ISYE6501 Homework week 5

Dylan Peters

June 14, 2017

Question 1

Using the crime data set from Homework 3, build a regression model using:

1. Stepwise regression
2. Lasso
3. Elastic net

Load the data and scale it:

```
library(caret)
```

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

```
## Loading required package: ggplot2
```

```
uscrime <- read.delim("http://www.statsci.org/data/general/uscrime.txt")
scaled.matrix <- scale(uscrime[,-16])
uscrime.scaled <- data.frame(scaled.matrix, Crime = uscrime[,16])
names(uscrime.scaled[,16]) <- c("Crime")
```

Run stepwise regression:

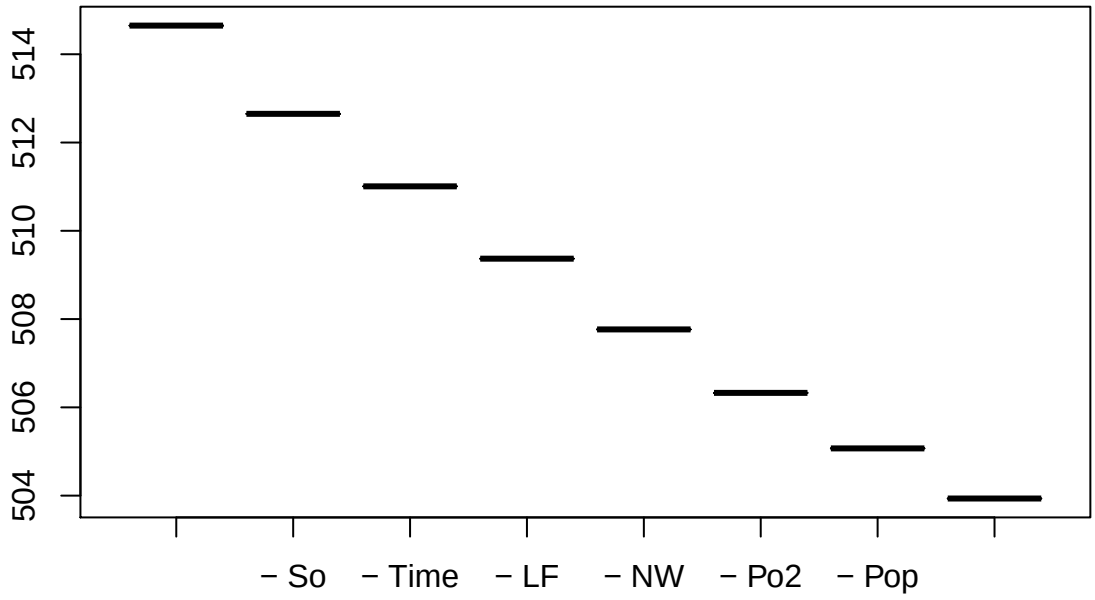
```
library(MASS)

model.base <- lm(Crime ~ ., data=uscrime.scaled)
model.step <- stepAIC(model.base, direction = "both", steps=1000, k=2, trace=0)

summary(model.step)
```

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

The final model uses the variables M, Ed, Po1, M.F, U1, U2, Ineq and Prob. Most of the p-values are significant. We can graph the improvement in AIC. The first number is the AIC for the full model, and the rest are the AIC for the variables as they are removed:

```
plot(reorder(model.step$anova$Step, 1:8), model.step$anova$AIC)
```



Run Lasso regression:

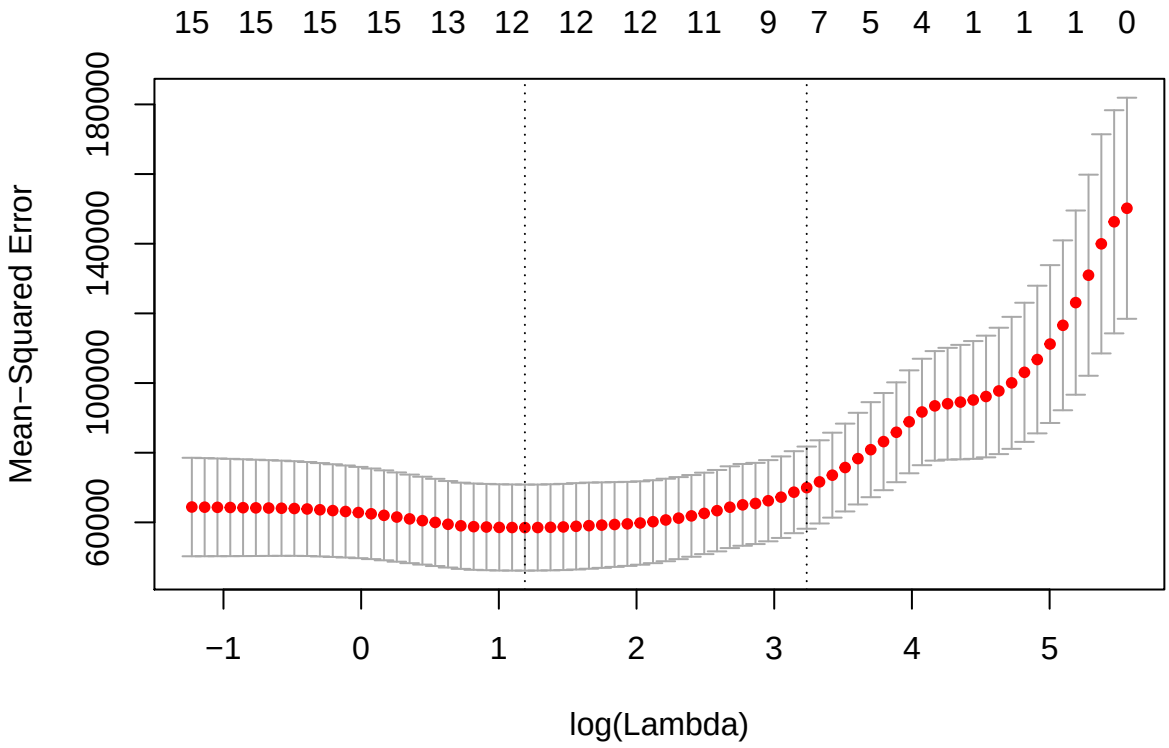
Use glmnet and set alpha = 1 for Lasso

```
library(glmnet)
```

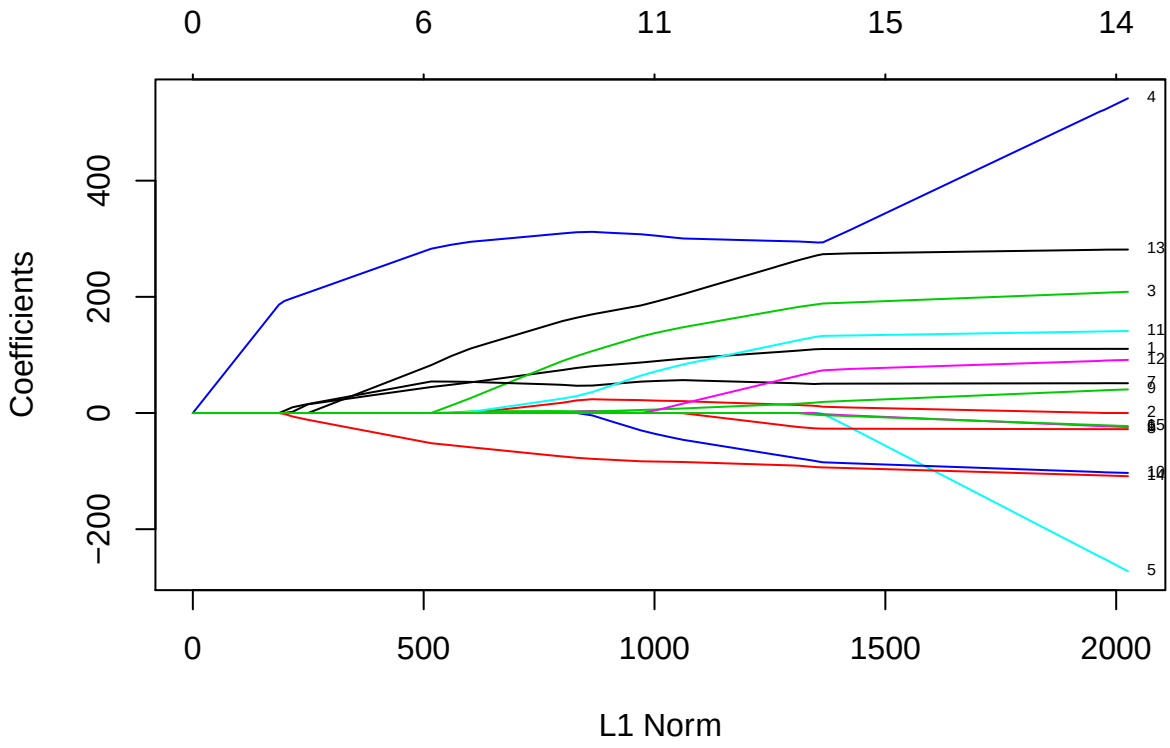
```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-10
```

```
set.seed(6501)

model.lasso <- cv.glmnet(as.matrix(uscrime.scaled[,1:15]), uscrime.scaled$Crime, family = "gaussian", al
```

```r
plot(model.lasso)
```




```r
plot(model.lasso$glmnet.fit, xvar="norm", label=TRUE)
```



The first plot shows the effect on MSE as the Lambda value is varied. The optimum value for lambda is 3.2843795 and the related MSE is 58500.96. The second plot shows the variables and how they fall out as the L1 Norm approaches 0.

The coefficients of the optimal model are below:

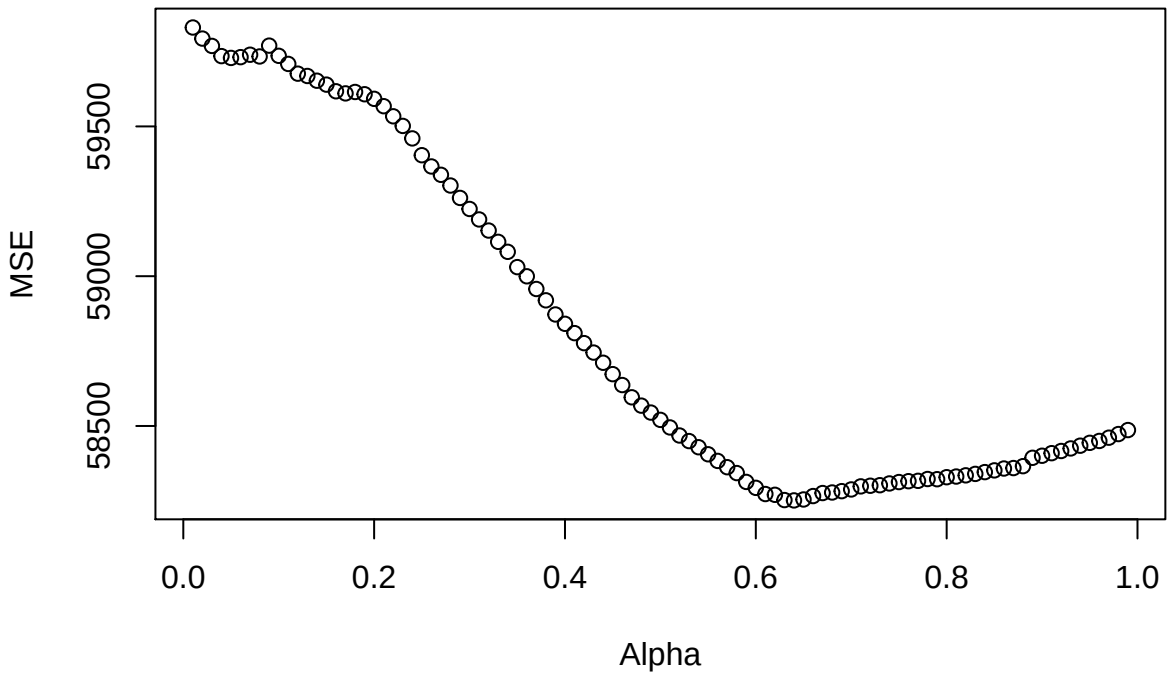
```
coef(model.lasso$glmnet.fit)[,which(model.lasso$lambda == model.lasso$lambda.min)]
```

```
## (Intercept)           M          So          Ed         Po1         Po2
##   905.08511   104.46095    15.15298   175.11702   296.32198     0.00000
##          LF         M.F         Pop          NW          U1          U2
##     0.00000    52.34557   -18.94369    14.32038   -71.07987   116.24494
##      Wealth        Ineq        Prob        Time
##    53.72200   250.17012   -89.19999     0.00000
```

Run elastic net regression:

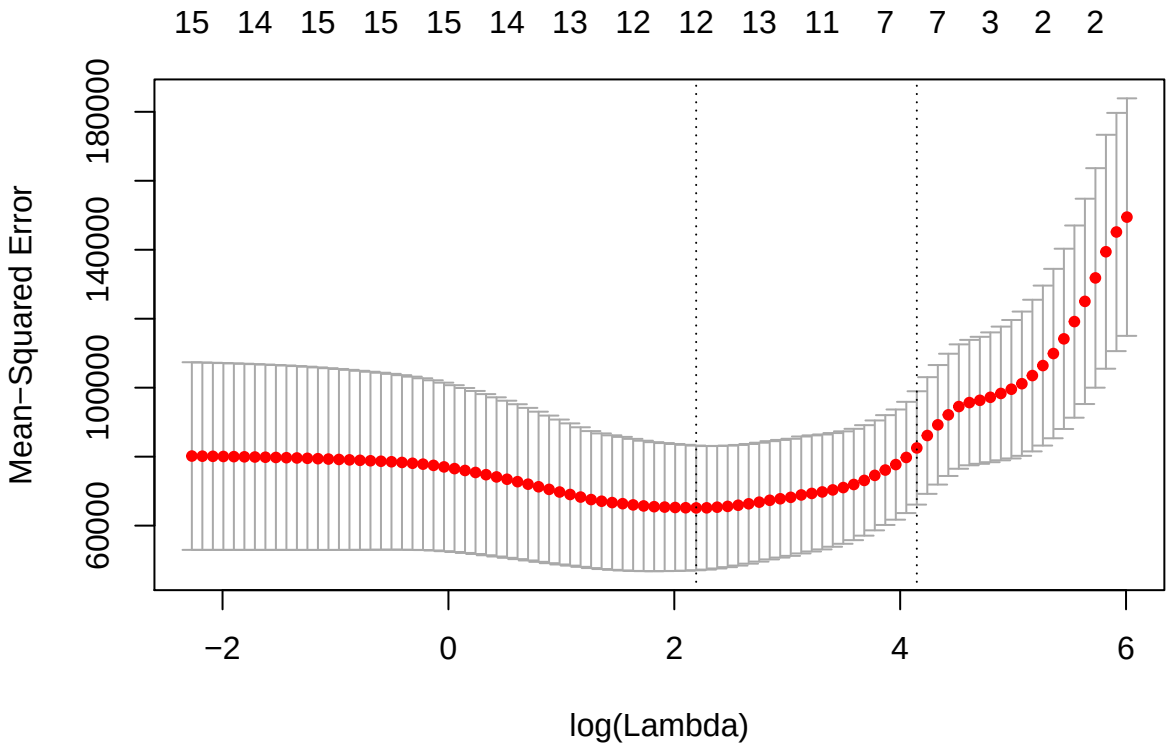
```
library(glmnet)

alphas <- seq(0.01,0.99, by=0.01)
alpha.df <- data.frame(alpha = numeric(), mse = numeric())

for (alpha in alphas)
{
  set.seed(6501)
  model.elnet <- cv.glmnet(as.matrix(uscrime.scaled[,1:15]), uscrime.scaled$Crime, family = "gaussian",
  mse <- model.elnet$cvm[which(model.elnet$lambda == model.elnet$lambda.min)]
  alpha.df <- rbind(alpha.df, c(alpha, mse))
```

```
}

colnames(alpha.df) <- c("Alpha","MSE")
plot(alpha.df)
```



The best alpha parameter is 0.64 and the related MSE is 58251.5822. So we'll use that alpha for the final model:

```
alpha <- alpha.df[which(alpha.df$MSE == min(alpha.df$MSE)),1]
model.elnet <- cv.glmnet(as.matrix(uscrime.scaled[,1:15]), uscrime.scaled$Crime, family = "gaussian", al

plot(model.elnet)
```



The optimal MSE is 65139.62. The coefficients are below:

```
coef(model.elnet$glmnet.fit)[,which(model.elnet$lambda == model.elnet$lambda.min)]
```

```
## (Intercept)           M          So          Ed         Po1         Po2
##  905.085106   94.099292   21.148947  151.240609  288.451632    0.000000
##          LF         M.F         Pop          NW          U1          U2
##    0.000000   59.783852   -3.320995   15.314980  -56.205772   93.928197
##      Wealth        Ineq        Prob        Time
##   30.639772  207.242822  -87.144663    0.000000
```

The variables Pop and Time and Po2 have been shrunk to coefficients of 0, matching our previous observations. The most important coefficients are Po1, Ineq, Ed, Po2.

Question 2

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

I work on software that analyzes call center data to improve agent performance and maintain compliance. The software can determine when agents use the right phrases, if specific customer complaints are made, sentiment, etc. One experiment might be to alter the agent's script for various phrases at different times. There could be 4 different versions of an Upsell script. A multi-armed bandit approach would to randomly gives agents one version of the script, run analysis on the recordings, and check the results for each script. The result would be either the upsell was succesful or not. Alternately, different scripts can be made for

Retention, varying the benefits offered.

Question 3

find a fractional factorial design for this experiment: what set of features should each of the 16 fictitious houses?

There are 16 houses, each of which as 10 binary (yes/no) features (factors). To make it more realistic, I will add feature names so the results look realistic.

```
library(FrF2)
```

```
## Warning: package 'FrF2' was built under R version 3.3.3
```

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

```
## Warning: package 'DoE.base' was built under R version 3.3.3
```

```
## 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
```

```
Feature.Matrix <- FrF2(nruns = 16, nfactors = 10, factor.names = c("Large.Yard","Solar","Garage","Pool"

print(Feature.Matrix)
```

```
##    Large.Yard Solar Garage Pool Basement Gas Security Brick Fireplace
## 1          -1    -1      1    1        1  -1       -1    -1        -1
## 2          -1     1     -1    1       -1   1       -1    -1        -1
## 3           1    -1      1   -1       -1   1       -1    -1         1
## 4           1    -1     -1    1       -1  -1        1     1         1
## 5          -1     1      1   -1       -1  -1        1     1        -1
## 6          -1    -1      1   -1        1  -1       -1     1         1
## 7           1    -1      1    1       -1   1       -1     1        -1
## 8           1     1     -1   -1        1  -1       -1    -1         1
## 9          -1    -1     -1   -1        1   1        1     1        -1
## 10         -1     1      1    1       -1  -1        1    -1         1
## 11         -1     1     -1   -1       -1   1       -1     1         1
## 12          1     1      1   -1        1   1        1    -1        -1
## 13          1    -1     -1   -1       -1  -1        1    -1        -1
## 14          1     1      1    1        1   1        1     1         1
## 15         -1    -1     -1    1        1   1        1    -1         1
```

```
## 16          1     1     -1    1        1  -1       -1     1        -1
##    Wood.floor
## 1           1
## 2           1
## 3           1
## 4           1
## 5           1
## 6          -1
## 7          -1
## 8           1
## 9           1
## 10         -1
## 11         -1
## 12         -1
## 13         -1
## 14          1
## 15         -1
## 16         -1
## class=design, type= FrF2
```

Each feature has 8 yes's (1's) and 8 no's (-1's), distributed across the 16 samples.

Question 4

For each of the following distributions, give an example of data that you would expect to follow this distribution (besides the examples already discussed in class).

a. Binomial

In the game Monopoly, you get an extra turn if you roll doubles on two dice. The odds of getting doubles for each throw are p = 0.167 (1/6). For each N dice throws, the number of doubles thrown is a binomial distribution.

b. Geometric

For a dice role-playing game, the odds of making a successful attack on an enemy depend on the enemy's defense, your attack strength, and a die roll (20 sided die). So if an enemy has defense 22 and your attack strength is +6, then your attack is succesful if you you roll a 16 or higher. Thus each attack has a 25% of being succesful (it is independent of any other dice roll). The number of attacks it takes to get a successful attack is a geometric distribution. The chance of success on the first attack is p = 0.25, second attack is (1-p) * p = 0.75 * 0.25, etc.

c. Poisson

At a traffic signal, the number of cars arriving from each direction for each red light can follow a Poisson distribution.

d. Exponential

Continuing the traffic example, for a traffic gate such as at a parking lot or toll booth, the time between cars arriving at the gate likely follows an exponential distribution.

e. Weibull

Software: When a software application is first released, the rate of bugs found follows a Weibull distribution with $k < 1$. Many bugs are found in the first few months of the software release. Later, as bugs are patched and the software is used by more people, the rate of bugs found starts to decrease, until it is mostly stable.