Replication

Dora Simon and Julian Langer

September 10, 2018

Please note that I have always specified `echo=TRUE` in the chunk options, so that you can see the code. In a real document, you want that option to be false.

Loading Libraries

Here, I load all libraries necessary for the project. I set `message = FALSE`, otherwise the document will be spammed by messages from the libraries.

```
library(AER)       # ivreg command
library(ivpack)    # robust and clustered standard errors
library(dplyr)     # data manipulation
library(ggplot2)   # graphs
library(tibble)    # nice dataframes
library(haven)     # dta files
library(stargazer)# tables
library(sandwich) # robust se
library(lmtest)    # for print robust
```

Exercise 1 : Graph

Read in Data

1. Read in `maketable2.dta` from the tidy-data folder

- Create factor variables out of `africa`, `asia`and `other`
- Only keep the base sample (`baseco==1`)

```
ajr_base <- read_dta("./tidy-data/maketable2.dta") %>%
  mutate(africa = factor(africa),
         asia = factor(asia),
         other = factor(other))  %>%
  filter(baseco == 1)
```

Replicate Figure 2

1. Initiate the plot

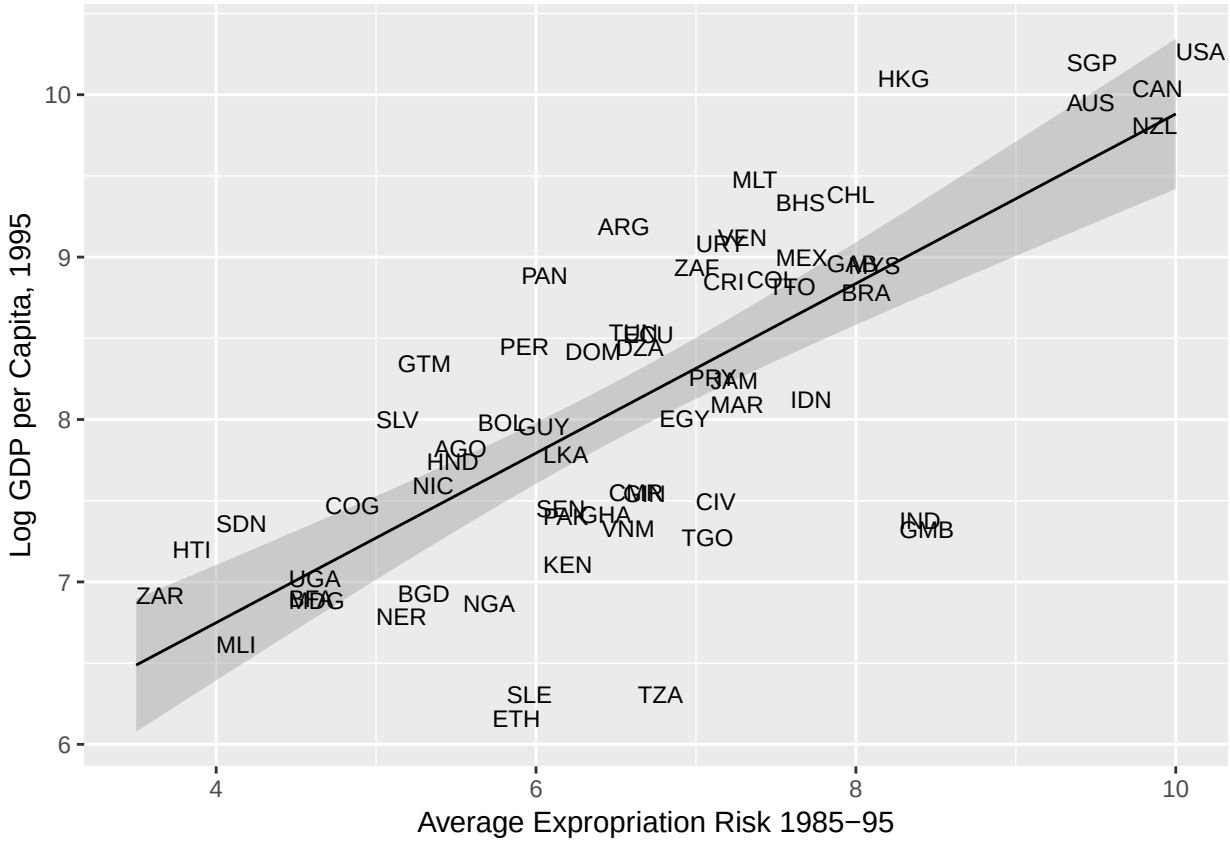
- Average Expropriation Risk is the x axis
- Log GDP per Capita is the y axis

2. Adjust the axis labels
3. Put a regression line in the graph

- you can adjust the color and the size

4. Add the data points as country names

```
ggplot(ajr_base, aes(x = avexpr, y = logpgp95)) +
  labs(x = "Average Expropriation Risk 1985-95",
       y = "Log GDP per Capita, 1995") +
  geom_smooth(method = "lm", color = "Black", size = 0.5) +
  geom_text(aes(label = shortnam), hjust = 0, vjust = 0, size = 3)
```



Exercise 2 : IV Regression

Read in data

1. Read in `maketable4.dta` from the tidy-data folder

- Create factor variables out of `rich4`
- Keep only the base sample (`baseco==1`)

```
ajr_base <- read_dta("./tidy-data/maketable4.dta") %>%
  mutate(rich4 = factor(rich4)) %>%
  filter(baseco == 1)
```

Regressions

Do the first two regressions using the `ivreg` command.

```
iv_1 <- ivreg(logpgp95 ~ avexpr | logem4, data =ajr_base)
iv_2 <- ivreg(logpgp95 ~ avexpr + lat_abst | logem4  + lat_abst, data =ajr_base)
```

Standard Errors

Compute the robust standard errors using the `robust.se` function. Your standard errors will be in the second column.

```
iv_1_se <- robust.se(iv_1)[,2]
```

```
## [1] "Robust Standard Errors"
```

```
iv_2_se <- robust.se(iv_2)[,2]
```

```
## [1] "Robust Standard Errors"
```

Table

Display the first two regressions as a table. (hint: specify the chunk option `results=asis`) - for viewing the table in .Rmd, specify `type="text"` - for a nice output in the .pdf, specify `type="latex"`

```
stargazer(iv_1, iv_2,
          se = list(iv_1_se, iv_2_se),
          dep.var.labels = "Log GDP in 1995",
          covariate.labels = c("Expropriation Risk", "Latitude", "Constant"),
          omit.stat = "ser",
          title = "Instrumental Variable Results",
          type = "latex")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Mo, Sep 10, 2018 - 13:36:01

Table 1: Instrumental Variable Results

	Dependent variable:	
	Log GDP in 1995	
	(1)	(2)
Expropriation Risk	0.944***	0.996***
	(0.176)	(0.240)
Latitude		−0.647
		(1.227)
Constant	1.910	1.692
	(1.174)	(1.448)
Observations	64	64
R^2	0.187	0.102
Adjusted R^2	0.174	0.073
Note:	*p<0.1; **p<0.05; ***p<0.01	

Additionally: Referencing variables

I can also reference variables. For instance, the coefficient on avexpr has the value 0.9442794.