STOR767 - HW 1 Computing

Leo Li (PID: 730031954)

Problem 1: Regularization

Crime data continuation: The data `CrimeData_clean.csv` is available on Sakai.

Our goal is to find the factors which relate to violent crime. This variable is included in crime as `crime.data$violentcrimes.perpop`.

A) Exploratory data analysis (EDA)

- **Show the heatmap with mean violent crime by state. You may also show a couple of your favorate summary statistics by state through the heatmaps.**
- **Write a brief summary based on your EDA.**

In this question, we would like to show the heat map with mean violent crime by state. We can start by extracting the mean of `crime.data$violentcrimes.perpop` by state.

```
data.s <- crime.data_clean %>%
  group_by(state) %>%
  summarise(
    mean.crime=mean(violentcrimes.perpop, na.rm=TRUE),
    crime.min=min(violentcrimes.perpop, na.rm=TRUE),
    crime.max=max(violentcrimes.perpop, na.rm=TRUE),
    n=n())
```

Then, we would like to create a new data frame with mean violent crimes and corresponding state names, and switch the abbreviations of the state names to the standard state names. For example, we need to change PA to Pennsylvania, and CA to California.

```
crime <- data.s[, c("state", "mean.crime")]
crime$region <- tolower(state.name[match(crime$state, state.abb)])
```

Next, we add the center coordinate for each state `state.center` contains the coordinate corresponding to `state.abb` in order.

```
crime$center_lat  <- state.center$x[match(crime$state, state.abb)]
crime$center_long <- state.center$y[match(crime$state, state.abb)]
```

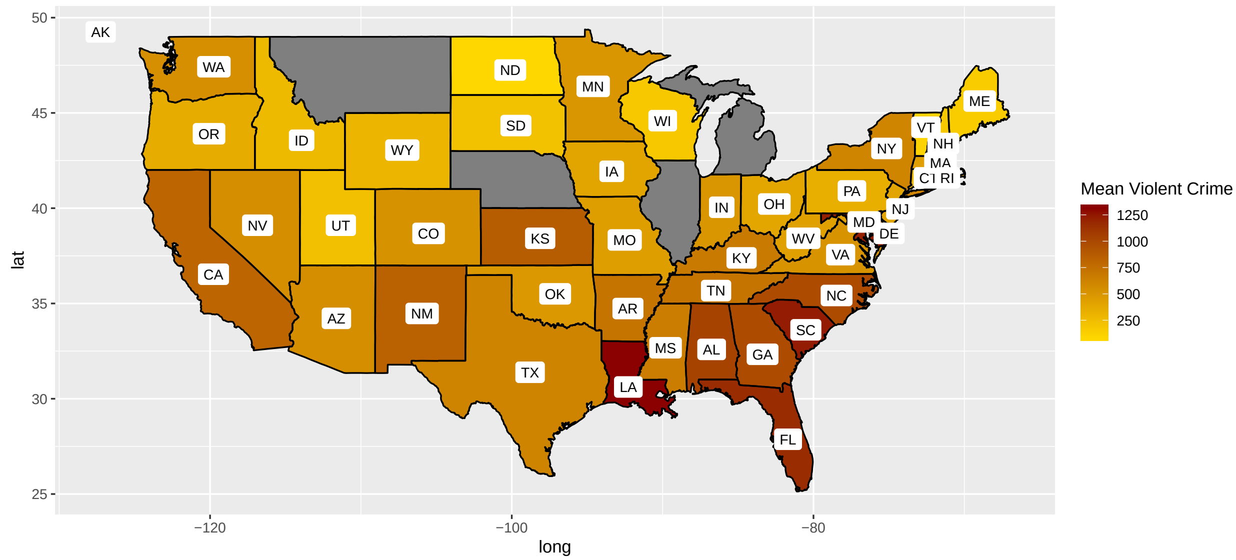
Then, we load the map of the US, in which for each state, it contains a vector of coordinates describing the shape of the state. And we would also like to combine the US map data with the violent crime data.

```
states <- map_data("state")
map <- merge(states, crime, sort=FALSE, by="region", all.x=TRUE)
map <- map[order(map$order),]
```

Now, we can plot the heatmap by using the ggplot function.

```
ggplot(map, aes(x=long, y=lat, group=group))+
  geom_polygon(aes(fill=mean.crime))+
  geom_path()+
  geom_label(data=crime,
```

```r
             aes(x=center_lat, y=center_long, group=NA, label=state),
             size=3, label.size = 0) +
  scale_fill_continuous(limits=c(min(map$mean.crime), max(map$mean.crime)),name="Mean Violent Crime",
                        low="gold1", high="red4")
```



According to the heat map, we can make the following observations:

- The south/southeast part of the US have higher rate of violent crime than other parts of the US.
- However, there are four states that do not have available data about violent crime, and they have been marked as grey on the graph, without a state name. In addition, this map only shows the situations of violent crime at the state level, which can be influenced by some outliers (for example, if there are only a few particular cities in a certain state with high rate of violent crime, then the mean violent crime for the whole state will rise rapidly, so that the heat map may not be a good representation of the situation of the violent crimes of a particular region at a finer scale).

B) We use a subset of the crime data discussed in class, but only look at Florida and California.

Use LASSO to choose a reasonable, small model. Fit an OLS model with the variables obtained. The final model should only include variables with p-values < 0.05. Note: you may choose to use lambda 1se or lambda min to answer the following questions where apply.

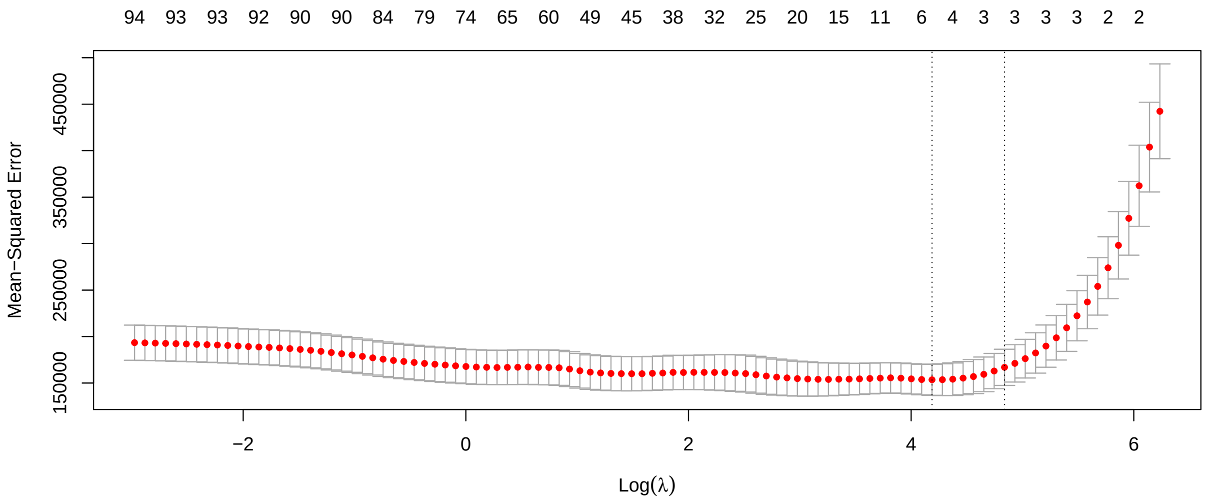
1. What is the model reported by LASSO?

First of all, we prepare our data by using Y to store the response and X to store the design matrix.

```r
Y <- crime.fl.ca[, 99]
X.fl.ca <- model.matrix(violentcrimes.perpop~., data=crime.fl.ca)[, -1]
```

Then, we use glmnet to fit LASSO on the crime data, and choose λ_{1se} as the tuning parameter.

```r
fit.lambda <- glmnet(X.fl.ca, Y, alpha=1)
# str(fit.lambda)
set.seed(233)
fit.cv <- cv.glmnet(X.fl.ca, Y, alpha=1, nfolds=20)
plot(fit.cv)
```




```
coef.1se <- coef(fit.cv, s="lambda.1se")
coef.1se <- coef.1se[which(coef.1se !=0),]
coef.1se
```

```
##         (Intercept)       race.pctblack    pct.kids2parents pct.kids.nvrmarried
##         1790.467384            7.375466          -17.786584           76.856774
```

Then, the model reported by LASSO has the covariates of `race.pctblack`, `pct.kids2parents`, and `pct.kids.nvrmarried` in addition to the intercept. Their corresponding LASSO estimates have also been shown above.

2. What is the model after running OLS?

Now, to make inference using the LASSO chosen variables, we run the OLS analysis on the LASSO chosen variables, by assuming all the linear model assumptions are satisfied.

```
coef.min <- coef(fit.cv, s="lambda.1se")
coef.min <- coef.min[which(coef.min !=0),]
var.min <- rownames(as.matrix(coef.min))
lm.input <- as.formula(paste("violentcrimes.perpop", "~", paste(var.min[-1], collapse = "+")))
lm.input
```

```
## violentcrimes.perpop ~ race.pctblack + pct.kids2parents + pct.kids.nvrmarried
```

Now, we print the OLS estimates to the regression coefficients of the model specified above:

```
fit.min.lm <- lm(lm.input, data=crime.fl.ca)
lm.output <- coef(fit.min.lm)
summary(fit.min.lm)
```

```
##
## Call:
## lm(formula = lm.input, data = crime.fl.ca)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1115.29  -210.52   -37.48   155.25  1911.97
##
## Coefficients:
```

```
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2012.949    266.282   7.559 3.32e-13 ***
## race.pctblack          13.956      2.742   5.089 5.78e-07 ***
## pct.kids2parents      -22.678      3.371  -6.728 6.70e-11 ***
## pct.kids.nvrmarried    94.953     12.269   7.739 9.95e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 378.3 on 364 degrees of freedom
## Multiple R-squared:  0.6791, Adjusted R-squared:  0.6765
## F-statistic: 256.8 on 3 and 364 DF,  p-value: < 2.2e-16
```

3. What is your final model, after excluding high p-value variables? You will need to use model selection method to obtain this final model. Make it clear what criterion/criteria you have used and justify why they are appropriate.

From the OLS estimates that have been shown above, we can see that all the variables chosen are significant at 0.05 level, so that the model in the previous part is our final model. To sum up, our final model has the covariates of `race.pctblack`, `pct.kids2parents`, and `pct.kids.nvrmarried` in addition to the intercept, and the estimated regression coefficients have also been shown in the previous part.

In this question, we use LASSO regression as a method of variable selection, and we also use 20-folds cross-validation to assess the various choices of the tunning parameter. Then, since we would like to find a small model, we follow the One SD Rule to pick the most parsimonious model whose CV score is no more than one SD of λ_{min}, where the SD is calculated as the standard deviation out of the 20 CV scores.

C) Now, instead of Lasso, we want to consider how changing the value of alpha (i.e. mixing between Lasso and Ridge) will affect the model. Cross-validate between alpha and lambda, instead of just lambda. Note that the final model may have variables with p-values higher than 0.05; this is because we are optimizing for accuracy rather than parsimoniousness.

1. What is your final elastic net model? What were the alpha and lambda values? What is the prediction error?

In this question, we run the elastic net on our data set, and cross-validate between alpha and lambda. Particularly, we conducted a grid search that considers the alpha values from 0.6 to 1, with an increment of 0.02 in each choice of the alpha level. Within each choice of alpha value, we choose the λ_{1se} value. Then, we compare the cv scores for all the models corresponding to the λ_{1se} for all the alpha values. Finally, we take the model with the lowest cv score as our final elastic net model.

```r
set.seed(2333)
alpha <- 0.6
fit.en.cv <- cv.glmnet(X.fl.ca, Y, alpha = alpha, nfolds=20)
coef.en.1se <- coef(fit.en.cv, s="lambda.1se")
coef.en.1se <- coef.en.1se[which(coef.en.1se !=0),]
cv.1se <- fit.en.cv$cvm[match(fit.en.cv$lambda.1se, fit.en.cv$lambda)]
lambda <- fit.en.cv$lambda.1se
for (i in 1:20){
  set.seed(200*i)
  alpha1 <- 0.6 + i*0.02
  fit.en.cv1 <- cv.glmnet(X.fl.ca, Y, alpha = alpha1, nfolds=20)
  coef.en.1se1 <- coef(fit.en.cv1, s="lambda.1se")
  coef.en.1se1 <- coef.en.1se1[which(coef.en.1se1 !=0),]
  cv.1se1 <- fit.en.cv1$cvm[match(fit.en.cv1$lambda.1se, fit.en.cv1$lambda)]
  lambda1 <- fit.en.cv1$lambda.1se
  if(cv.1se1 < cv.1se){
    alpha <- alpha1
```

```r
    cv.1se <- cv.1se1
    coef.en.1se <- coef.en.1se1
    lambda <- lambda1
  }else{
    alpha <- alpha
    cv.1se <- cv.1se
    coef.en.1se <- coef.en.1se
    lambda <- lambda
  }
}
```

If we follow the algorithm mentioned above, our final model has the following variables and their corresponding regression coefficients are estimated as:

```r
coef.en.1se
```

```
##           (Intercept)         race.pctblack       pct.fam2parents 
##           1921.081056              9.269339             -1.205247 
##      pct.kids2parents pct.youngkids2parents   pct.kids.nvrmarried 
##            -15.189998             -2.838765             72.859485 
##      pct.house.vacant 
##              4.894572
```

Our alpha value is,

```r
alpha
```

```
## [1] 0.64
```

The lambda value is,

```r
lambda
```

```
## [1] 149.3133
```

And finally, the prediction error, which is estimated by cross-validation, equals to

```r
cv.1se
```

```
## [1] 160656.6
```

2. Use the elastic net variables in an OLS model. What is the equation, and what is the prediction error?

Now, we would like to use the elastic net variables in an OLS model, and the equation is the following:

```r
var.en.min <- rownames(as.matrix(coef.en.1se))
lm.input <- as.formula(paste("violentcrimes.perpop", "~", paste(var.en.min[-1], collapse = "+")))
lm.input
```

```
## violentcrimes.perpop ~ race.pctblack + pct.fam2parents + pct.kids2parents + 
##     pct.youngkids2parents + pct.kids.nvrmarried + pct.house.vacant
```

The estimated regression coefficients are,

```r
fit.min.ols <- lm(lm.input, data=crime.fl.ca)
lm.output <- coef(fit.min.ols)
summary(fit.min.ols)
```

```
## 
## Call:
## lm(formula = lm.input, data = crime.fl.ca)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1060.96  -212.11   -42.18   153.21  1886.31
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1922.322    294.986   6.517 2.42e-10 ***
## race.pctblack           13.431      2.761   4.865 1.71e-06 ***
## pct.fam2parents         11.859     10.528   1.126  0.26074
## pct.kids2parents       -31.798     10.419  -3.052  0.00244 **
## pct.youngkids2parents   -1.763      5.157  -0.342  0.73264
## pct.kids.nvrmarried     80.640     13.459   5.992 5.03e-09 ***
## pct.house.vacant        27.718     11.055   2.507  0.01260 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 376 on 361 degrees of freedom
## Multiple R-squared:  0.6857, Adjusted R-squared:  0.6805
## F-statistic: 131.3 on 6 and 361 DF,  p-value: < 2.2e-16
```

Then, we would like to use cross-validation to estimate the prediction error:

```
set.seed(2333)
train.control <- trainControl(method = "cv", number = 20)
lm.cv <- train(lm.input, data = crime.fl.ca, method = "lm",
               trControl = train.control)
print(lm.cv)
```

```
## Linear Regression
##
## 368 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (20 fold)
## Summary of sample sizes: 349, 349, 350, 351, 349, 350, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   375.4458  0.683126  271.9399
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

From the output above, the RMSE equals 375.45, so that the estimated prediction error equals $375.45^2 = 140961$.

3. Summarize your findings, with particular focus on the difference between the two equations.

After running the elastic net, our final model includes the following variables: `race.pctblack`, `pct.fam2parents`, `pct.kids2parents`, `pct.youngkids2parents`, `pct.kids.nvrmarried`, and `pct.house.vacant`. Among these variables, `race.pctblack`, `pct.kids2parents`, `pct.kids.nvrmarried`, and `pct.house.vacant` have p-values less than 0.05. The other two variables are also included because we are optimizing for accuracy rather than parsimoniousness.

The following output compares the regression coefficients estimated from elastic net to those from ordinary least square:

```
comp <- data.frame(coef.en.1se, lm.output )
names(comp) <- c("estimates from Elastic Net", "lm estimates")
comp
```

```
##                        estimates from Elastic Net lm estimates
## (Intercept)                           1921.081056  1922.321884
## race.pctblack                            9.269339    13.431224
## pct.fam2parents                         -1.205247    11.859043
## pct.kids2parents                       -15.189998   -31.798270
## pct.youngkids2parents                   -2.838765    -1.763001
## pct.kids.nvrmarried                     72.859485    80.640325
## pct.house.vacant                         4.894572    27.717915
```

From the output, we can see that in general, the estimated regression coefficients from an OLS model often has larger absolute values, but there can be some exceptions. In addition, the estimated prediction error from OLS model, which is 140961, is smaller than that from elastic net, which is 160657, which implies that the OLS with elastic net variables can have better prediction than the elastic net model.