# Regression

[Dina Demchenko]

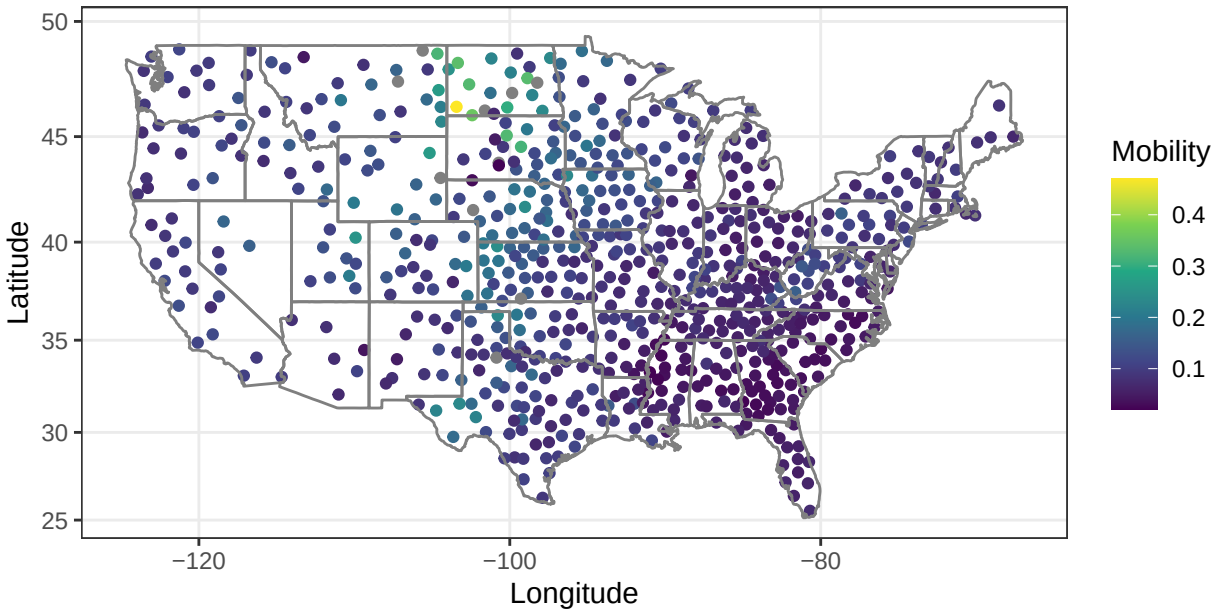
2024-02-01

## Regularized methods

I'm analyzing `mobility` data to examine economic mobility across generations in modern USA. This data, derived from extensive tax records, links adult income to parental income from previous decades. It presents aggregate statistics on economic mobility for hundreds of communities, covering most of the American population, along with covariate information. My focus is on predicting economic mobility based on community characteristics.

1. Generating a map of `Mobility` ignoring Alaska and Hawaii.

```
ggplot(
  mobility %>% filter(!(State %in% c("AK", "HI") )),
  aes(x=Longitude, y=Latitude, color = Mobility)) +
  geom_point() +
  coord_map() +
  borders("state") +
  scale_color_viridis_c()
```

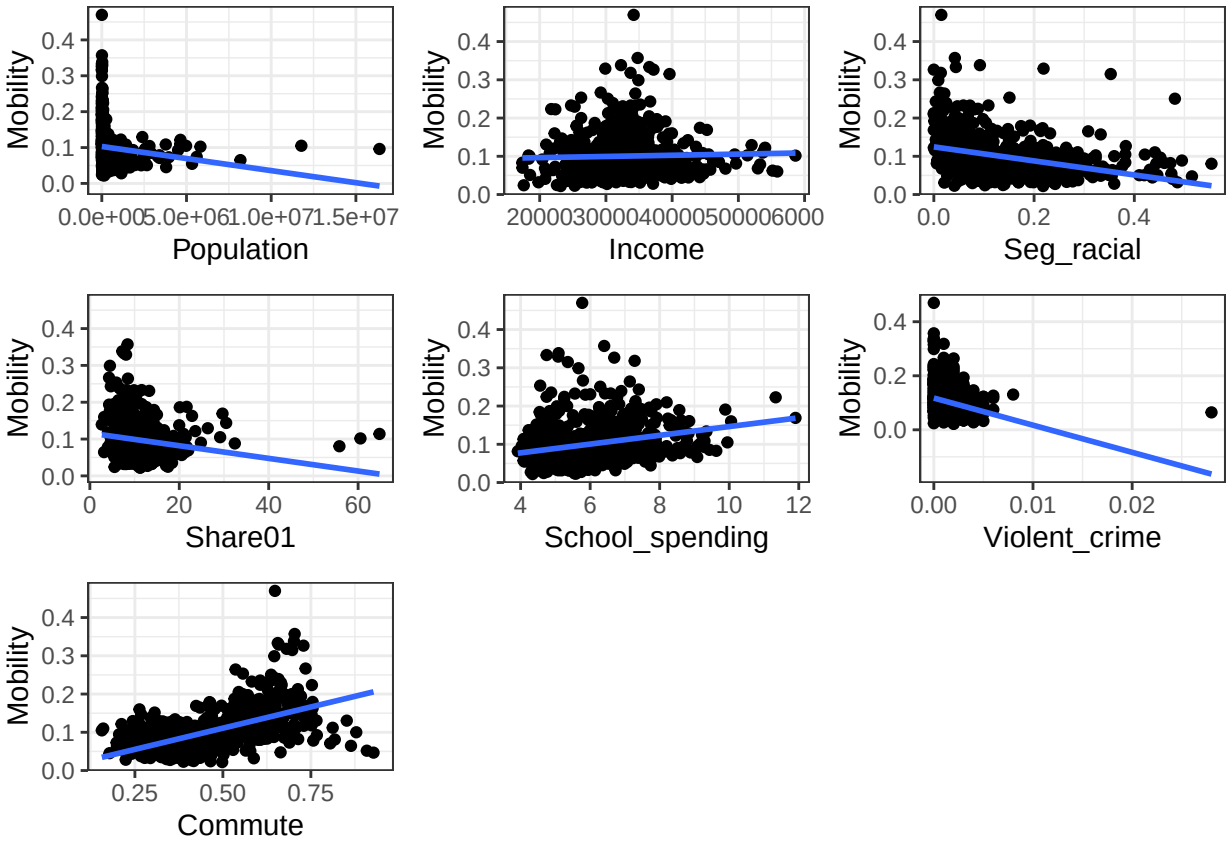



According to the map above, North Dakota has one of the highest mobility rates across the US. One community in North Dakota has the highest mobility rate across the whole US (mobiloty higher than 0.4). Other states in the rural Midwest (South Dakota, Nebraska, Kansas, Colorado, etc.) seem to also have a higher mobility rate compared with the states on both East and West coasts. The East Coast has a lot more observations than the West coast. Overall, states on the West Coast (California, Washington, Oregon, Nevada) and on the East Coast (New York, New Jersey, Connecticut, Pennsylvania) don't have as high mobility rates. This may be due to the fact that those states always had a higher standard of living, hence, their growth is not as significant as an increase in living conditions for those in the middle states of the US. The Southeast of the US seems to have the lowest mobility rate across the whole country.

2. Generating scatter plots depicting mobility against the variables: : `Population`, `Income`, `Seg_racial`, `Share01`, `School_spending`, `Violent_crime`, and `Commute`. Including on each plot a line for the univariate regression of mobility on the variable, and giving a table of the slope coefficients. Then explaining the interpretation of each coefficient in the context of the problem.

```
plot1 <- ggplot(mobility, aes(Population, Mobility)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)

plot2 <- ggplot(mobility, aes(Income, Mobility)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)

plot3 <- ggplot(mobility, aes(Seg_racial, Mobility)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)
```

```
plot4 <- ggplot(mobility, aes(Share01, Mobility)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)

plot5 <- ggplot(mobility, aes(School_spending, Mobility)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)

plot6 <- ggplot(mobility, aes(Violent_crime, Mobility)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)

plot7 <- ggplot(mobility, aes(Commute, Mobility)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)
require(gridExtra)
grid.arrange(plot1, plot2,plot3, plot4, plot5, plot6, plot7, ncol=3, nrow=3)
```




Here is the table of the slope coefficients:

```
lm1 <- lm(Mobility~Population, data=mobility)
lm2 <- lm(Mobility~Income, data=mobility)
lm3 <- lm(Mobility~Seg_racial, data=mobility)
lm4 <- lm(Mobility~Share01, data=mobility)
lm5 <- lm(Mobility~School_spending, data=mobility)
```

```r
lm6 <- lm(Mobility~Violent_crime, data=mobility)
lm7 <- lm(Mobility~Commute, data=mobility)

df <- data.frame(variable=c("`Population`", "`Income`", "`Seg_racial`",
"`Share01`", "`School_spending`", "`Violent_crime`","`Commute`"),
                 slope=c(lm1$coefficients[2], lm2$coefficients[2],lm3$coefficients[2],
lm4$coefficients[2], lm5$coefficients[2], lm6$coefficients[2], lm7$coefficients[2]))
head(df, n=7)
```

```
##                          variable         slope
## Population           ‘Population‘ -6.732228e-09
## Income                   ‘Income‘  3.094097e-07
## Seg_racial           ‘Seg_racial‘ -1.835019e-01
## Share01                ‘Share01‘ -1.718199e-03
## School_spending ‘School_spending‘  1.146429e-02
## Violent_crime     ‘Violent_crime‘ -1.005057e+01
## Commute                ‘Commute‘  2.218695e-01
```

*Interpretation of each coefficient:*

Our response variable is `Mobility` which is the probability that a child born in 1980–1982 into the lowest quintile (20%) of household income will be in the top quantile at age 30. Individuals are assigned to the community they grew up in, not the one they were in as adults. (according to the data source)

Population: If the population in commuting zone (CZ) increases by 1 person, then the mobility rate of this city will drop by about $6.732228 * 10(-9)$.
Income: If the average income per capita in 2000 increases by 1 US dollar, then the mobility rate of this city will increase by about $3.094097 * 10^( - 7)$.
Racial Segregation: Racial Segregation is a measure of residential segregation by race (value from 0 to 1). Hence, if this measure increases by 0.1, the the mobility rate will drop by 0.01835019.
Share1: If the share of the total income of a community going to its richest 1% increases by 1, then mobility rate will decrease by about 0.001718199.
School spending: If average spending per pupil in public schools grows by 1$, then mobility rate in the city will increase by 0.01146429.
Violent crime: If arrests per person per year for violent crimes in a given city increases by 1 arrest, the mobility rate will drop by approximately 10.05057.
Commute: If the fraction of workers with a commute of less than 15 minutes increases by 0.1 (since it is the value from 0 to 1), then mobility rate will grow by 0.02218695.

3. Conducting a linear regression of `mobility` against suitable covariates. Presenting all regression coefficients and their standard errors with appropriate precision. Comparing coefficients from problem 2 with those from this regression to assess differences in magnitude, changes in sign, if any, and overall variation.

**a)**

```r
# Removing ID, State, Name
drops <- c("ID","State", "Name")
mobility <-mobility[ , !(names(mobility) %in% drops)]

# Running the linear regression
ols <- lm(Mobility ~ ., data = mobility)
df1 <- data.frame(coefficient=c(signif((summary(ols)$coefficients[2:length(mobility)]), 4)),
```

```
                St.Error=c(signif((summary(ols)$coefficients[2:length(mobility) , 2]), 4)))
head(df1, length(mobility))
```

```
##                            coefficient  St.Error
## Population                   1.561e-09 2.172e-09
## Urban                        1.568e-03 3.515e-03
## Black                        8.856e-02 2.540e-02
## Seg_racial                  -4.837e-02 1.654e-02
## Seg_income                   1.064e+00 8.313e-01
## Seg_poverty                 -8.582e-01 4.471e-01
## Seg_affluence               -3.178e-01 4.163e-01
## Commute                      7.548e-02 2.551e-02
## Income                       3.059e-07 5.984e-07
## Gini                         2.929e+00 2.888e+00
## Share01                     -2.937e-02 2.889e-02
## Gini_99                     -3.033e+00 2.888e+00
## Middle_class                 8.649e-02 4.265e-02
## Local_tax_rate               1.329e-01 2.379e-01
## Local_gov_spending           9.929e-07 2.761e-06
## Progressivity                5.602e-03 1.119e-03
## EITC                        -5.897e-04 4.092e-04
## School_spending             -1.286e-03 2.066e-03
## Student_teacher_ratio       -5.020e-04 1.021e-03
## Test_scores                  4.603e-04 2.758e-04
## HS_dropout                  -1.918e-01 7.679e-02
## Colleges                    -1.053e-01 7.219e-02
## Tuition                     -3.329e-08 4.002e-07
## Graduation                  -1.386e-02 1.264e-02
## Labor_force_participation   -6.895e-02 4.756e-02
## Manufacturing               -1.727e-01 2.528e-02
## Chinese_imports             -8.122e-04 6.989e-04
## Teenage_labor               -2.125e+00 1.928e+00
## Migration_in                -8.819e-02 2.763e-01
## Migration_out               -5.249e-01 3.380e-01
## Foreign_born                 1.071e-01 4.983e-02
## Social_capital              -2.021e-03 2.430e-03
## Religious                    6.082e-02 1.157e-02
## Violent_crime               -3.194e+00 1.481e+00
## Single_mothers              -3.469e-01 8.331e-02
## Divorced                     7.964e-02 1.417e-01
## Married                     -8.914e-02 6.706e-02
## Longitude                    1.129e-04 2.049e-04
## Latitude                     1.424e-03 5.312e-04
```

**b)** Variable `ID` should not be included because it simply represents a numerical code, identifying the community. This variable is created just for the analysis convenience since some commuting zone/cities have long names and having an ID values makes it easier and faster to identify a zone. This value does not affect the mobility rate in any way.

**c)** At first I removed `Name`, since they represents names of the community and do not affect the mobility rate. This is the same as `ID` but just a text. After removing `ID` and `Name`, the linear regression still had two present singularities. This may be due to the variable `State`. This variable also just represents the name of the state (characters) and does not also affect the mobility. Hence, as a result we should exclude `ID`, `State`, `Name` from our model.

**d)** *Comparing the coefficients I found in problem 2 to the coefficients for the same variables in this regression. Are they much different? Have any changed sign?*

```
df2 <-
  data.frame(
    variable = c(
      "`Population`",
      "`Income`",
      "`Seg_racial`",
      "`Share01`",
      "`School_spending`",
      "`Violent_crime`",
      "`Commute`"
    ),
    coef_in_Q2 = signif(c(
        lm1$coefficients[2],
        lm2$coefficients[2],
        lm3$coefficients[2],
        lm4$coefficients[2],
        lm5$coefficients[2],
        lm6$coefficients[2],
        lm7$coefficients[2]), 4),

    coef_in_Q3 = signif(c(
        ols$coefficients["Population"],
        ols$coefficients["Income"],
        ols$coefficients["Seg_racial"],
        ols$coefficients["Share01"],
        ols$coefficients["School_spending"],
        ols$coefficients["Violent_crime"],
        ols$coefficients["Commute"]), 4))

head(df2, 7)
```

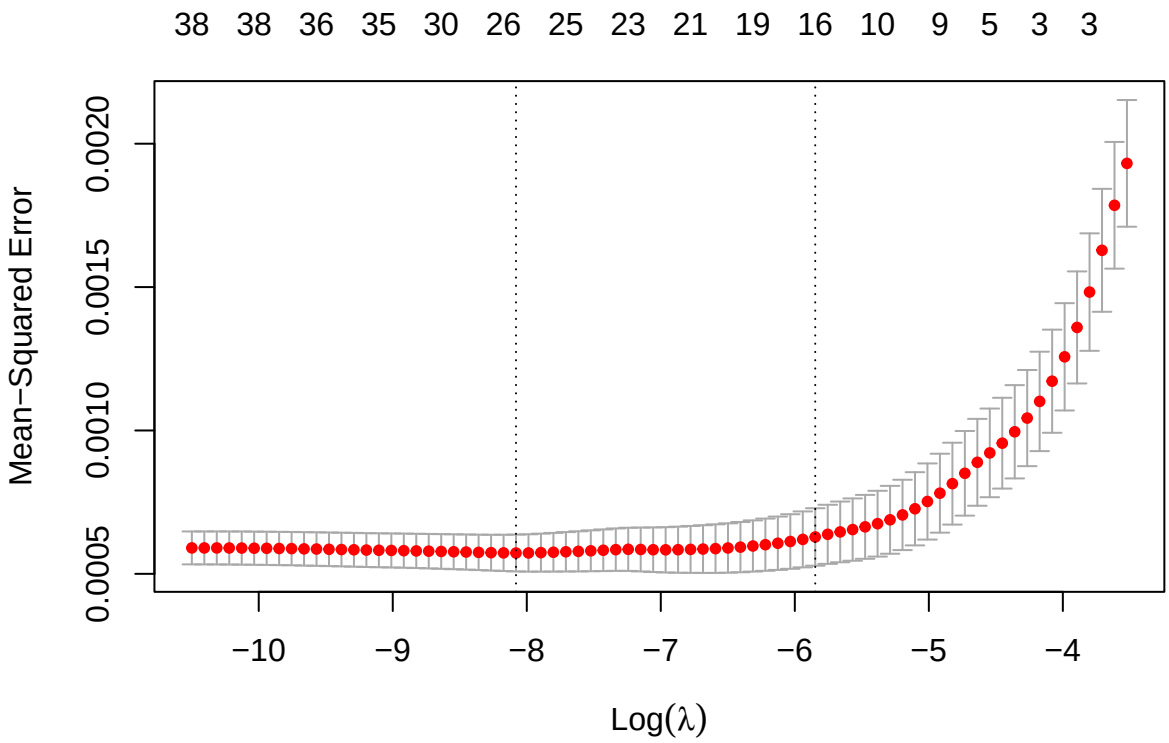
```
##                          variable coef_in_Q2 coef_in_Q3
## Population           `Population` -6.732e-09  1.561e-09
## Income                   `Income`  3.094e-07  3.059e-07
## Seg_racial           `Seg_racial` -1.835e-01 -4.837e-02
## Share01                 `Share01` -1.718e-03 -2.937e-02
## School_spending `School_spending`  1.146e-02 -1.286e-03
## Violent_crime     `Violent_crime` -1.005e+01 -3.194e+00
## Commute                 `Commute`  2.219e-01  7.548e-02
```

Coefficients for the same variables in Q2 and Q3 models are very different, except for the variable `Income` which has almost the similar coefficient in both models. Coefficients for variables like `Population` and `School_spending` have an opposite sign in the model Q3 as opposed to their sign in Q2.(These differences in coefficients may be due to the fact that in Q2 the regression was univariate since we regressed mobility on each variable separately. In model in Q3 the regression is multivariate since mobility is regressed on all the variables at the same time.) For the variable Share1 in model for Q3 there is a large increase in the coefficient value as compared to its value in Q2.

4. With all the covariates I used in the previous question, using ridge regression and lasso (with the `{glmnet}` package). Using cross validation as implemented in `cv.glmnet()` I am plotting the CV curve for Lasso and the coefficient trace for ridge regression.

**a)**

```
library(glmnet)
x <- model.matrix(ols) # grabs the design matrix from the internal lm() output
x <- x[, -1] # remove the intercept column, glmnet() will do this for us
y <- mobility$Mobility[-ols$na.action] # remove those observations with
# missing values. glmnet() hates those. They are already dropped in x.

set.seed(01101101)
# LASSO
lasso <- cv.glmnet(x, y) #alpha=1 default
plot(lasso)
```

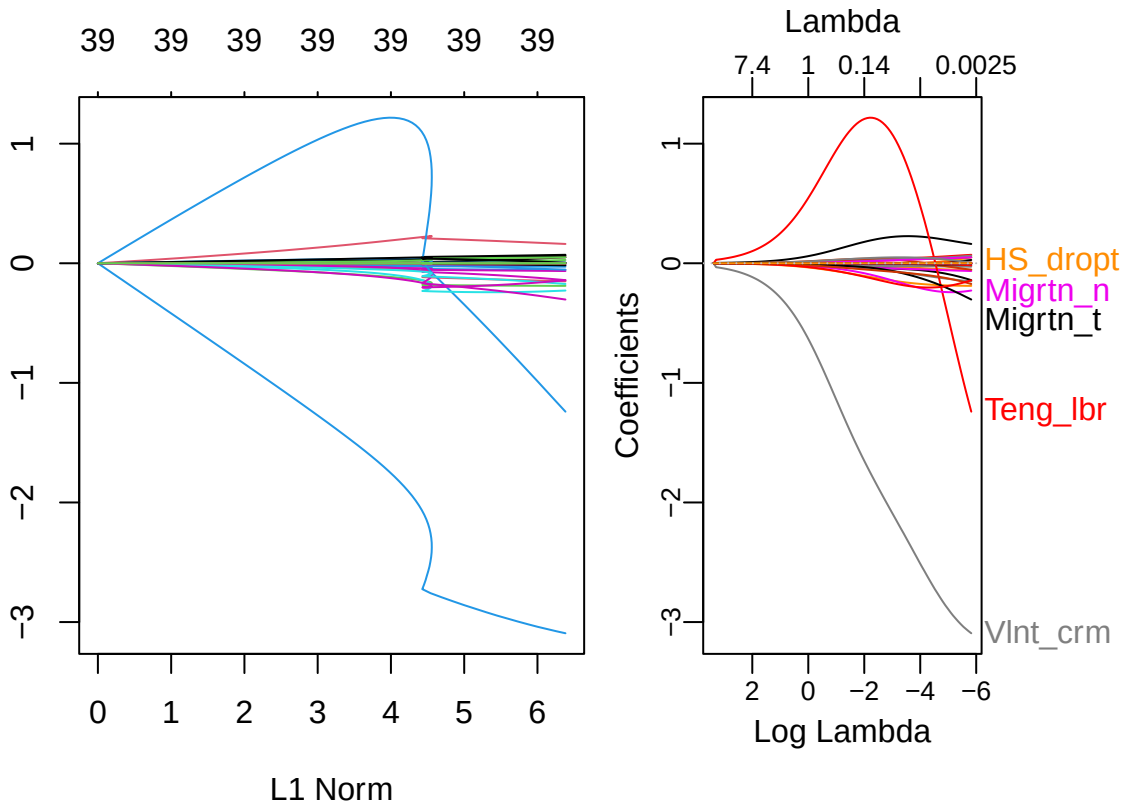



*Explaining the difference between the two vertical lines shown on the figure.* These vertical lines correspond to different ways of choosing lambda. In CV-plot, the vertical line on the represents the value of log(lambda) that minimizes the cv error. The right vertical line represents `lambda.1se`: the value of log(lambda) that allows to be within 1 standard error of the lowest cv error. The line corresponding to 1se has less predictors in the model. Note that numbers at the top of the plot represent the number of coefficients for predictors in the model (or number of predictors that are present in the model).

**b)** *Plotting the coefficient trace for ridge regression.*

```
set.seed(01101101)
# Ridge regression
ridge <- cv.glmnet(
    x = x ,
```

```
    y = y,
    alpha = 0)

par(mfrow =c(1,2), mar =c(5,3,3,0.1))
plot(ridge$glmnet.fit)

library(plotmo) # for plot_glmnet
plot_glmnet(ridge$glmnet.fit, label=5)
```

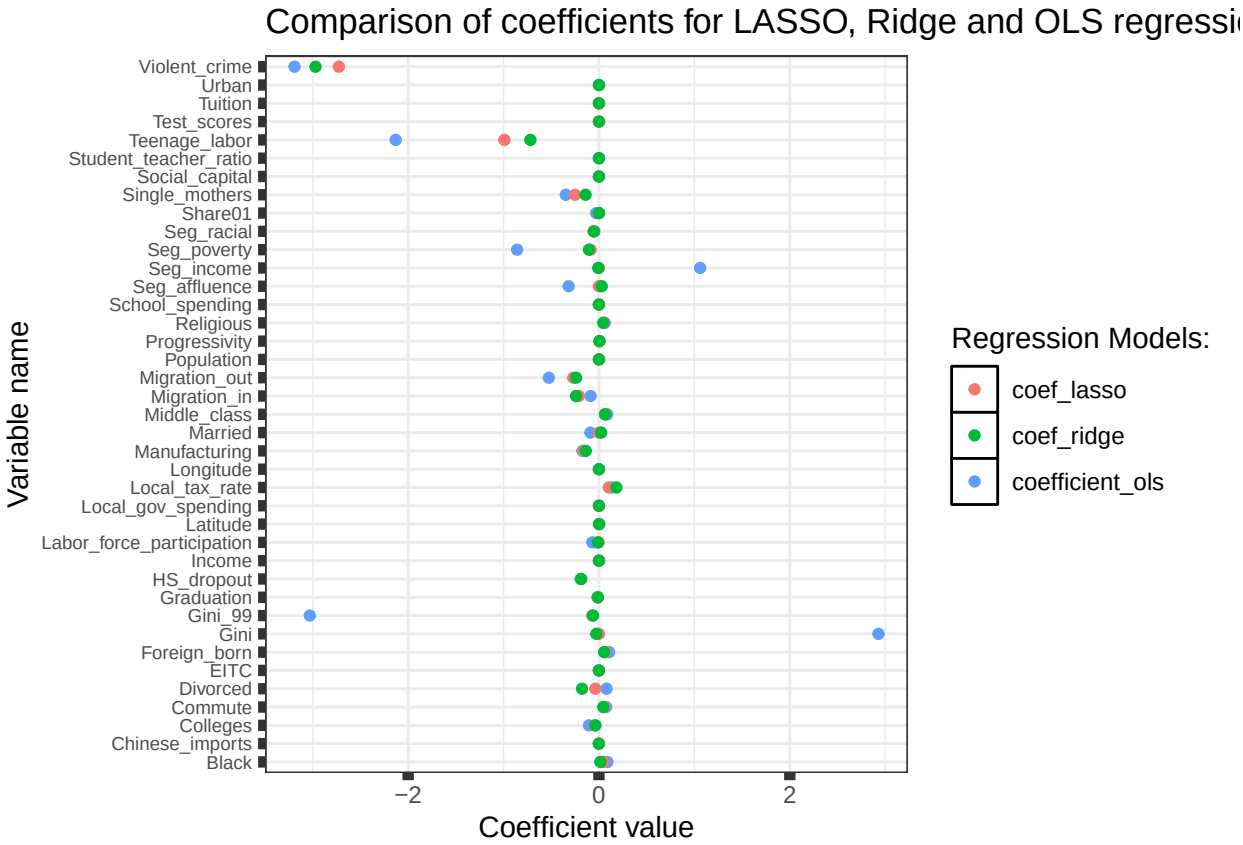



L1 norm on the x-axis means: for each value of $\lambda$ take L1 norm of the corresponding beta coefficient. (Important to remember that for ridge regression, we are minimizing using L2-norm but plotting against the L1-norm).
No, there are no coefficient estimates exactly 0 for any value of the penalty parameter. If any coefficient would be 0, then lambda will converge to infinity.
As $\lambda \to 0$, you move towards the right on the x-axis. (As $\lambda$ decreases, the size of the L1 ball will increase and it will get closer to the OLS-solution.)

5. For both the Ridge regression and Lasso regression in the previous section, choosing the set of coefficient estimates using `lambda.min`. Comparing these coefficient estimates with those in problem 3 by producing a graph.

```
#LASSO regression coefficient estimates using `lambda.min`
best.lam.lasso = lasso$lambda.min
coef.lasso <- coef(lasso, s=best.lam.lasso)

# Ridge regression coefficient estimates using `lambda.min`
```

```r
best.lam.ridge = ridge$lambda.min
coef.ridge <- coef(ridge, s=best.lam.ridge)


# Producing a comparing plot
cf <-coef(summary(ols,complete = TRUE))
x_axis <- c(rownames(cf)[-1])

df_all_coef <- data.frame(
  coefficient_ols=c(signif((summary(ols)$coefficients[2:length(mobility)]), 3)),
                coef_lasso =c(coef(lasso, s=best.lam.lasso)[2:40]),
                coef_ridge =c(coef(ridge, s=best.lam.ridge)[2:40]),
                variableX = c(x_axis))


require(ggplot2)
require(reshape2)
library(tidyverse)
  df_all_coef.long <- df_all_coef %>%
  select("coefficient_ols","coef_lasso", "coef_ridge", "variableX") %>%
  pivot_longer(-variableX, names_to = "variable", values_to = "value")

ggplot(df_all_coef.long, aes(value, variableX, colour = variable)) + geom_point()+
  labs(y = "Variable name",
         x = "Coefficient value",
         color = "Regression Models:")+
  ggtitle("Comparison of coefficients for LASSO, Ridge and OLS regressions")+
  theme(axis.text.y = element_text(angle=0, size =7),
        axis.ticks = element_line(size = 2),
        legend.key = element_rect(fill = "white", colour = "black"))
```




According to the plot above, it seems like the coefficients from lasso and ridge are similar for most of the variables. There are variables like `EITC`, `Local_tax_rate`, `Social_capital`, `Migration_in` that have very different coefficients for lasso and ridge. OLS gives us coefficients that are overall larger than in lasso or ridge. For example, `Gini` in ols has a coefficient value of more than 2, while ridge and lasso coefficients are close to 0. Same happens with `Gini_99`,and `Seg_income`. Variable `Violent_crime` has large negative coefficient in three of the models. Overall, seems like ridge and lasso have much more similar coefficients than ols with any of them. Overall, seems like many coefficients in all three models are close to 0.

6. Calculating the LOOCV score for my OLS model from problem 3. Comparing this score with the scores from problem 4. Selecting 1 of these three models to use for out-of-sample prediction. For the chosen model, making a map of the residuals.

```
set.seed(01101101)

# Calculating leave-one-out CV for OLS model. (function from class)
loocv <- function (mod) {
mean((residuals(mod))^2 /(1-hatvalues(mod))^2)
}
loocv_value = loocv(ols)

lasso
```

```
##
## Call:  cv.glmnet(x = x, y = y)
##
## Measure: Mean-Squared Error
```

```
##
##        Lambda Index   Measure        SE Nonzero
## min 0.0003096    50 0.0005726 6.416e-05      26
## 1se 0.0028872    26 0.0006286 1.001e-04      16
```

```
ridge
```

```
##
## Call:  cv.glmnet(x = x, y = y, alpha = 0)
##
## Measure: Mean-Squared Error
##
##      Lambda Index   Measure        SE Nonzero
## min 0.00516    94 0.0005775 6.826e-05      39
## 1se 0.05286    69 0.0006399 1.057e-04      39
```

```
cat("Leave-one-out CV for OLS model: ", loocv_value, sep="\n")
```

```
## Leave-one-out CV for OLS model:
## 0.0005750473
```

```
cat("Mean-squared error associated with lambda.min in for LASSO is: ",
    0.0005726, sep="\n")
```

```
## Mean-squared error associated with lambda.min in for LASSO is:
## 0.0005726
```

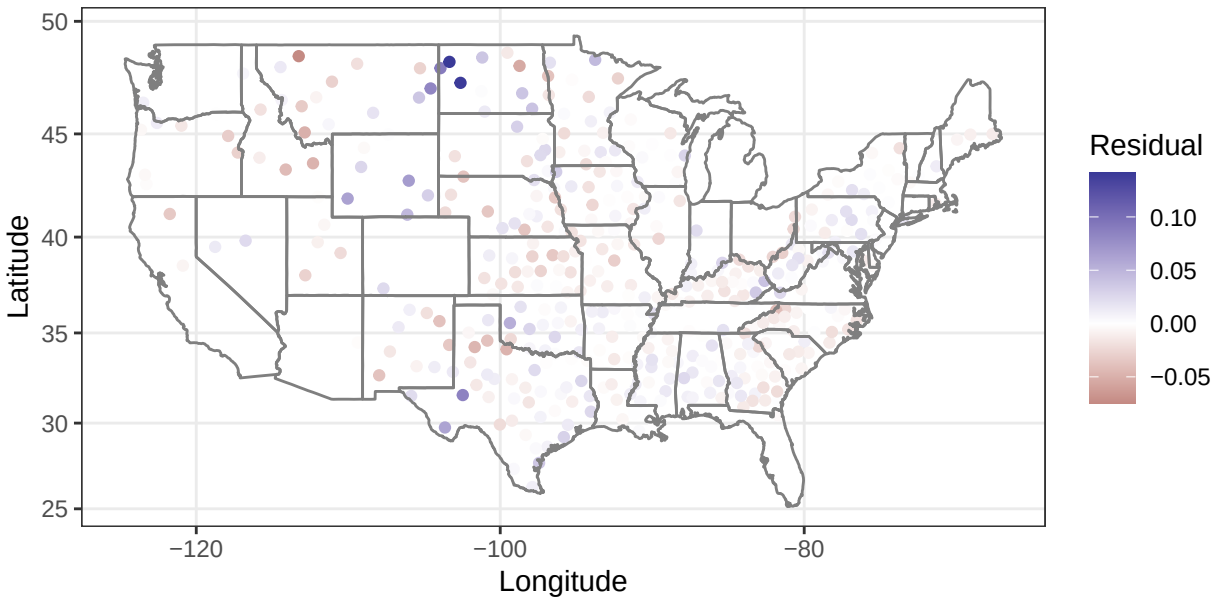
```
cat("Mean-squared error associated with lambda.min in for Ridge is: ",
    0.0005775 , sep="\n")
```

```
## Mean-squared error associated with lambda.min in for Ridge is:
## 0.0005775
```

Comparing the LOOCV value of 0.000575 for the ols model with mse from lasso and ridge, it seems like the LASSO has the smallest CV-error, hence, better estimate of test-error. Therefore, `LASSO` regression should be used for the out-of-sample prediction.

```
#Number of non-zero coefs in lasso is 26 excluding the intercept
#sum(abs(coef(lasso, s="lambda.min")) > 0) -1

# Use best lambda to predict with LASSO
lasso_pred = predict(lasso, s = best.lam.lasso, newx = x)
Residual = y-lasso_pred
x_datframe <- data.frame(x)

#Plotting the map
ggplot(
  x_datframe,
  aes(x=Longitude, y=Latitude, color = Residual)) +
  geom_point() +
```

```
  coord_map() +
  borders("state") +
  scale_color_viridis_c()+
  scale_color_gradient2(mid="white")
```




According to the map above, the majority of absolute residuals values seems to be within 0 and 0.05. There is only one part of the US (border of North Dakota and Montana) where residual values seem to be the largest (around 0.1 or higher). This state also had the highest mobility rate across the US. The one city/community in Montana has the largest negative residual. It seems like there are more negative residuals in the rural Midwest (center of the US). Interestingly, states like Arizona and Michigan seem to have zero residuals (or observations from those states were omitted due to missing entries). Ideally, we want all the residuals to be close to 0.