Lab 4 - Simulation Approaches to Uncertainty

TA: Isabel Laterzo

January 30, 2019

This lab has been adapted from code by Simon Hoellerbauer.

So far in lab, we have often simulated data to allow us to test out various methods. Simulation can help us do much more, as well. Simulations are a useful tool that is being used more and more often within the discipline. They can help us: 1) Construct a hypothetical DGP and then see if the methods we use/develop can successfully retrieve the parameters of that DGP 2) Make our data go further with re-sampling 3) Create plots to help us assess our results when our variance/covariance matrix gets complex (when distributional assumptions can be tricky)

Today, we will use simulation to help us assess the uncertainty around our parameter estimates and will then incorporate this uncertainty into the predicted values of our outcome variable.

```
#clear global environ
rm(list=ls())

#the packages we will need
library(MASS) #required for multivariate normal
library(tidyverse) #we want to load this second in order for its functions
                   #to mask same-name functions from MASS
library(ggplot2)
```

Download the "aid_data_95_ee.csv" in the Labs folder on Sakai and then read it into R:

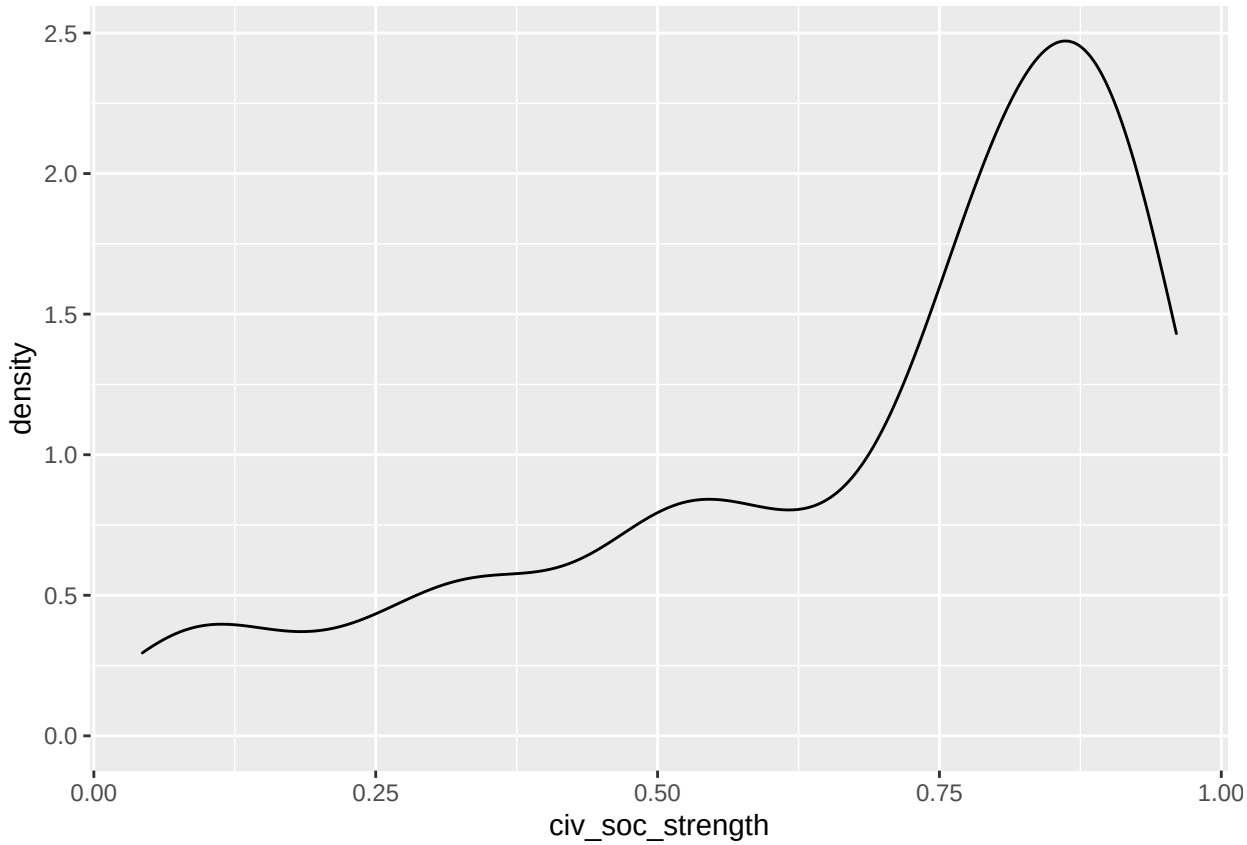
```
#remember to set your working directory!
setwd("/Users/IsabelLaterzo/Dropbox/Poli784_2021/Labs/Lab4_SimulationUncertainty")

#read_csv comes from readr from the tidyverse
aid_data_95_ee <- read_csv("aid_data_95_ee.csv")
```

Let's take a look at this data very quickly:

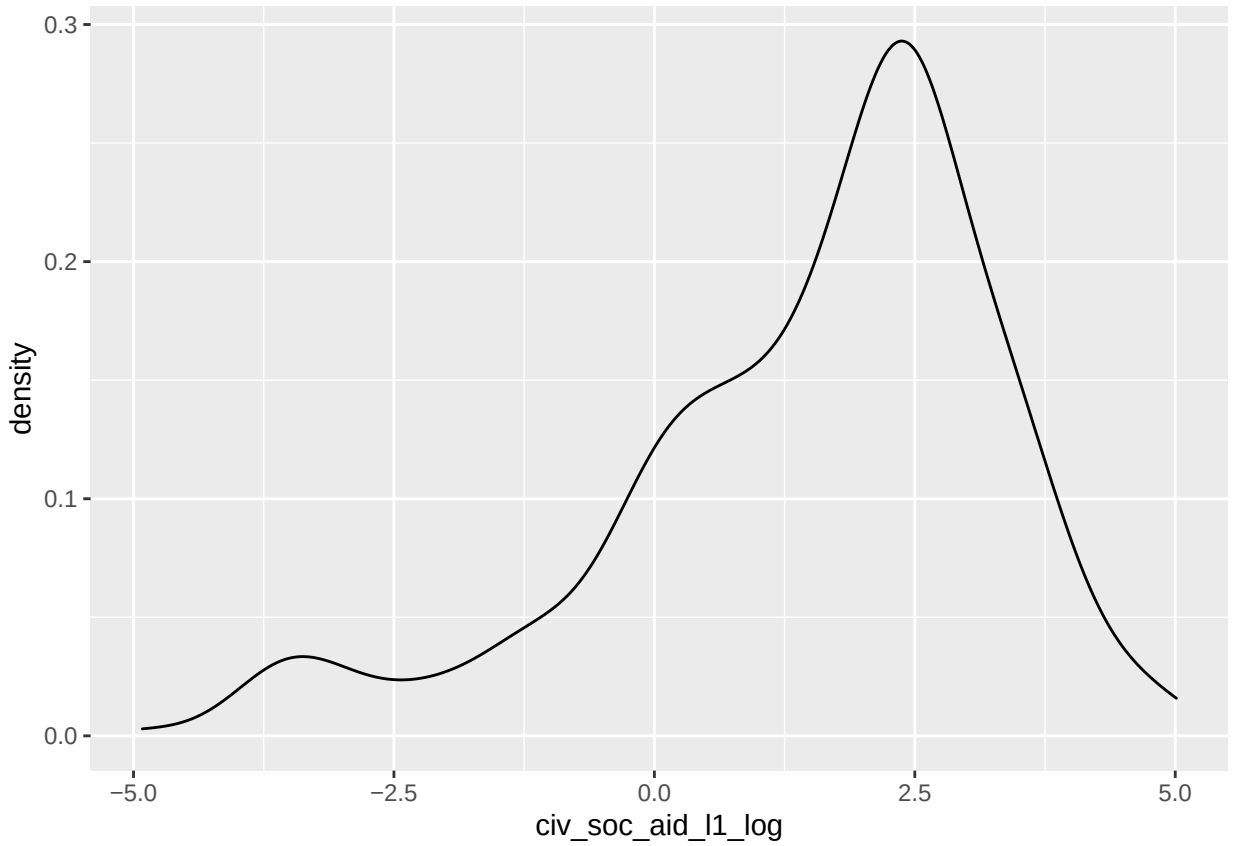
We are going to see how civil society strength in Eastern Europe varies according to lagged civil society assistance, EU accession leverage, and lagged GDP per capita.

Let's look at the distribution of the outcome variable (with some simple, barebones graphs):

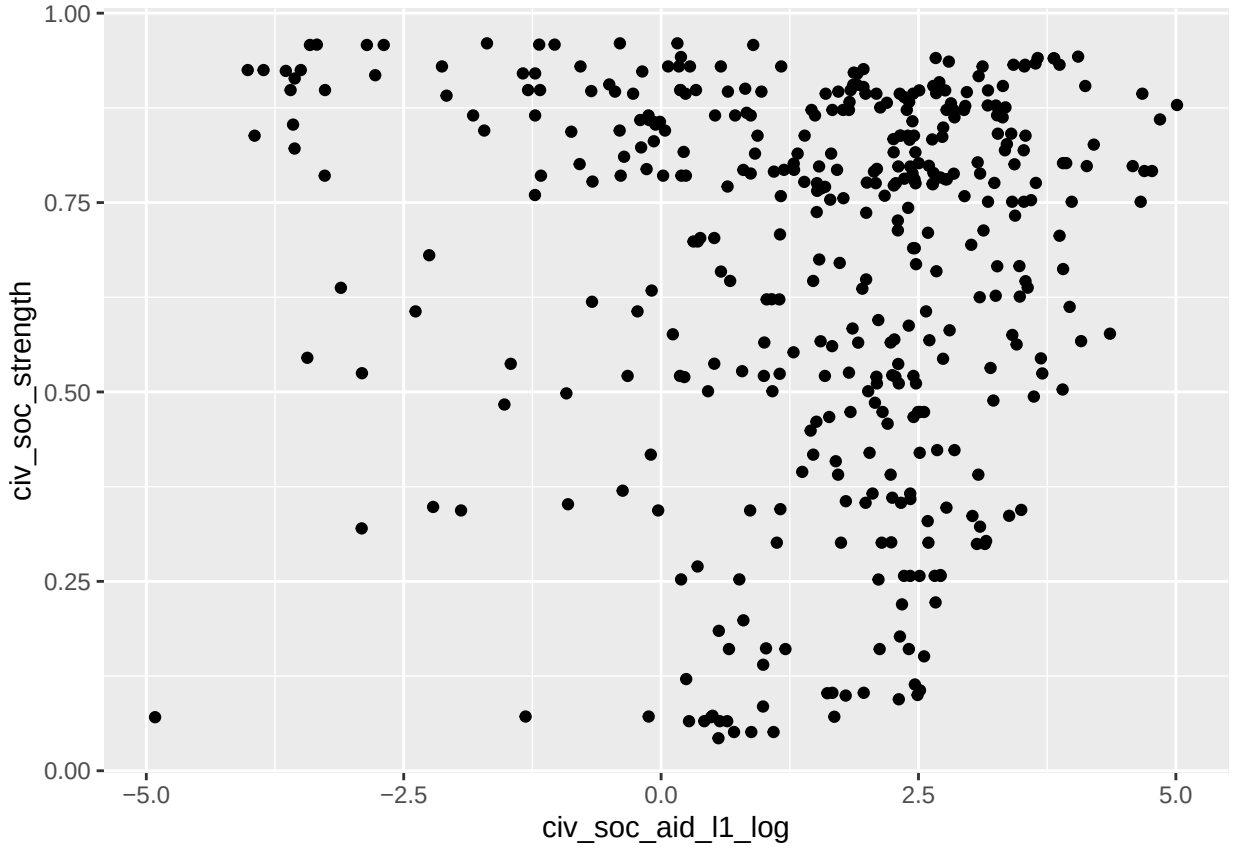
```
#distribution of civil society strength
ggplot(data = aid_data_95_ee, aes(x = civ_soc_strength)) +
  geom_density()
```



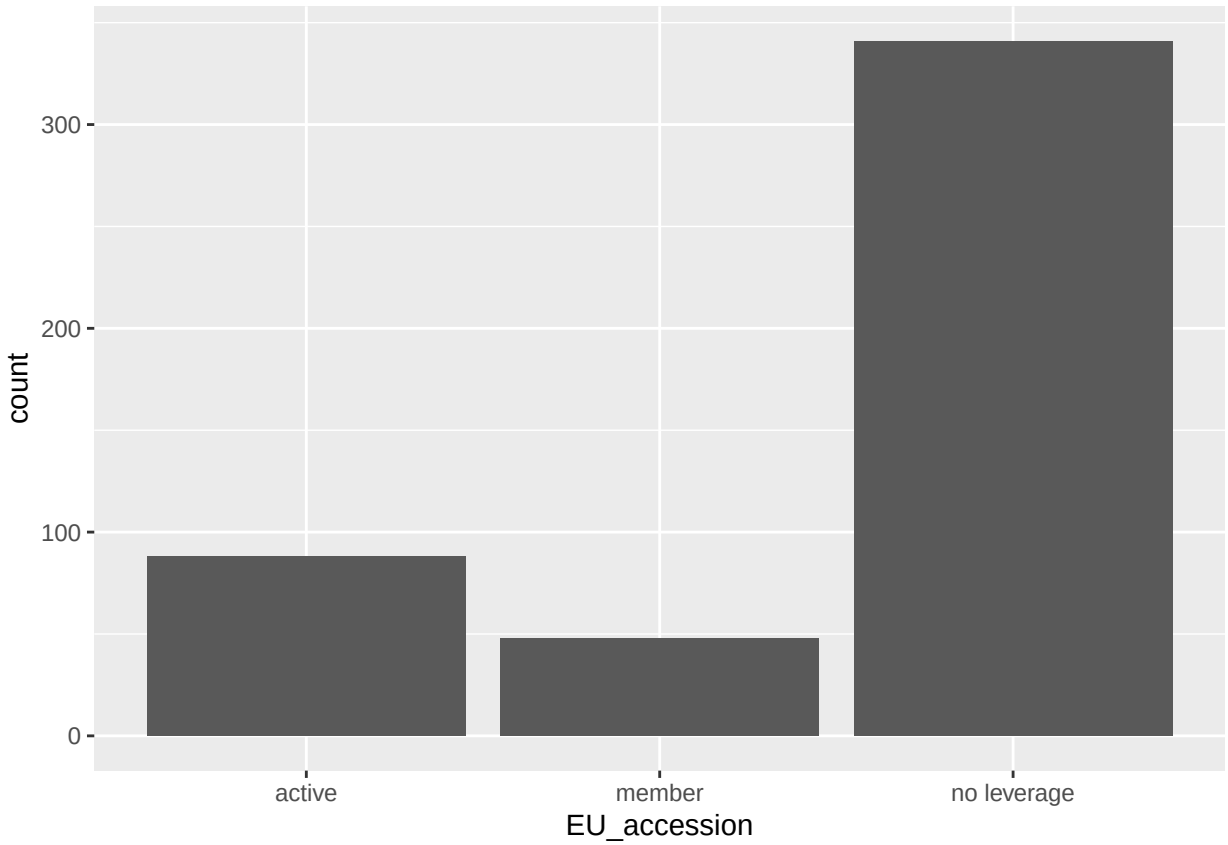

```
#Is it necessarily a problem that our outcome variable is skewed?

#what does our main predictor of interest look like?
ggplot(data = aid_data_95_ee, aes(x = civ_soc_aid_l1_log)) +
  geom_density()
```




```
#why do you think we log civil society aid?

#let's look at relationship between the two
ggplot(data = aid_data_95_ee, aes(x = civ_soc_aid_l1_log, y = civ_soc_strength)) +
  geom_point()
```




```
#what about EU accession?
ggplot(data = aid_data_95_ee, aes(x = EU_accession)) + geom_bar()
```



Let's fit our model:

```
#first we have to make sure that EU_accession is a factor, and that the
#reference category makes sense for us
aid_data_95_ee$EU_accession <- factor(aid_data_95_ee$EU_accession) %>%
  relevel(ref = "no leverage")
levels(aid_data_95_ee$EU_accession)

#Estimating the model
cs_model <- lm(civ_soc_strength ~ GDPpc_ppp_WB_2011_l1_log +
                 EU_accession*civ_soc_aid_l1_log, data = aid_data_95_ee)
summary(cs_model)
```

Now simulation comes in! We can use simulation to get confidence intervals. Here, we will be using the function `mvrnorm` to simulate some data. Look at the help file for that function and what it does.

```
#set seed for replication
set.seed(53)

#We will take 500 samples from the sampling distribution
#of our model's coefficients
samp_beta <- mvrnorm(500,
                     coef(cs_model),
                     vcov(cs_model))
dim(vcov(cs_model))
```

```
## [1] 7 7
```

```
dim(samp_beta)
```

```
## [1] 500   7
```

```
# 87% Monte Carlo/simulation based confidence interval:
t(apply(samp_beta,
        2, #why 2? look at help file and the "MARGIN" argument
        quantile,
        probs = c(0.065, 0.935)))
```

```
##                                                 6.5%       93.5%
## (Intercept)                                0.004934925  0.44644760
## GDPpc_ppp_WB_2011_l1_log                   0.008049019  0.05962058
## EU_accessionactive                         0.289361296  0.39967224
## EU_accessionmember                         0.223037506  0.45232882
## civ_soc_aid_l1_log                         0.022982351  0.04812884
## EU_accessionactive:civ_soc_aid_l1_log -0.063520254 -0.01745147
## EU_accessionmember:civ_soc_aid_l1_log -0.081337783  0.01239318
```

Let's compare that to CIs generated from the function `confint()`. Both produce asymptotic confidence intervals, but the first uses resampling/Monte Carlo methods. I suggest using these methods, as it is easier for us to get CIs for slightly more complicated models via simulation.

```
confint(cs_model, level = 0.87)
```

```
##                                                  6.5 %      93.5 %
## (Intercept)                                -0.006319174  0.44363131
## GDPpc_ppp_WB_2011_l1_log                    0.008633669  0.06090779
## EU_accessionactive                          0.284496766  0.39471026
## EU_accessionmember                          0.223760575  0.44972868
## civ_soc_aid_l1_log                          0.023451120  0.04710138
## EU_accessionactive:civ_soc_aid_l1_log -0.065353020 -0.01862332
## EU_accessionmember:civ_soc_aid_l1_log -0.081477035  0.01561493
```

We are now going to use the "average value approach" to simulating predicted outcomes and uncertainty around them. We want to see how predicted civil society strength varies according to lagged civil society assistance. In order to do this, there are a few thing we have to do first:

```
# we are going to let lagged log civil society assistance vary along its
# inner-quartile range. These will be the values over which we plot the predicted
# civil society strength
cs_aid_sim <- with(aid_data_95_ee,
                   #quantile() here produces sample quantities based on the given
                   #probability we provide it, here 0.25
                   seq(quantile(civ_soc_aid_l1_log, 0.25, na.rm = T),
                       quantile(civ_soc_aid_l1_log, 0.75, na.rm = T),
                       length.out = 100))

# we now need to create a matrix of hypothetical predictors. We will use the
# mean of lagged log GDP per capita; in order to see the effect of EU accession
# leverage, we will also allow that to vary (otherwise, you usually use the mode
# of categorical variables)
x_mat_sim <- with(aid_data_95_ee,
                  model.matrix(~ GDPpc_ppp_WB_2011_l1_log +
                                 EU_accession*civ_soc_aid_l1_log,

                               data = data.frame(GDPpc_ppp_WB_2011_l1_log =
```

```
                                        mean(GDPpc_ppp_WB_2011_l1_log,
                                             na.rm = T), #mean of lagged log GDP

                                      EU_accession = relevel(
                                        factor(c("no leverage",
                                                 "active", # vary EU accession
                                                 "member")),
                                        ref = "no leverage"),

                                      #calling above data for civ soc assist
                                      civ_soc_aid_l1_log = rep(cs_aid_sim,
                                                               each = 3))))


#let's check dimensions
dim(x_mat_sim)

#how does it look? note the interaction effects included!
head(x_mat_sim)
```

Now we have to do XBeta to get our predicted values! But instead of beta being a k x 1 vector, it is now a k x 500 (number of samples from our multivariate normal) matrix. So, what will the dimensions of this new object be?

```
cs_str_hat <- x_mat_sim %*% t(samp_beta)

dim(cs_str_hat)
```

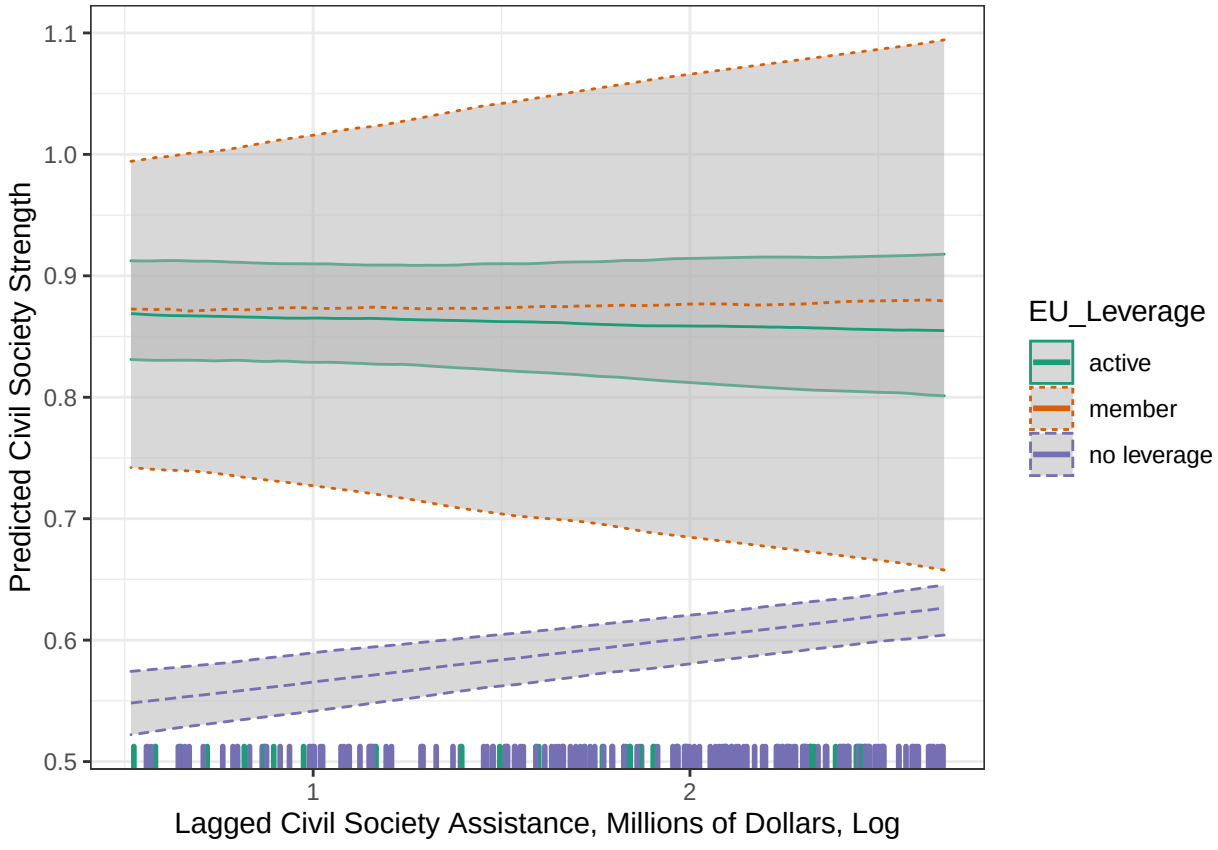
```
## [1] 300 500
```

```
#we only need quantiles for our confidence intervals
cs_str_hat_quant <- apply(cs_str_hat, 1, #why 1? again look at MARGIN
                          quantile, c(0.065, 0.5, 0.935))

#creating data frame of predictions
pred_df <- data.frame(PCS = cs_str_hat_quant[2, ], #why calling 2, 3, and 1?
                      UB = cs_str_hat_quant[3, ],
                      LB = cs_str_hat_quant[1, ],
                      cs_aid = x_mat_sim[, "civ_soc_aid_l1_log"],
                      EU_Leverage = rep(c("no leverage", "active", "member"),
                                        100)) #why can we do this?

#for rug plot
cs_aid_IQ <- dplyr::select(aid_data_95_ee, civ_soc_aid_l1_log, EU_accession) %>%
  filter(civ_soc_aid_l1_log > quantile(civ_soc_aid_l1_log, 0.25, na.rm = T) &
           civ_soc_aid_l1_log < quantile(civ_soc_aid_l1_log, 0.75, na.rm = T)) %>%
  rbind(data.frame(civ_soc_aid_l1_log = rep(NA, nrow(pred_df) - nrow(.)),
                   EU_accession = rep(NA, nrow(pred_df) - nrow(.))))
```

Now let's graph this!

```
#a library with different colors for ggplot2
library("RColorBrewer")


ggplot(data = pred_df, aes(x = cs_aid, y = PCS, lty = EU_Leverage)) +
```

```
geom_ribbon(aes(ymin = LB, ymax = UB, color = EU_Leverage),
            alpha = 0.5,
            fill = "gray70") +

geom_line(aes(color = EU_Leverage)) +
#what do you think this does?

geom_rug(data = cs_aid_IQ, aes(x = civ_soc_aid_l1_log, color = EU_accession),
         size = 1, inherit.aes = F) +
#rug plot

scale_color_brewer(palette = "Dark2") +
#if you want to have some fun, change around the colors
#scale_color_brewer(palette = "Set3") +
#scale_color_brewer(palette = "Greys") +
#scale_color_brewer(palette = "PuOr") +
#scale_color_brewer(palette = "Set1") +

xlab("Lagged Civil Society Assistance, Millions of Dollars, Log") +
ylab("Predicted Civil Society Strength") +
#labels

theme_bw()
```




```
#theme
```

To do on your own:

Take the `Guns` dataset in the `AER` package. Take a look at the dataset and use `?Guns` to get the codebook. Regress the robbery rate on the percent of the state population that is male, the real per capita personal income in the state, and then an interaction between the population density and whether a shall carry law was in effect that year. Use Monte Carlo simulation to get 80% confidence intervals around the coefficient estimates. Then, use simulation to plot the predicted robbery rate as the population density varies from its 10th percentile to its 90th percentile (don't worry about the rug plot).

```
library(AER)
data(Guns)
```