POLS 207

Problem Set 5*

Villaseñor-Derbez, J.C.

Problem 1: 2SLS

a) Show results from (1) the first stage regression, (2) the reduced form regression, and (3) the 2SLS estimation using the following two specifications:

$$log(PGP_i^{1995}) = \beta_0 + \beta_1 avexpr_i + \epsilon_i$$

$$avexpr_i = \gamma_0 + \gamma_1 logem4 + \mu_i$$

And

$$log(PGP_i^{1995}) = \beta_0 + \beta_1 avexpr_i + \beta_2 latabst + \epsilon_i$$

$$avexpr_i = \gamma_0 + \gamma_1 logem4 + \gamma_2 latabst + \mu_i$$

```
# Load packages
suppressPackageStartupMessages({
  library(startR)
  library(here)
  library(countrycode)
  library(rnaturalearth)
  library(stargazer)
  library(foreign)
  library(AER)
  library(magrittr)
  library(tidyverse)
})
```

```
# Load the data
arj <- read.dta(file = here("ps5", "arj.dta"))
```

```
# First stage
first_simple <- lm(avexpr ~ logem4, data = arj)
first_lat <- lm(avexpr ~ logem4 + lat_abst, data = arj)

# Reduced form
reduced_simple <- lm(logpgp95 ~ logem4, data = arj)
reduced_lat <- lm(logpgp95 ~ logem4 + lat_abst, data = arj)

# Two-stage using IVreg
two_stage_simple <- ivreg(logpgp95 ~ avexpr | logem4, data = arj)
two_stage_lat <- ivreg(logpgp95 ~ avexpr + lat_abst | logem4 + lat_abst, data = arj)
```

*Available on GitHub: https://github.com/jcvdav/POLS207/blob/master/ps5/

```r
stargazer(list(first_simple, first_lat),
          single.row = T,
          header = F,
          title = "Coefficients for first-stage regression.")
```

Table 1: Coefficients for first-stage regression.

	Dependent variable:	
	avexpr	
	(1)	(2)
logem4	-0.607^{***} (0.127)	-0.510^{***} (0.141)
lat_abst		2.002 (1.337)
Constant	9.341^{***} (0.611)	8.529^{***} (0.812)
Observations	64	64
R^2	0.270	0.296
Adjusted R^2	0.258	0.273
Residual Std. Error	1.265 (df = 62)	1.252 (df = 61)
F Statistic	22.947^{***} (df = 1; 62)	12.824^{***} (df = 2; 61)
Note:		*p<0.1; **p<0.05; ***p<0.01

```r
stargazer(list(reduced_simple, reduced_lat),
          single.row = T,
          header = F,
          title = "Coefficients for reduced form regression.")
```

Table 2: Coefficients for reduced form regression.

	Dependent variable:	
	logpgp95	
	(1)	(2)
logem4	-0.573^{***} (0.076)	-0.508^{***} (0.084)
lat_abst		1.346^{*} (0.800)
Constant	10.731^{***} (0.367)	10.185^{***} (0.486)
Observations	64	64
R^2	0.477	0.500
Adjusted R^2	0.469	0.484
Residual Std. Error	0.760 (df = 62)	0.749 (df = 61)
F Statistic	56.603^{***} (df = 1; 62)	30.551^{***} (df = 2; 61)
Note:		*p<0.1; **p<0.05; ***p<0.01

```r
stargazer(list(two_stage_simple, two_stage_lat),
          single.row = T,
          header = F,
          title = "Coefficients for 2SLS.")
```

Table 3: Coefficients for 2SLS.

	Dependent variable:	
	logpgp95	
	(1)	(2)
avexpr	0.944*** (0.157)	0.996*** (0.222)
lat_abst		−0.647 (1.335)
Constant	1.910* (1.027)	1.692 (1.293)
Observations	64	64
R^2	0.187	0.102
Adjusted R^2	0.174	0.073
Residual Std. Error	0.948 (df = 62)	1.005 (df = 61)
Note:		*p<0.1; **p<0.05; ***p<0.01

Table 4: Two-stage least squares regression table

		(1) no covariates	(2) including latitude
First stage (dep: `avexpr`):	`logem4`	-0.6067782	-0.5102681
	`lat_abst`		2.0017746
Reduced form (dep: `logpgp95`):	`logem4`	-0.5729682	-0.508076
	`lat_abst`		1.3459679
2SLS (dep: `logpgp95`):	`avexpr`	0.9442794	0.995704
	`lat_abst`		-0.6558067

Regress `logpgp95`, `avexpr`, and `logem4` on `lat_abst` ("partialling out" the effect of latitude) and re-do the 2SLS estimation using the residuals. Do you get the same result as in Column 2 in the previous question? (Don't worry about the standard errors – actually they are quite close to the right ones.)

The regression table below shows the OLS estimates for each stage, as well as the IV regression. Using the residuals after removing the effect of latitude, we obtain the same coefficients as the second column in the previous question.

```r
res_logpgp95 <- lm(logpgp95 ~ lat_abst, data = arj)$residuals
res_avexpr <- lm(avexpr ~ lat_abst, data = arj)$residuals
res_logem4 <- lm(logem4 ~ lat_abst, data = arj)$residuals
```

```r
# First stage
res_first_simple <- lm(res_avexpr ~ res_logem4)

# Reduced form
res_reduced_simple <- lm(res_logpgp95 ~ res_logem4)

# Two-stage using IVreg
res_two_stage_simple <- ivreg(res_logpgp95 ~ res_avexpr | res_logem4)
```

```r
stargazer(... = list(res_first_simple, res_reduced_simple, res_two_stage_simple),
          single.row = T,
          header = F,
          title = "Two-stage regression on the residuals of each variable on latabst.")
```

Table 5: Two-stage regression on the residuals of each variable on latabst.

	Dependent variable:		
	res_avexpr	res_logpgp95	
	OLS	*OLS*	*instrumental variable*
	(1)	(2)	(3)
res_logem4	-0.510^{***} (0.140)	-0.508^{***} (0.084)	
res_avexpr			0.996^{***} (0.220)
Constant	-0.000 (0.155)	0.000 (0.093)	0.000 (0.125)
Observations	64	64	64
R^2	0.177	0.373	-0.127
Adjusted R^2	0.163	0.363	-0.145
Residual Std. Error (df = 62)	1.242	0.743	0.996
F Statistic (df = 1; 62)	13.308^{***}	36.838^{***}	
Note:			*p<0.1; **p<0.05; ***p<0.01

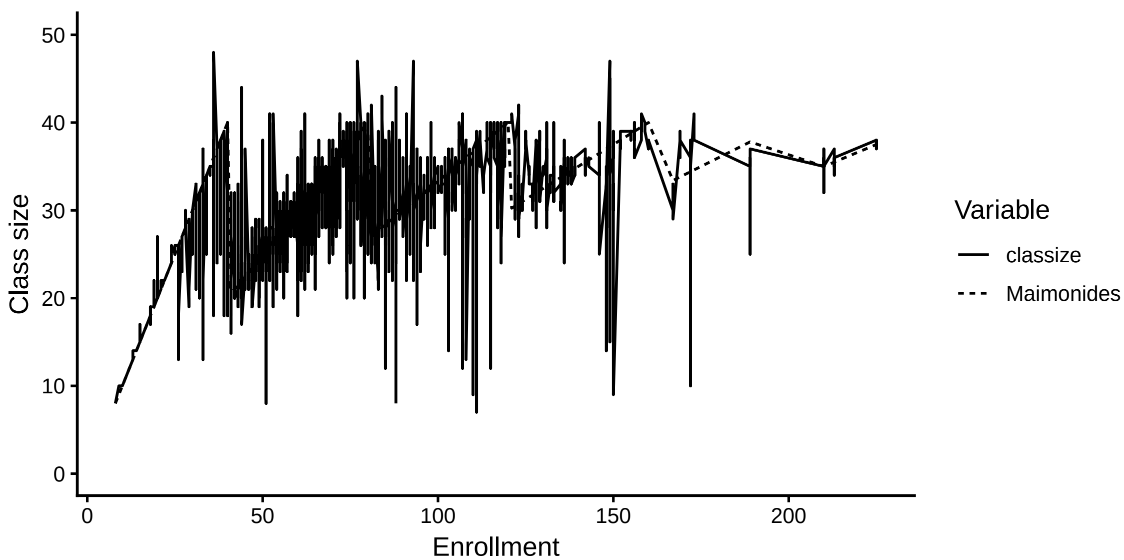
Problem 2: Fuzzy IV

```r
# Load data
angrist_lavy <- read.dta(here("ps5", "angrist_lavy.dta"))
```

a) Say you were to run a regression of reading scores on class sizes. Would this provide a valid estimate of the causal effect of class size? Why or why not?

b) Use the data and plot the actual average class size (solid line) and the class size implied by Maimonides rule (dashed line) against enrollment count. That is, replicate Figure 1 of the paper for the fourth grade. What do the results imply about the determinants of class size? (you may find the `floor()` function useful).

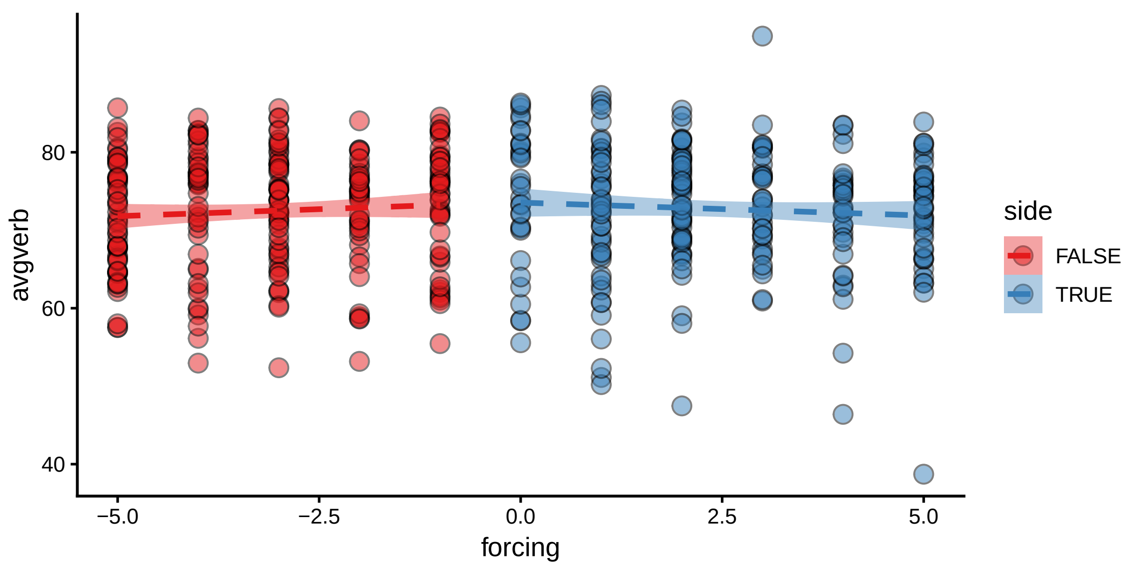
```r
angrist_lavy %>%
  mutate(Maimonides = angrist_lavy$enrollment / (floor((angrist_lavy$enrollment - 1) / 40) + 1)) %>%
  select(enrollment, classize, Maimonides) %>%
  gather(variable, value, -enrollment) %>%
  ggplot(aes(x = enrollment, y = value, linetype = variable)) +
  geom_line() +
  ggtheme_plot() +
  lims(y = c(0, 50)) +
  labs(x = "Enrollment", y = "Class size") +
  guides(linetype = guide_legend(title = "Variable"))
```



```r
# Define discontinuity points to keep
disc_vec <- c(36:46, 76:86, 116:126, 156:166, 196:206)

# Define forcing variable
angrist_lavy_disc <- angrist_lavy %>%
  filter(enrollment %in% disc_vec) %>%
  rowwise() %>%
  mutate(disc_point = case_when(between(enrollment, 36, 46) ~ 41,
                               between(enrollment, 76, 86) ~ 81,
                               between(enrollment, 116, 126) ~ 121,
                               between(enrollment, 156, 166) ~ 161,
                               T ~ 201),
         forcing = enrollment - disc_point,
         side = forcing >= 0)
```

```r
ggplot(data = angrist_lavy_disc,
       mapping = aes(x = forcing, y = avgverb, fill = side)) +
  geom_point(size = 3, shape = 21, alpha = 0.5) +
  geom_smooth(method = "lm", linetype = "dashed", aes(color = side)) +
  ggtheme_plot() +
  scale_fill_brewer(palette = "Set1") +
  scale_color_brewer(palette = "Set1")
```




```r
# Effect of discontinuities on class size
class_model <- lm(classize ~ forcing, data = angrist_lavy_disc)

# Effect of disc on reading comprehension scores
score_model <- lm(avgverb ~ forcing + side, data = angrist_lavy_disc)
```

```r
stargazer(list(class_model, score_model),
          single.row = T,
          header = F,
          title = "Effect of the discontinuities on class size and average reading comprehension scores
```

Table 6: Effect of the discontinuities on class size and average reading comprehension scores.

	Dependent variable:	
	classize	avgverb
	(1)	(2)
forcing	-1.101^{***} (0.092)	-0.038 (0.228)
side		0.449 (1.446)
Constant	30.920^{***} (0.292)	72.377^{***} (0.862)
Observations	482	482
R^2	0.231	0.0003
Adjusted R^2	0.229	-0.004
Residual Std. Error	6.381 (df = 480)	7.726 (df = 479)
F Statistic	143.826^{***} (df = 1; 480)	0.071 (df = 2; 479)
Note:		*p<0.1; **p<0.05; ***p<0.01

Probelm 3: Bootstrapping

Estimate this difference in means in the sample above, and then calculate the precise standard error associated with this estimate.

```
# Set a random seed
set.seed(43)

# Generate vectors of random variables
vector1 <- rnorm(500, mean = 7, sd = 3)
vector2 <- rnorm(500, mean = 5, sd = 2)

# Calculate difference in means and print
(diff_mean <- mean(vector1) - mean(vector2))
```

```
## [1] 1.919017
```

$$SE = \sqrt{\frac{\sigma^2_{Y1}}{N_1} + \frac{\sigma^2_{Y2}}{N_2}}$$

```
# Get variances
sigma1 <- var(vector1)
sigma2 <- var(vector2)

# Get standard error and print
(SE <- sqrt((sigma1 / 500) + (sigma2 / 500)))
```

```
## [1] 0.1596008
```

Now write code to calculate the standard error associated with the difference-in-means estimate by bootstrapping. Your code should 1) sample from your vectors 2) calculate the difference-inmeans associated with that sample, 3) repeat 10 000 times, 4) take the standard deviation of the resulting sampling distribution.

```
boot <- function(vec1, vec2, rep) {
  n <- length(vec1)
  mean_vec <- numeric(length = n)

  for (i in 1:rep) {

    vec1_i <- sample(x = vec1, size = n, replace = T)
    vec2_i <- sample(x = vec2, size = n, replace = T)

    mean_i <- mean(vec1_i) - mean(vec2_i)

    mean_vec[i] <- mean_i
  }

  return(mean_vec)
}
```

```r
set.seed(43)
bootstrapped_diff_means <- boot(vec1 = vector1,
                                vec2 = vector2,
                                rep = 10000)
sd(bootstrapped_diff_means)
```

```
## [1] 0.1609404
```

Problem 4: Effective Samples

```
jensen <- read.dta(here("ps5", "jensen-rep.dta"))
```

a) How many countries are included in the dataset? How many of these countries have complete data on all covariates?

```
# Number of countries
length(unique(jensen$country))
```

```
## [1] 114
```

```
# Countries without any NAs
jensen %>%
  drop_na() %$%
  unique(country) %>%
  length()
```

```
## [1] 114
```

b) Now run a regression with `Fvar5` as your DV, and including `regime`, `market`, `lgdppc`, `gdpgrowt`, `tradeofg`, `overallb`, `generalg`, `country`, `d2` and `d3` as controls. Interpret these results using standard multivariate regression logic.

```
Fvar5_model <- lm(Fvar5 ~ regime + market + lgdppc + gdpgrowt + tradeofg + overallb + generalg + d2 + d3
                  data = jensen)
```

```
stargazer(Fvar5_model,
          se = estimatr::starprep(Fvar5_model, se_type = "HC2"),
          single.row = T,
          header = F,
          t.auto = T,
          p.auto = T,
          omit = "country")
```

Table 7:

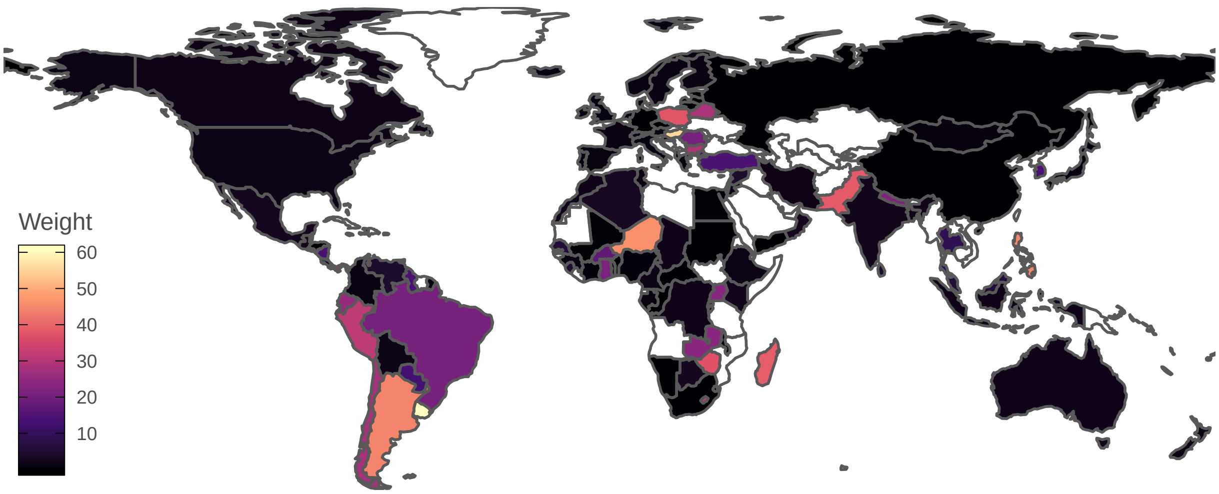
	Dependent variable:
	Fvar5
regime	0.027
market	−1.264
lgdppc	1.875
gdpgrowt	0.034
tradeofg	0.015
overallb	−0.034
generalg	−0.062
d2	−0.159
d3	0.325
Constant	16.459
Observations	1,630
R^2	0.554
Adjusted R^2	0.518
Residual Std. Error	1.375 (df = 1507)
F Statistic	15.341*** (df = 122; 1507)
Note:	*p<0.1; **p<0.05; ***p<0.01

c) Now run a regression where `regime` is your DV on the remainder of the controls in part (b). Save the residuals from this regression and square them. Calculate the mean value across each residual for each country. These weights tell you the relative contribution of each unit to the effective sample. Reinterpret the results from part (b), now in terms of a Local Average Treatment Effect.

```
# Calculate the regime model
regime_model <- lm(regime ~ + market + lgdppc + gdpgrowt + tradeofg + overallb + generalg + d2 + d3 + co
                   data = jensen)

# Get mean squared residuals by country
country_weights <- jensen %>%
  mutate(residuals = residuals(regime_model) ^ 2,
         country = countrycode(sourcevar = country,
                               origin = "country.name",
                               destination = "iso3c")) %>%
  group_by(country) %>%
  summarize(weight = mean(residuals)) %>%
  ungroup()

# Map mean squared residuals
ne_countries(scale = "small", type = "countries", returnclass = "sf") %>%
  select(iso_a3) %>%
  filter(!iso_a3 == "ATA") %>% #Remove antartica for a nicer map
  left_join(country_weights, by = c("iso_a3" = "country")) %>%
  ggplot(aes(fill = weight)) +
  geom_sf() +
  scale_fill_viridis_c(option = "A", na.value = "transparent") +
  ggtheme_map() +
  scale_y_continuous(expand = c(0,0)) +
  scale_x_continuous(expand = c(0,0)) +
  guides(fill = guide_colorbar(title = "Weight",
                               ticks.colour = "black",
                               frame.colour = "black")) +
  theme(legend.justification = c(0, 0),
        legend.position = c(0, 0))
```


Weight
60
50
40
30
20
10