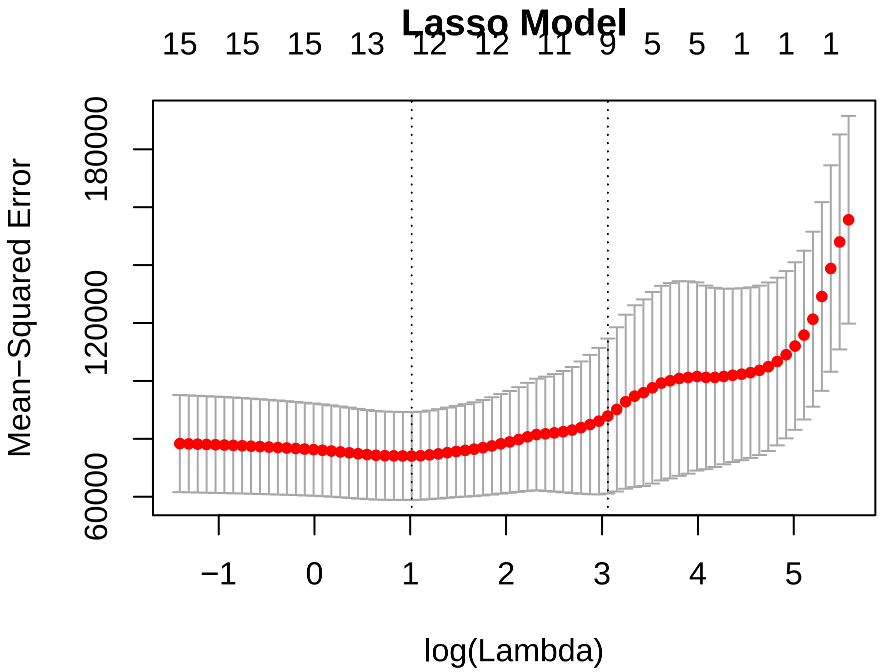
Part 2) Lasso:

```r
#install.packages("glmnet")
library(glmnet)
```

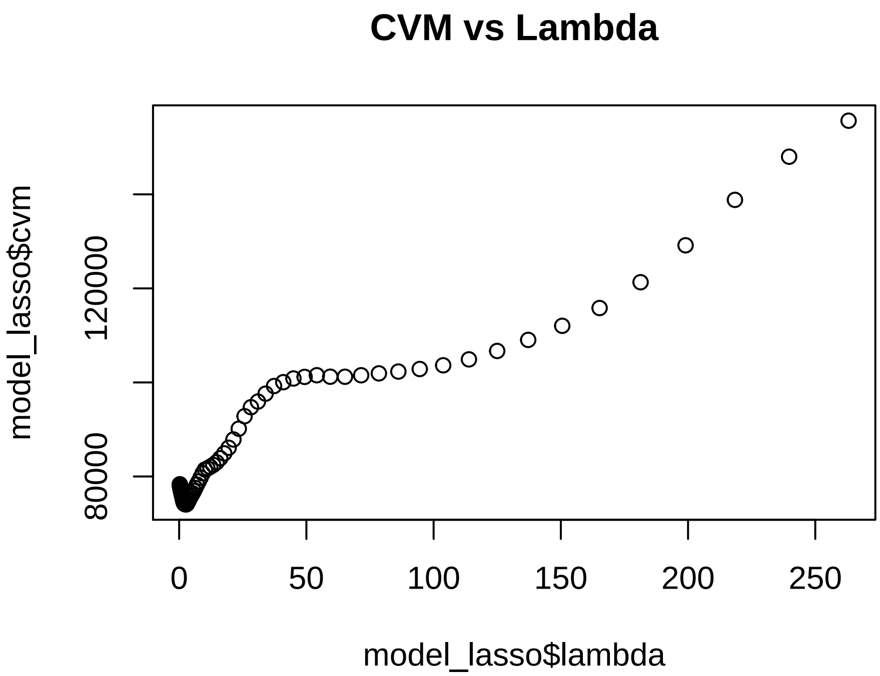
```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16
```

```r
#use scaled data uscrime3

#the cv.glmnet function does use some randomization
set.seed(12)
model_lasso = cv.glmnet(x=as.matrix(uscrime3[,-16]),
                        y=as.matrix(uscrime3[,16]),
                        alpha=1,
                        nfolds=5,
```

```r
                          type.measure = "mse", #helps us automatically pick budget "T" lambda
                          family = "gaussian")

#model_lasso
plot(model_lasso, main = "Lasso Model")
```




```r
#cbind(model_lasso$lambda, model_lasso$cvm)
plot(model_lasso$lambda, model_lasso$cvm, main = "CVM vs Lambda")
```



Analysis: the output and plots indicate that the optimal lambda (lambda.min) is 2.76. We can also see that the higher the lambda, the higher the MSE.

Final Lasso model, quality of fit and coefficients:

```
# Model using all variables selected by
# Lasso
model_lasso_2 = lm(formula = Crime ~ M +
    So + Ed + Po1 + M.F + Pop + NW + U1 +
    U2 + Wealth + Ineq + Prob, data = uscrime3)
summary(model_lasso_2)
```

```
##
## Call:
## lm(formula = Crime ~ M + So + Ed + Po1 + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob, data = uscrime3)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -434.2 -107.0   18.6  115.9  470.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    897.3       51.9   17.29  < 2e-16 ***
## M              112.7       49.4    2.28   0.0288 *
## So              22.9      125.3    0.18   0.8562
## Ed             195.7       62.9    3.11   0.0038 **
## Po1            293.2       65.0    4.51  7.3e-05 ***
```

```
## M.F             48.9       48.1    1.02   0.3166
## Pop            -33.3       45.6   -0.73   0.4711
## NW              19.2       57.7    0.33   0.7419
## U1             -89.8       65.7   -1.37   0.1807
## U2             140.8       66.8    2.11   0.0424 *
## Wealth          83.3       95.5    0.87   0.3893
## Ineq           285.8       85.2    3.35   0.0020 **
## Prob           -92.8       41.1   -2.26   0.0307 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 203 on 34 degrees of freedom
## Multiple R-squared:  0.797,  Adjusted R-squared:  0.726
## F-statistic: 11.1 on 12 and 34 DF,  p-value: 1.52e-08
```

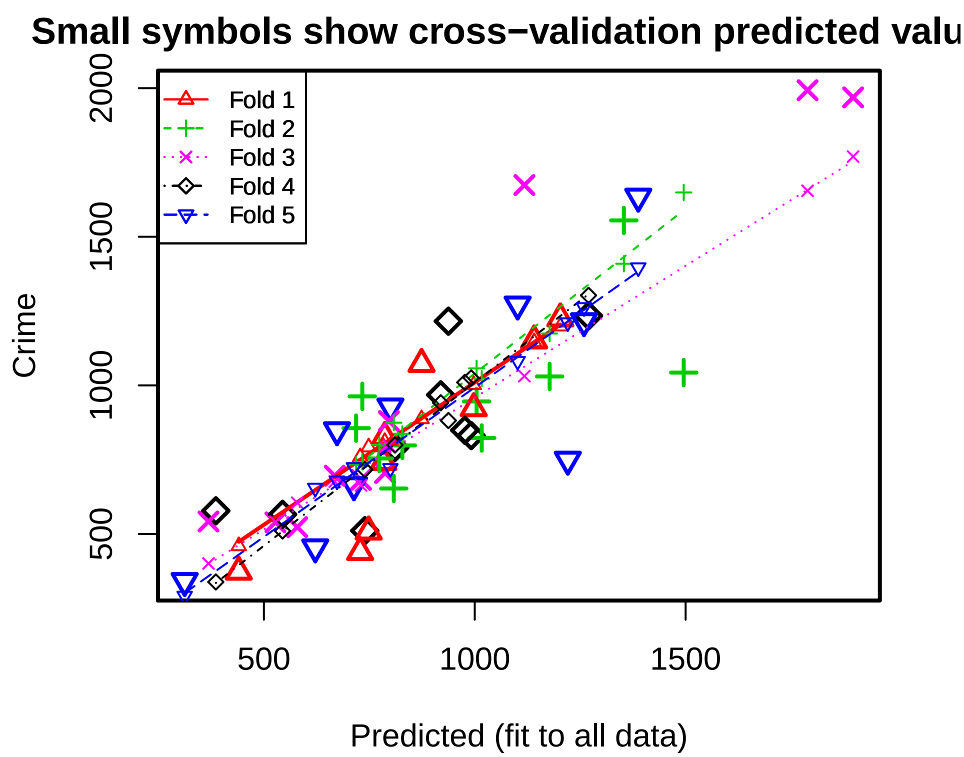
```r
# Eliminate low p-value variables: So,
# M.F., Pop, NW, U1, and Wealth

# Model after eliminating variables
# with low p-values
model_lasso_final = lm(formula = Crime ~
    M + Ed + Po1 + U2 + Ineq + Prob, data = uscrime3)
summary(model_lasso_final)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = uscrime3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -470.7  -78.4  -19.7  133.1  556.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    905.1       29.3   30.92  < 2e-16 ***
## M              132.0       41.8    3.15   0.0031 **
## Ed             219.8       50.1    4.39  8.1e-05 ***
## Po1            341.8       40.9    8.36  2.6e-10 ***
## U2              75.5       34.5    2.18   0.0348 *
## Ineq           269.9       55.6    4.85  1.9e-05 ***
## Prob           -86.4       34.7   -2.49   0.0171 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201 on 40 degrees of freedom
## Multiple R-squared:  0.766,  Adjusted R-squared:  0.731
## F-statistic: 21.8 on 6 and 40 DF,  p-value: 3.42e-11
```

```r
# asses quality of fit using
# cross-validation
# install.packages('DAAG')
library(DAAG)
model_lasso_final_cv = cv.lm(uscrime3,
    model_lasso_final, m = 5, seed = 42)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## M          1   55084   55084    1.37 0.24914
## Ed         1  725967  725967   18.02 0.00013 ***
## Po1        1 3173852 3173852   78.80 5.3e-11 ***
## U2         1  217386  217386    5.40 0.02534 *
## Ineq       1  848273  848273   21.06 4.3e-05 ***
## Prob       1  249308  249308    6.19 0.01711 *
## Residuals 40 1611057   40276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Warning in cv.lm(uscrime3, model_lasso_final, m = 5, seed = 42):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 9
##                20    21   22    31   33     34    39      40  46
## Predicted    1203.0 783.3  728 440.4  874  997.5 786.7 1140.79 748
## cvpred       1198.8 793.6  759 459.5  887 1001.7 810.2 1146.01 792
## Crime        1225.0 742.0  439 373.0 1072  923.0 826.0 1151.00 508
```

```
## CV residual   26.2 -51.6 -320 -86.5  185  -78.7  15.8    4.99 -284
##
## Sum of squares = 234509    Mean square = 26057    n = 9
##
## fold 2
## Observations in test set: 10
##                7    8   9    15   16   29  32   35   43   44
## Predicted   733 1354 719 828.3 1004 1495 774  808 1017 1178
## cvpred      742 1409 733 836.9 1057 1649 800  874 1023 1175
## Crime       963 1555 856 798.0  946 1043 754  653  823 1030
## CV residual 221  146 123 -38.9 -111 -606 -46 -221 -200 -145
##
## Sum of squares = 578275    Mean square = 57827    n = 10
##
## fold 3
## Observations in test set: 10
##                 4     6    10   11     17  25   26  30    41  42
## Predicted   1897 730.3 787.3 1118 527.37 579 1789 668 796.4 369
## cvpred      1770 663.4 792.1 1031 541.22 605 1655 676 797.7 401
## Crime       1969 682.0 705.0 1674 539.00 523 1993 696 880.0 542
## CV residual  199  18.6 -87.1  643  -2.22 -82  338  20  82.3 141
##
## Sum of squares = 609041    Mean square = 60904    n = 10
##
## fold 4
## Observations in test set: 9
##                  1   3      5   13   23    24   37    38   47
## Predicted   810.83 386 1269.8  739  938 919.4  992 544.4  976
## cvpred      799.34 339 1302.7  717  882 941.3 1025 510.1 1010
## Crime       791.00 578 1234.0  511 1216 968.0  831 566.0  849
## CV residual  -8.34 239  -68.7 -206  334  26.7 -194  55.9 -161
##
## Sum of squares = 283612    Mean square = 31512    n = 9
##
## fold 5
## Observations in test set: 9
##                2  12    14  18   19    27     28   36   45
## Predicted   1388 673 713.6 800 1221 312.2 1259.0 1102  622
## cvpred      1396 679 724.2 721 1212 291.4 1262.5 1082  655
## Crime       1635 849 664.0 929  750 342.0 1216.0 1272  455
## CV residual  239 170 -60.2 208 -462  50.6  -46.5  190 -200
##
## Sum of squares = 426429    Mean square = 47381    n = 9
##
## Overall (Sum over all 9 folds)
##    ms
## 45359
```

```r
# CV R^2 = 1 - SSR/SST SST = sum of
# squared totals SST = sum of (Crime
# value - mean(uscrime3$Crime))^2 over
# all data points
mean_crime = mean(uscrime3$Crime)
# mean_crime
```

```r
sq_tot = (uscrime3$Crime - mean_crime)^2
# sq_tot
SST = sum(sq_tot)
# SST Extract MSE below
MSE = attr(model_lasso_final_cv, "ms")
# MSE SSR = sum of squared residuals
# for regular R^2 SSR = MSE*N for CV.
# R^2
SSR = MSE * 47
# SSR
cv_r2 = 1 - (SSR/SST)
cv_r2
```

```
## [1] 0.69
```

```r
# R^2 adj = 1 - (1-R^2)(N-1)/(N-K-1) K
# = number of predictors
cv_r2_adj = 1 - (1 - cv_r2) * (47 - 1)/(47 -
    8 - 1)
cv_r2_adj
```

```
## [1] 0.625
```

```r
# Coefficients
model_lasso_final$coefficients
```

```
## (Intercept)           M          Ed         Po1          U2        Ineq
##       905.1       132.0       219.8       341.8        75.5       269.9
##        Prob
##       -86.4
```

Analysis: after assessing the p-values, I decided to eliminate So, M.F., Pop, NW, U1, and Wealth. The final model had 6 factors (M + Ed + Po1 + U2 + Ineq + Prob). Then I ran cross-validation to assess quality of fit, and the result R^2 was 69%, and Adjusted R^2 was 62.5% (both only a very tiny bit lower than the final model using stepwise regression). The model coefficients are seen in the final line of output above.

Part 3) Elastic Net:

```r
rm(list = ls())
setwd("~/Desktop/Edx/Intro to Analytics Modeling/Week 5/WK5 Homework")
uscrime = read.table("11.1uscrimeSummer2018.txt", stringsAsFactors = FALSE, header = TRUE)
#head(uscrime)

#Scaling data except So (binary) and Crime (response)
uscrime3 = as.data.frame(scale(uscrime))
uscrime3$So = uscrime$So
uscrime3$Crime = uscrime$Crime
library(glmnet)
set.seed(12)
```

```r
#Run a loop on different values of alpha (0-1)
lambda_min_values = c()
dev_ratio_values = c()
cvm_values = c()
alpha_values = seq(0, 1, by=0.10)

for (a in alpha_values) {
  tmp_model_elasticnet = cv.glmnet(x=as.matrix(uscrime3[,-16]),
                                   y=as.matrix(uscrime3[,16]),
                                   alpha=a,
                                   nfolds=5,
                                   type.measure = "mse", #helps us automatically pick budget lambda; lambda
                                   family = "gaussian")

  cat("Alpha value: ", a, "\n")

  tmp_lambda_min = tmp_model_elasticnet$lambda.min
  lambda_min_values = c(lambda_min_values, tmp_lambda_min)
  cat("Lambda min value: ", tmp_lambda_min, "\n")

  tmp_lambda_min_position = which(tmp_model_elasticnet$glmnet.fit$lambda == tmp_lambda_min)

  tmp_dev_ratio = tmp_model_elasticnet$glmnet.fit$dev.ratio[tmp_lambda_min_position]
  dev_ratio_values = c(dev_ratio_values, tmp_dev_ratio)
  cat("Dev ratio: ", tmp_dev_ratio, "\n")

  tmp_cvm_value = tmp_model_elasticnet$cvm[tmp_lambda_min_position]
  cvm_values = c(cvm_values, tmp_cvm_value)
  cat("CVM: ", tmp_cvm_value, "\n")
  cat("\n")
}
```

```
## Alpha value:  0
## Lambda min value:  28.9
## Dev ratio:  0.772
## CVM:  77848
##
## Alpha value:  0.1
## Lambda min value:  36.4
## Dev ratio:  0.758
## CVM:  63736
##
## Alpha value:  0.2
## Lambda min value:  13.8
## Dev ratio:  0.782
## CVM:  78210
##
## Alpha value:  0.3
## Lambda min value:  16.1
## Dev ratio:  0.776
## CVM:  80563
##
## Alpha value:  0.4
```

```
## Lambda min value:  21
## Dev ratio:  0.76
## CVM:  68881
##
## Alpha value:  0.5
## Lambda min value:  10.6
## Dev ratio:  0.782
## CVM:  66771
##
## Alpha value:  0.6
## Lambda min value:  20.4
## Dev ratio:  0.748
## CVM:  69375
##
## Alpha value:  0.7
## Lambda min value:  23.1
## Dev ratio:  0.729
## CVM:  67565
##
## Alpha value:  0.8
## Lambda min value:  20.2
## Dev ratio:  0.734
## CVM:  75227
##
## Alpha value:  0.9
## Lambda min value:  3.69
## Dev ratio:  0.794
## CVM:  59965
##
## Alpha value:  1
## Lambda min value:  10.1
## Dev ratio:  0.768
## CVM:  75399
```

```r
max_dev_ratio_position = which.max(dev_ratio_values)
best_alpha = alpha_values[max_dev_ratio_position]

cat("Best alpha value: ", best_alpha, "\n")
```

```
## Best alpha value:  0.9
```

```r
cat("Max deviance ratio: ", dev_ratio_values[max_dev_ratio_position], "\n")
```

```
## Max deviance ratio:  0.794
```

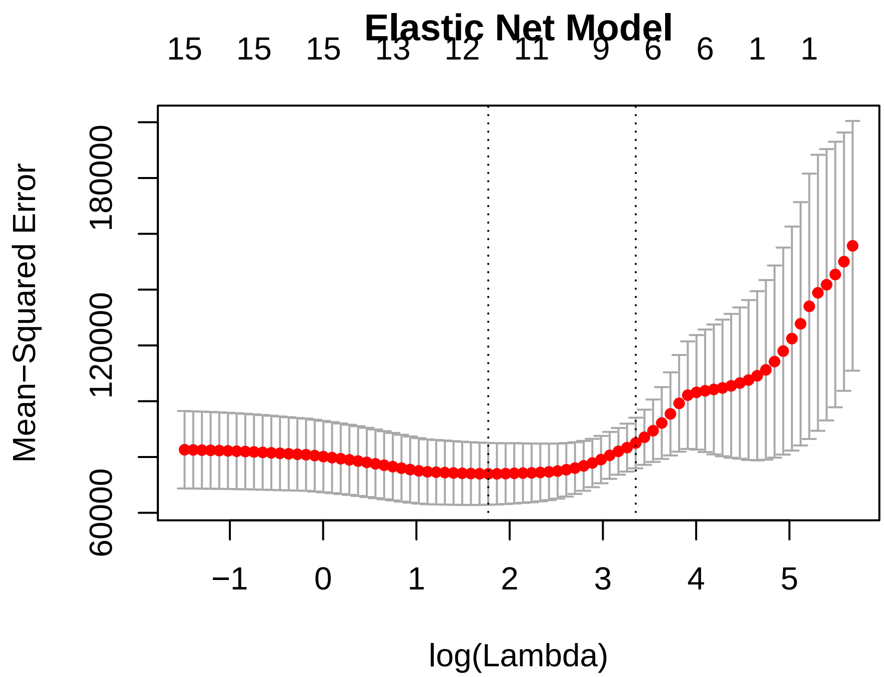
```r
cat("Lambda min: ", lambda_min_values[max_dev_ratio_position], "\n")
```

```
## Lambda min:  3.69
```

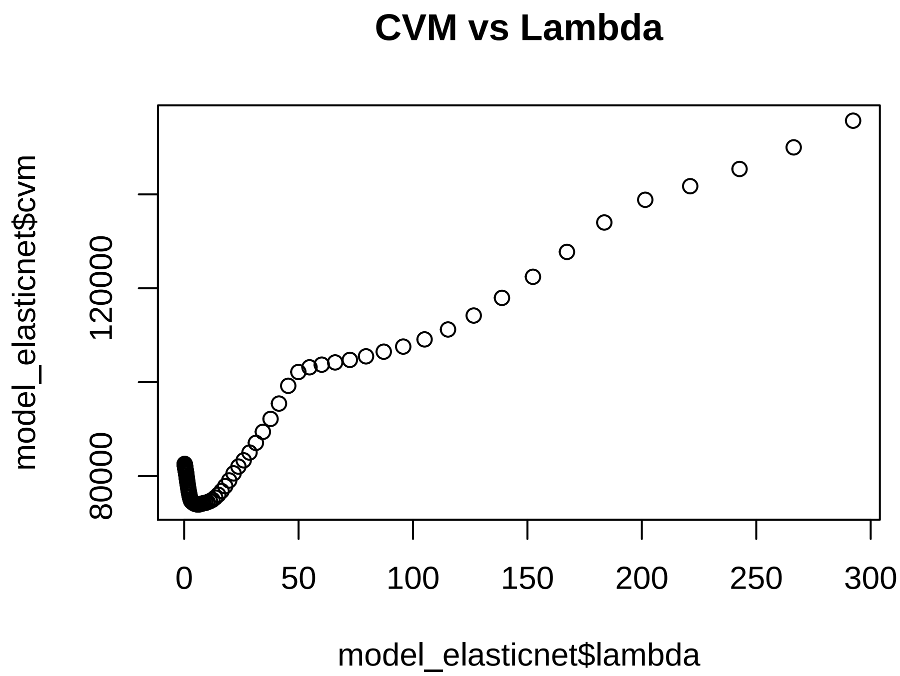
Analysis: as we can see from the output above, using a loop to compare different alpha values, the resulting best alpha value is 0.9.

Final Elastic Net model, quality of fit and coefficients:

```r
model_elasticnet = cv.glmnet(x=as.matrix(uscrime3[,-16]),
                             y=as.matrix(uscrime3[,16]),
                             alpha=best_alpha,
                             nfolds=5,
                             type.measure = "mse", #helps us automatically pick budget T lambda
                             family = "gaussian")

#model_elasticnet
plot(model_elasticnet, main = "Elastic Net Model")
```




```r
#cbind(model_elasticnet$lambda, model_elasticnet$cvm)
plot(model_elasticnet$lambda, model_elasticnet$cvm, main = "CVM vs Lambda")
```




```r
coef(model_elasticnet, s=model_elasticnet$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                 1
## (Intercept) 892.2
## M            98.8
## So           37.8
## Ed          161.2
## Po1         296.3
## Po2           .
## LF            .
## M.F          55.2
## Pop          -9.6
## NW           12.3
## U1          -59.7
## U2          100.8
## Wealth       36.4
## Ineq        226.3
## Prob        -87.2
## Time          .
```

```r
#Model using all variables selected by Elastic Net
model_elasticnet_2 = lm(formula = Crime ~ M + So + Ed + Po1 + M.F + Pop + NW + U1 + U2 + Wealth + Ineq +
                   data = uscrime3)
summary(model_elasticnet_2)
```

```
##
## Call:
```

```
## lm(formula = Crime ~ M + So + Ed + Po1 + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob, data = uscrime3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -434.2 -107.0   18.6  115.9  470.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    897.3       51.9   17.29  < 2e-16 ***
## M              112.7       49.4    2.28   0.0288 *
## So              22.9      125.3    0.18   0.8562
## Ed             195.7       62.9    3.11   0.0038 **
## Po1            293.2       65.0    4.51  7.3e-05 ***
## M.F             48.9       48.1    1.02   0.3166
## Pop            -33.3       45.6   -0.73   0.4711
## NW              19.2       57.7    0.33   0.7419
## U1             -89.8       65.7   -1.37   0.1807
## U2             140.8       66.8    2.11   0.0424 *
## Wealth          83.3       95.5    0.87   0.3893
## Ineq           285.8       85.2    3.35   0.0020 **
## Prob           -92.8       41.1   -2.26   0.0307 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 203 on 34 degrees of freedom
## Multiple R-squared:  0.797,  Adjusted R-squared:  0.726
## F-statistic: 11.1 on 12 and 34 DF,  p-value: 1.52e-08
```

```r
#Model after eliminating variables with low p-values (So, M.F.,NW, U1, Wealth)
model_elasticnet_final = lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob,
                         data = uscrime3)
summary(model_elasticnet_final)
```

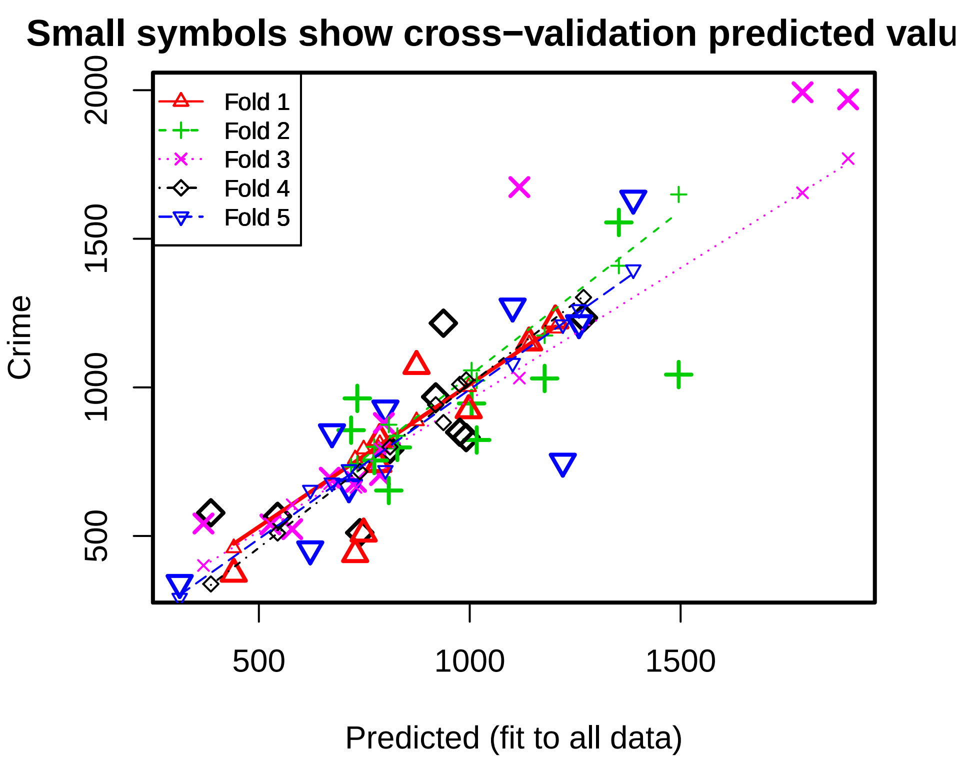
```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = uscrime3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -470.7  -78.4  -19.7  133.1  556.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    905.1       29.3   30.92  < 2e-16 ***
## M              132.0       41.8    3.15   0.0031 **
## Ed             219.8       50.1    4.39  8.1e-05 ***
## Po1            341.8       40.9    8.36  2.6e-10 ***
## U2              75.5       34.5    2.18   0.0348 *
## Ineq           269.9       55.6    4.85  1.9e-05 ***
## Prob           -86.4       34.7   -2.49   0.0171 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 201 on 40 degrees of freedom
## Multiple R-squared:  0.766,  Adjusted R-squared:  0.731
## F-statistic: 21.8 on 6 and 40 DF,  p-value: 3.42e-11
```

```
#asses quality of fit using cross-validation
#install.packages("DAAG")
library(DAAG)
model_elasticnet_final_cv = cv.lm(uscrime3,model_elasticnet_final, m=5,seed=42)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value   Pr(>F)
## M          1   55084   55084    1.37 0.24914
## Ed         1  725967  725967   18.02 0.00013 ***
## Po1        1 3173852 3173852   78.80 5.3e-11 ***
## U2         1  217386  217386    5.40 0.02534 *
## Ineq       1  848273  848273   21.06 4.3e-05 ***
## Prob       1  249308  249308    6.19 0.01711 *
## Residuals 40 1611057   40276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Warning in cv.lm(uscrime3, model_elasticnet_final, m = 5, seed = 42):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```



```
##
## fold 1
## Observations in test set: 9
##                 20    21   22    31   33     34    39      40   46
## Predicted   1203.0 783.3  728 440.4  874  997.5 786.7 1140.79  748
## cvpred      1198.8 793.6  759 459.5  887 1001.7 810.2 1146.01  792
## Crime       1225.0 742.0  439 373.0 1072  923.0 826.0 1151.00  508
## CV residual   26.2 -51.6 -320 -86.5  185  -78.7  15.8    4.99 -284
##
## Sum of squares = 234509    Mean square = 26057    n = 9
##
## fold 2
## Observations in test set: 10
##               7    8   9    15   16   29  32   35   43   44
## Predicted   733 1354 719 828.3 1004 1495 774  808 1017 1178
## cvpred      742 1409 733 836.9 1057 1649 800  874 1023 1175
## Crime       963 1555 856 798.0  946 1043 754  653  823 1030
## CV residual 221  146 123 -38.9 -111 -606 -46 -221 -200 -145
##
## Sum of squares = 578275    Mean square = 57827    n = 10
##
## fold 3
## Observations in test set: 10
##                4     6    10   11     17  25   26  30    41  42
## Predicted   1897 730.3 787.3 1118 527.37 579 1789 668 796.4 369
## cvpred      1770 663.4 792.1 1031 541.22 605 1655 676 797.7 401
## Crime       1969 682.0 705.0 1674 539.00 523 1993 696 880.0 542
## CV residual  199  18.6 -87.1  643  -2.22 -82  338  20  82.3 141
##
## Sum of squares = 609041    Mean square = 60904    n = 10
##
## fold 4
## Observations in test set: 9
##                  1   3      5   13   23    24   37    38   47
## Predicted   810.83 386 1269.8  739  938 919.4  992 544.4  976
## cvpred      799.34 339 1302.7  717  882 941.3 1025 510.1 1010
## Crime       791.00 578 1234.0  511 1216 968.0  831 566.0  849
## CV residual  -8.34 239  -68.7 -206  334  26.7 -194  55.9 -161
##
## Sum of squares = 283612    Mean square = 31512    n = 9
##
## fold 5
## Observations in test set: 9
##                2  12    14  18   19    27     28   36   45
## Predicted   1388 673 713.6 800 1221 312.2 1259.0 1102  622
## cvpred      1396 679 724.2 721 1212 291.4 1262.5 1082  655
## Crime       1635 849 664.0 929  750 342.0 1216.0 1272  455
## CV residual  239 170 -60.2 208 -462  50.6  -46.5  190 -200
##
## Sum of squares = 426429    Mean square = 47381    n = 9
##
## Overall (Sum over all 9 folds)
##    ms
## 45359
```

```r
#CV R^2 = 1 - SSR/SST
#SST = sum of squared totals
#SST = sum of (Crime value - mean(uscrime3$Crime))^2 over all data points
mean_crime = mean(uscrime3$Crime)
#mean_crime
sq_tot = (uscrime3$Crime - mean_crime)^2
#sq_tot
SST = sum(sq_tot)
#SST
#Extract MSE below
MSE = attr(model_elasticnet_final_cv,"ms")
#MSE
#SSR = sum of squared residuals for regular R^2
#SSR = MSE*N for CV. R^2
SSR = MSE*47
#SSR
cv_r2 = 1 - (SSR/SST)
cv_r2
```

```
## [1] 0.69
```

```r
#R^2 adj = 1 - (1-R^2)(N-1)/(N-K-1)
# K = number of predictors
cv_r2_adj = 1 - (1-cv_r2)*(47-1)/(47-8-1)
cv_r2_adj
```

```
## [1] 0.625
```

```r
#Coefficients
model_elasticnet_final$coefficients
```

```
## (Intercept)           M          Ed         Po1          U2        Ineq
##       905.1       132.0       219.8       341.8        75.5       269.9
##        Prob
##       -86.4
```

Analysis: using the alpha = 0.9, we get a final model using the same 6 factors (M + Ed + Po1 + U2 + Ineq + Prob) as the Lasso model. Consequently we get the same quality of fit, of R^2 69%, and Adjsuted R^2 62.5% and the same coefficents for those 6 factors.

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.

It is appropriate to use a DOE for the Census we conduct every 10 years in the U.S. It would be extremely expensive, if not impossible, to survey every person currently residing in the U.S. Consequently, we use a "representative sample".

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.

```
rm(list = ls())
# install.packages('FrF2')
library(FrF2)
```

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

## Loading required package: grid

## Loading required package: 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
```

```
set.seed(42)

FrF2(nruns = 16, nfactors = 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
```

```
# columns will be 10 different
# features, 16 rows are houses
```

Analysis: There are 10 different yes/no features. It is important to have the yes/no category for fractional design. In the above output we can take a look at the 8th row/house, for example, and see that we should only include features A and G. For the 13th row/house, for example, we should only include features B,C,G,H and K.

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: universities sending out donation requests to 1/20th of alumni association every quarter b. Geometric: how many alumni reject donation requests (failure) before one alumni agrees to donate (success) c. Poisson: number of calls to alumni center d. Exponential: time between calls to alumni center e. Weibull:time between alumni rejecting donations (failures)