Gravity in R: a short workshop

Krisna Gupta

May 22, 2024

This page is dedicated to teach students on running the gravity model in R. We uses dataset from CEPII, specifically BACI and the gravity dataset. We run in R and RStudio IDE, and we rely on ppml from the gravity package when demonstrating PPML. We try to replicate Silva and Tenreyro (2006), an excellent paper for an introduction to the log gravity model and PPML.

For the latest version click here.

Introduction

The gravity model is probably the most popular model in international trade. Many uses them. It is very intuitive, great predictive power, and most importantly, tweakable (Yotov 2022). But the even most important is that UI students love them. If you're doing trade for your thesis, then you probably going to use the gravity model as your backbone.

This guide is my attempt to help you learn gravity model much easier. The most important part is probably the data and the model itself. What is the minimum things you need in the gravity model, how to arrange the database, run them, and interpret them. You must familiarize yourself with the data and its wrangling (80% of your coding) as well as the main gravity specification to date. I encourage students to pay careful attention to Yotov (2022) as it hosts the recent development in the gravity model, a must read if you're planning to utilize gravity model.

I use R here because I use R much more than Stata these days. However, the two language aren't very different. You can do the same thing on both, but you may need to google a bit. It's okay to use google a lot. I did as well even right now. Oh yeah I also informed you guys know R already so I won't go into too much basic stuff.

Next is the preparation you'll need. Make sure you read it carefully and install & download everything in advance!

Preparation

This workshop is conducted with the R statistical software, RStudio IDE, and gravity (Woelwer et al. 2023) package. Of course you're going to need tidyverse as well, or specifically dplyr package. You want to procure data beforehand too, and I will use CEPII data. let's discuss one by one.

Software

You'd want to use R and RStudio for this. The main reason I use R is because it's free. Stata is not. I think Stata is faster and a bit easier (R people will kill me if they see this) but not cheap. If you have Stata it's fine too. The command you'd want in Stata is `ppmlhdfe`.

Now onto R. You can procure R and RStudio from Posit's website. Get it here. I wrote the guide to install R and RStudio here, so you better check it out. It's written in Indonesian.

After that, you are going to need to install some packages. Follow my step until I told you to do type this on the console `install.packages(c("tidyverse","WDI","readxl","kableExtra"))`. You are going to do the same but you're going to few different stuff. Specifically, you need to add "gravity" and "writexl" on the list. That is, you need to type

```
install.packages(c("tidyverse","WDI","readxl","writexl","gravity"))
```

This step requires internet connection, but you'll need to do this only once.

Data

I procure data for this workshop from CEPII. From their website, CEPII is:

> he CEPII is the leading French center for research and expertise on the world economy. It contributes to the policy making process trough its independent in-depth analyses on international trade, migrations, macroeconomics and finance. The CEPII also produces databases and provides a platform for debate among academics, experts, practitioners, decision makers and other private and public stakeholders. Founded in 1978, the CEPII is part of the network coordinated by France Strategy, within the Prime Minister's services.

I use their BACI dataset (Gaulier and Zignago 2010) and gravity dataset (Conte, Cotterlaz, and Mayer 2022). You can get those from this link. BACI is under "international trade" banner while gravity is under "Gravity" banner. Specifically, I downloaded the 2017-2022 version of BACI and for the gravity dataset I downloaded the R version. You can of course download whichever version you like but for the purpose of this workshop maybe its best to stick with the same dataset as I.

You can also download from my drive.

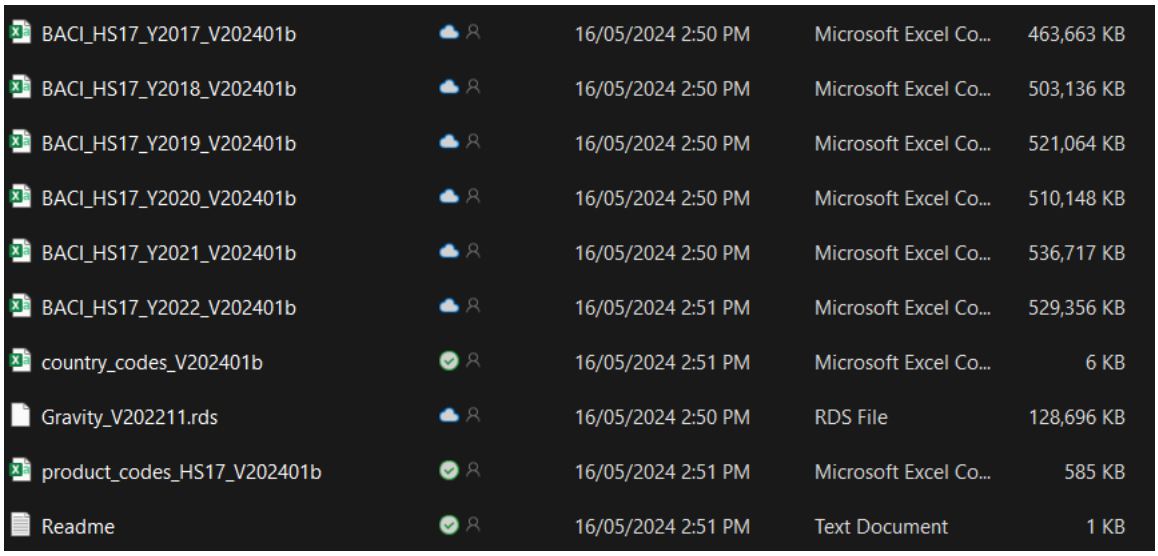
Note that the data here is **extremely large** in size so be mindful. You need hefty internet quota and reasonable speed. Also, you can try opening it with spreadsheet software but unless you have a strong computer, i'd advice against it. Use R instead.

In the CEPII website you can use various other dataset that may be useful for you. At the same time, there are various other source you can utilise for your actual project that's not necessarily from CEPII.

working directory

If you finished downloading data and installing softwares, you then need to set up a working directory. A working directory is basically a folder where you have all the data and your R script (R version of do file). For now what you want is to have a **folder filled with your downloaded data**. Make sure you know the path to this folder. I tend to use easy path for my projects and move it somewhere else when i finished. If you use github or the likes, it'll be even nicer because you can actually wipe out your local repo if you finish.

All in all, you should have a folder with these stuff in it:



Notes about the data country_codes, product_codes and Readme are all for reading BACI.

Packages

for this page I use these packages but you may not need all of them

```
library(tidyverse)
library(penppml) ## no need
library(writexl)
library(modelsummary) ## no need
library(gravity)
```

Simple gravity specification

Theory

The earliest (e.g., naive) gravity model taking directly from Newtonian gravity theory looks something like this:

$$X_{ij} = \tilde{G}\frac{Y_i E_j}{T_{ij}^{\theta}} \tag{0.1}$$

where X_{it} is the value of trade flow from country i to country j, $\tilde{G}$ is the gravitational constant (aka our usual constant), Y_i is the output in country i E_j is the value of expenditure in country j and T_{ij} is the total bilateral trade frictions / trade cost between country i and country j.

There are various other types of gravity equations, but let's start with a relatively simple one. One of my favorite simple gravity specification is a budget version of Silva and Tenreyro (2006) which is taken from Anderson and Wincoop (2003) which looks like this:

$$X_{ij} = \alpha_0 Y_i^{\alpha_1} Y_j^{\alpha_2} D_{ij}^{\alpha_3} e^{\theta_i d_i + \theta_j d_j} \tag{0.2}$$

where α_0 is your $\tilde{G}$, while Y is the output and expenditure which is proxied with GDP. D_{ij} is the distance between the two countries, which can be generalized as a vector of trade cost measures. Typically we use physical distance but also other types of bilateral trade cost. Lastly, the d_i and d_j is country-specific characteristics.

There are various variables used in Silva and Tenreyro (2006). log of exporter's and importer's GDP and GDP per capita. Various "distance" variables is used as well e.g., physical distance and variables like contiguity, common-language dummy, colonial-tie dummy and free trade agreement dummy.

Note that our regression consists only of two indices: exporter i and importer j. We are going to use the gravity data I mentioned earlier, slice the dataset to cover only one year chosen arbitrarily (which is 2019), and run Equation 0.2.

Setting data

first we load all the necessary data:

```
## Readubg data
gravity <- readRDS("Gravity_V202211.rds")
key<-read_csv("country_codes_V202401b.csv")
```

The gravity is the data from CEPII while key is storing some country codes. You can see the first 10 rows of the data and its variable names you call their name. Just type `gravity` or `key` in the console then hit enter. However, if you just want to look at the variable names, you can use `colnames()`

```
colnames(gravity)
```

```
 [1] "year"                "country_id_o"        "country_id_d"
 [4] "iso3_o"              "iso3_d"              "iso3num_o"
 [7] "iso3num_d"           "country_exists_o"    "country_exists_d"
[10] "gmt_offset_2020_o"   "gmt_offset_2020_d"   "distw_harmonic"
[13] "distw_arithmetic"    "distw_harmonic_jh"   "distw_arithmetic_jh"
[16] "dist"                "main_city_source_o"  "main_city_source_d"
[19] "distcap"             "contig"              "diplo_disagreement"
[22] "scaled_sci_2021"     "comlang_off"         "comlang_ethno"
[25] "comcol"              "col45"               "legal_old_o"
```

```
[28] "legal_old_d"            "legal_new_o"            "legal_new_d"
[31] "comleg_pretrans"        "comleg_posttrans"       "transition_legalchange"
[34] "comrelig"               "heg_o"                  "heg_d"
[37] "col_dep_ever"           "col_dep"                "col_dep_end_year"
[40] "col_dep_end_conflict"   "empire"                 "sibling_ever"
[43] "sibling"                "sever_year"             "sib_conflict"
[46] "pop_o"                  "pop_d"                  "gdp_o"
[49] "gdp_d"                  "gdpcap_o"               "gdpcap_d"
[52] "pop_source_o"           "pop_source_d"           "gdp_source_o"
[55] "gdp_source_d"           "gdp_ppp_o"              "gdp_ppp_d"
[58] "gdpcap_ppp_o"           "gdpcap_ppp_d"           "pop_pwt_o"
[61] "pop_pwt_d"              "gdp_ppp_pwt_o"          "gdp_ppp_pwt_d"
[64] "gatt_o"                 "gatt_d"                 "wto_o"
[67] "wto_d"                  "eu_o"                   "eu_d"
[70] "fta_wto"                "fta_wto_raw"            "rta_coverage"
[73] "rta_type"               "entry_cost_o"           "entry_cost_d"
[76] "entry_proc_o"           "entry_proc_d"           "entry_time_o"
[79] "entry_time_d"           "entry_tp_o"             "entry_tp_d"
[82] "tradeflow_comtrade_o"   "tradeflow_comtrade_d"  "tradeflow_baci"
[85] "manuf_tradeflow_baci"   "tradeflow_imf_o"        "tradeflow_imf_d"
```

As you can see, the column names are so plenty. Consult to the CEPII website or Conte, Cotterlaz, and Mayer (2022) to learn more. We will only use some of them, so we will filter these data to make it more concise. Specifically, we will (1) remove some countries, (2) remove non-2019, and (3) remove variables we are not using.

For variables, we will keep iso3_o, iso3_d, distw_harmonic, contig, comcol, comlang_off,gdp_o,gdp_d, gdpcap_o, gdpcap_d,fta_wto. Note that o means origin / exporter and d means destination / importer.

```r
## create a country list
ctr<-c("Albania", "Denmark", "Kenya", "Romania", "Algeria", "Djibouti", "Kiribati", "Russi

vrb<-c("iso3num_o","iso3num_d","year","iso3_o", "iso3_d", "distw_harmonic", "contig", "com

## keep 2019
gravity2<-gravity|>filter(year==2019)

## Keep countries in the list
key2<-key |> filter(country_name%in%ctr)
gravity2<-gravity2 |> filter(country_id_o %in% key2$country_iso3 & country_id_d %in% key2$
gravity2<-gravity2 |> select(vrb)

## Change trade units from 1000 USD to billion usd
gravity2<-gravity2|>
  mutate(tradeflow=tradeflow_baci/10e6)
```

```
17
18 ## Make a log versin
19 gravity2<-gravity2 |>
20   mutate(ldist=log(distw_harmonic),
21          lgdpo=log(gdp_o),
22          lgdpd=log(gdp_d),
23          lgdpco=log(gdpcap_o),
24          lgdpcd=log(gdpcap_d),
25          logtrade=log(1+tradeflow))
```

You can see in your environment tab the difference between gravity and gravity2 as well as between key and key2 on the number of observations and variables. Note that we also log non-dummy variables for gravity2 to redo Silva and Tenreyro (2006).

We will focus on the gravity2 as it will be the dataset we will run. You can quickly show summary statistics by typing `summary(gravity2)` on the console tab.

```
summary(gravity2)
```

```
  iso3num_o       iso3num_d          year         iso3_o
Min.   :  8.0   Min.   :  8.0   Min.   :2019   Length:12321
1st Qu.:204.0   1st Qu.:204.0   1st Qu.:2019   Class :character
Median :400.0   Median :400.0   Median :2019   Mode  :character
Mean   :415.5   Mean   :415.5   Mean   :2019
3rd Qu.:616.0   3rd Qu.:616.0   3rd Qu.:2019
Max.   :894.0   Max.   :894.0   Max.   :2019

   iso3_d          distw_harmonic      contig            comcol
Length:12321       Min.   :    4   Min.   :0.00000   Min.   :0.00000
Class :character   1st Qu.: 4459   1st Qu.:0.00000   1st Qu.:0.00000
Mode  :character   Median : 7587   Median :0.00000   Median :0.00000
                   Mean   : 7932   Mean   :0.01753   Mean   :0.09739
                   3rd Qu.:11024   3rd Qu.:0.00000   3rd Qu.:0.00000
                   Max.   :19676   Max.   :1.00000   Max.   :1.00000

 comlang_off         gdp_o               gdp_d              gdpcap_o
Min.   :0.0000   Min.   :1.779e+05   Min.   :1.779e+05   Min.   : 0.224
1st Qu.:0.0000   1st Qu.:1.419e+07   1st Qu.:1.419e+07   1st Qu.: 1.909
Median :0.0000   Median :4.805e+07   Median :4.805e+07   Median : 6.321
Mean   :0.1789   Mean   :4.785e+08   Mean   :4.785e+08   Mean   :15.262
3rd Qu.:0.0000   3rd Qu.:3.512e+08   3rd Qu.:3.512e+08   3rd Qu.:18.480
Max.   :1.0000   Max.   :1.428e+10   Max.   :1.428e+10   Max.   :85.335
                 NA's   :111         NA's   :111         NA's   :111
   gdpcap_d         fta_wto       tradeflow_baci        tradeflow
Min.   : 0.224   Min.   :0.0000   Min.   :        0   Min.   : 0.0000
```

```
 1st Qu.: 1.909   1st Qu.:0.0000   1st Qu.:      273   1st Qu.: 0.0000
 Median : 6.321   Median :0.0000   Median :     6343   Median : 0.0006
 Mean   :15.262   Mean   :0.2023   Mean   :   611172   Mean   : 0.0611
 3rd Qu.:18.480   3rd Qu.:0.0000   3rd Qu.:    87003   3rd Qu.: 0.0087
 Max.   :85.335   Max.   :1.0000   Max.   :149568313   Max.   :14.9568
 NA's   :111                       NA's   :2185        NA's   :2185
     ldist            lgdpo            lgdpd            lgdpco
 Min.   :1.386   Min.   :12.09   Min.   :12.09   Min.   :-1.4961
 1st Qu.:8.403   1st Qu.:16.47   1st Qu.:16.47   1st Qu.: 0.6466
 Median :8.934   Median :17.69   Median :17.69   Median : 1.8438
 Mean   :8.721   Mean   :17.93   Mean   :17.93   Mean   : 1.8087
 3rd Qu.:9.308   3rd Qu.:19.68   3rd Qu.:19.68   3rd Qu.: 2.9167
 Max.   :9.887   Max.   :23.38   Max.   :23.38   Max.   : 4.4466
                 NA's   :111     NA's   :111     NA's   :111
     lgdpcd            logtrade
 Min.   :-1.4961   Min.   :0.0000
 1st Qu.: 0.6466   1st Qu.:0.0000
 Median : 1.8438   Median :0.0006
 Mean   : 1.8087   Mean   :0.0393
 3rd Qu.: 2.9167   3rd Qu.:0.0087
 Max.   : 4.4466   Max.   :2.7699
 NA's   :111       NA's   :2185
```

Regression

Let's do 2 types of regression. First we do a regression using a normal ols, and secondly we do ppml.

```
reg1<-lm(logtrade~lgdpo+lgdpd+lgdpco+lgdpcd+ldist+contig+comcol+comlang_off+fta_wto)
reg2<-lm(logtrade~lgdpo+lgdpd+lgdpco+lgdpcd+ldist+contig+comcol+comlang_off+fta_wto+iso3_o
reg3<-ppml(data=gravity2,dependent_variable="tradeflow_baci",distance="distw_harmonic",add
reg4<-ppml(data=gravity2,dependent_variable="tradeflow_baci",distance="distw_harmonic",add
```

You can call each reg's table with `summary(reg1)`.

You can compare results with Silva and Tenreyro (2006). Note that they don't use fixed effects.

Table 1: Simple regression results

	OLS no ctr	OLS with ctr	PPML no ctr	PPML with ctr
(Intercept)	−0.660***	−1.472	−31.498***	$-1.103\,722 \times 10^{10}$
	(0.027)	(2.093)	(0.255)	$(1.692\,226 \times 10^{12})$
lgdpo	0.028***	0.052	0.895***	$6.730\,846 \times 10^{8}$
	(0.001)	(0.093)	(0.008)	$(1.031\,973 \times 10^{11})$
lgdpd	0.026***	0.058	0.814***	1.349 000
	(0.001)	(0.088)	(0.008)	(1.885 000)
lgdpco	0.000	−0.058	−0.041***	$-1.432\,036 \times 10^{9}$
	(0.001)	(0.218)	(0.012)	$(2.195\,597 \times 10^{11})$
lgdpcd	0.001	−0.101	−0.037**	−1.280 000
	(0.001)	(0.204)	(0.011)	(4.394 000)
contig	0.112***	0.091***	0.185***	$3.340\,000 \times 10^{-1}$***
	(0.011)	(0.010)	(0.037)	$(3.100\,000 \times 10^{-2})$
comcol	0.026***	0.027***	0.110	$5.190\,000 \times 10^{-1}$***
	(0.005)	(0.006)	(0.090)	$(7.400\,000 \times 10^{-2})$
comlang_off	−0.001	−0.009*	0.238***	$1.620\,000 \times 10^{-1}$***
	(0.004)	(0.004)	(0.035)	$(3.100\,000 \times 10^{-2})$
fta_wto	0.008*	0.008*	0.383***	$3.800\,000 \times 10^{-1}$***
	(0.004)	(0.004)	(0.028)	$(2.500\,000 \times 10^{-2})$
ldist	−0.034***	−0.046***		
	(0.002)	(0.002)		
dist_log			−0.606***	$-7.110\,000 \times 10^{-1}$***
			(0.013)	$(1.200\,000 \times 10^{-2})$
Num.Obs.	9990	9990	9990	9990
R2	0.264	0.404		
R2 Adj.	0.263	0.390		
AIC	−11 600.7	−13 283.8		
BIC	−11 521.4	−11 661.7		
Log.Lik.	5811.356	6866.892		
RMSE	0.14	0.12	0.21	0.16

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

TABLE 3.—THE TRADITIONAL GRAVITY EQUATION

Estimator: Dependent Variable:	OLS $\ln(T_{ij})$	OLS $\ln(1 + T_{ij})$	Tobit $\ln(a + T_{ij})$	NLS T_{ij}	PPML $T_{ij} > 0$	PPML T_{ij}
Log exporter's GDP	0.938**	1.128**	1.058**	0.738**	0.721**	0.733**
	(0.012)	(0.011)	(0.012)	(0.038)	(0.027)	(0.027)
Log importer's GDP	0.798**	0.866**	0.847**	0.862**	0.732**	0.741**
	(0.012)	(0.012)	(0.011)	(0.041)	(0.028)	(0.027)
Log exporter's GDP per capita	0.207**	0.277**	0.227**	0.396**	0.154**	0.157**
	(0.017)	(0.018)	(0.015)	(0.116)	(0.053)	(0.053)
Log importer's GDP per capita	0.106**	0.217**	0.178**	−0.033	0.133**	0.135**
	(0.018)	(0.018)	(0.015)	(0.062)	(0.044)	(0.045)
Log distance	−1.166**	−1.151**	−1.160**	−0.924**	−0.776**	−0.784**
	(0.034)	(0.040)	(0.034)	(0.072)	(0.055)	(0.055)
Contiguity dummy	0.314*	−0.241	−0.225	−0.081	0.202	0.193
	(0.127)	(0.201)	(0.152)	(0.100)	(0.105)	(0.104)
Common-language dummy	0.678**	0.742**	0.759**	0.689**	0.752**	0.746**
	(0.067)	(0.067)	(0.060)	(0.085)	(0.134)	(0.135)
Colonial-tie dummy	0.397**	0.392**	0.416**	0.036	0.019	0.024
	(0.070)	(0.070)	(0.063)	(0.125)	(0.150)	(0.150)
Landlocked-exporter dummy	−0.062	0.106*	−0.038	−1.367**	−0.873**	−0.864**
	(0.062)	(0.054)	(0.052)	(0.202)	(0.157)	(0.157)
Landlocked-importer dummy	−0.665**	−0.278**	−0.479**	−0.471**	−0.704**	−0.697**
	(0.060)	(0.055)	(0.051)	(0.184)	(0.141)	(0.141)
Exporter's remoteness	0.467**	0.526**	0.563**	1.188**	0.647**	0.660**
	(0.079)	(0.087)	(0.068)	(0.182)	(0.135)	(0.134)
Importer's remoteness	−0.205*	−0.109	−0.032	1.010**	0.549**	0.561**
	(0.085)	(0.091)	(0.073)	(0.154)	(0.120)	(0.118)
Free-trade agreement dummy	0.491**	1.289**	0.729**	0.443**	0.179*	0.181*
	(0.097)	(0.124)	(0.103)	(0.109)	(0.090)	(0.088)
Openness	−0.170**	0.739**	0.310**	0.928**	−0.139	−0.107
	(0.053)	(0.050)	(0.045)	(0.191)	(0.133)	(0.131)
Observations	9613	18360	18360	18360	9613	18360
RESET test p-values	0.000	0.000	0.204	0.000	0.941	0.331

Figure 1: source: Silva and Tenreyro (2006)

By the way, you can save the regression table using `modelsummary()`. don't forget to run `library(modelsummary)` first. You can use xls extension, but also doc. I personally like .html more.

```
regtab<- list(
  "OLS no ctr" = reg1,
  "OLS with ctr"  = reg2,
  "PPML no ctr"=reg3,
  "PPML with ctr"=reg4
)
modelsummary(regtab,output="regtab.xlsx")
```

Product level gravity

Theory

We then proceed to a higher-dimension trade data which you may be interested in. In the field, UI students often interested largely in Indonesian affairs. That is, we are not interested so much in the bilateral flow of all countries, but only on Indonesia. However, we often use more granular dimension than just exporter/importer. Often times we use indices

like time, commodities or industries, or even firms (shamelessly inserting my paper here Gupta (2023)).

Now, if you are planning to do these kinds of studies, then you are going to need to tackle higher degree dataset and merging the gravity variables. Most often you can get these variables from World Development Indicators but CEPII is ok for now (note the main problem of CEPII is its timeliness).

The theory isn't so different compared to our previous gravity model. What we want is an additional indices. We are going to estimate something similar as Equation 0.2 but with more indices. We need to care about multilateral resistance (MR) and we can use dummies since we now have more variations from indices like time and HS code.

According to Yotov (2022), we need at least 3 dummies to run a multi-country, multi-time and multi-goods/sectors[1]. We need to have exporter-time dummy, importer-time dummy and country-pair dummy. We need to construct this first. Note that these dummies will likely absorb some of your variables like distance (consistant between pair across time, typically).

So we will do the HS,time varying version of Equation 0.2:

$$X_{ijpt} = \alpha_0 Y_{it}^{\alpha_1} Y_{jt}^{\alpha_2} D_{ijpt}^{\alpha_3} e^{\theta_1 o_{it} + \theta_2 d_{jt} + \theta_3 p_{ij}} \tag{0.3}$$

Setting data

This time we need BACI data. Brace yourself because this dataset is HUGE. We read 5 different years.

```
t2017<-read_csv("BACI_HS17_Y2017_V202401b.csv")
t2018<-read_csv("BACI_HS17_Y2018_V202401b.csv")
t2019<-read_csv("BACI_HS17_Y2019_V202401b.csv")
t2020<-read_csv("BACI_HS17_Y2020_V202401b.csv")
t2021<-read_csv("BACI_HS17_Y2021_V202401b.csv")


## Combining all
trade<-rbind(t2017,t2018,t2019,t2020,t2021)

remove(t2017,t2018,t2019,t2020,t2021)
```

I used `read_csv` from the tydiverse package for reading .csv. rbind is to stack all BACI data (it was separated per year), then I remove the individual BACI to save environment space.

[1] unless you have domestic trade data which we typically don't. If you do, then there's borders dummy. More on Yotov (2022).

At this point, you can try checking out the two datasets. You can try looking at both data by calling their names. Alternatively, just look at the column names with `colnames()`. Let's try the BACI frist.

```
colnames(trade)
```

```
[1] "t" "i" "j" "k" "v" "q"
```

There are only 6 columns / variables. Here's some information on what thos means

Table 2: Variable explanations

var	meaning
t	year
i	exporter
j	importer
k	product
v	value
q	quantity

Products in Harmonized System 6-digit nomenclature. Values in thousand USD and quantities in metric tons. Exporter and importer is codified using CEPII codes. the codes and it means can be found in the "key" dataset. To have country identities into the BACI dataset, we need to join the two.

To join the two datasets, we need a key variable. A key variable is the variable connecting the two variables. Both needs the same name. So first we need to assign the same name for exporter and importer codes between BACI and gravity.

We know that i in BACI is iso3num_o in gravity, while j in BACI is iso3num_d in gravity. So we rename the one in BACI so both have the same name:

```
## Rename variable
trade2<-trade|>rename(iso3num_o=i,iso3num_d=j,year=t)

## Change ctr to reduce computation problem
ctr<-c("IDN","SGP","VNM","MYS","THA","PHL","USA","CHN","JPN","KOR")

## Kita ulangi gravity2 karena sekarang perlu tahun 2017-2021
gravity2<-gravity|>filter(year>2016 & year<2022)
gravity2<-gravity2 |> filter(iso3_o %in%ctr & iso3_d %in% ctr) ## notice the change
gravity2<-gravity2 |> select(vrb)

## gabung dengan trade2
```

```r

gabung<-left_join(gravity2,trade2,by=c("year","iso3num_o","iso3num_d"))

## make v from thousand usd to million usd

gabung<-gabung|>mutate(v=v/1000)
```

Check the results with gabung or `View(gabung)`. The most important thing here is that you have to make sure you understand the changes in variations! Now that we have time and HS (k), a pair of countries can have multiple observations in different year and different goods. tradeflow_baci will be repeated because this is the total trade, while now we focus on v and q as the X_{ijpt}.

Before we go, however, we need to generate our dummies! Remember, we need to make three dummies, o_{it},d_{jt} and p_{ij} (see Equation 0.3). To do that, we do this:

```r
gabung <- gabung |>
  mutate(ooo=interaction(iso3num_o,year),
         ddd=interaction(iso3num_d,year),
         ppp=interaction(iso3num_o,iso3num_d))
```

You can check again whether it's made. if you do `tibble(gabung)` you will see that we have created our factor variables. Oh yes, do not forget to log non-factors.

```r
gabung<-gabung |>
  mutate(ldist=log(distw_harmonic),
         lgdpo=log(gdp_o),
         lgdpd=log(gdp_d),
         lgdpco=log(gdpcap_o),
         lgdpcd=log(gdpcap_d),
         logtrade=log(1+v)) ## note the difference with before
```

Why don't we show the quick summary statistics?

```r
summary(gabung)
```

```
   iso3num_o       iso3num_d          year         iso3_o
 Min.   :156.0   Min.   :156.0   Min.   :2017   Length:1882603
 1st Qu.:360.0   1st Qu.:360.0   1st Qu.:2018   Class :character
 Median :458.0   Median :458.0   Median :2019   Mode  :character
 Mean   :482.5   Mean   :504.9   Mean   :2019
 3rd Qu.:702.0   3rd Qu.:702.0   3rd Qu.:2020
 Max.   :840.0   Max.   :840.0   Max.   :2021
```

```
   iso3_d          distw_harmonic       contig           comcol
Length:1882603     Min.   :   10    Min.   :0.0      Min.   :0
Class :character   1st Qu.: 1193    1st Qu.:0.0      1st Qu.:0
Mode  :character   Median : 2289    Median :0.0      Median :0
                   Mean   : 2598    Mean   :0.1      Mean   :0
                   3rd Qu.: 3836    3rd Qu.:0.0      3rd Qu.:0
                   Max.   :15486    Max.   :1.0      Max.   :1
                   NA's   :775034   NA's   :775034   NA's   :775034
 comlang_off         gdp_o               gdp_d              gdpcap_o
Min.   :0.0      Min.   :2.814e+08   Min.   :2.814e+08   Min.   : 3.0
1st Qu.:0.0      1st Qu.:3.755e+08   1st Qu.:3.626e+08   1st Qu.: 7.2
Median :0.0      Median :1.042e+09   Median :5.060e+08   Median :10.4
Mean   :0.1      Mean   :3.605e+09   Mean   :2.766e+09   Mean   :22.5
3rd Qu.:0.0      3rd Qu.:5.038e+09   3rd Qu.:1.725e+09   3rd Qu.:38.8
Max.   :1.0      Max.   :2.300e+10   Max.   :2.300e+10   Max.   :72.8
NA's   :775034   NA's   :397835      NA's   :456083      NA's   :397835
   gdpcap_d         fta_wto       tradeflow_baci           k
Min.   : 3.0     Min.   :0.0      Min.   :   592555   Length:1882603
1st Qu.: 3.9     1st Qu.:1.0      1st Qu.:  8542294   Class :character
Median :10.3     Median :1.0      Median : 14608561   Mode  :character
Mean   :20.6     Mean   :0.9      Mean   : 29785446
3rd Qu.:34.8     3rd Qu.:1.0      3rd Qu.: 39777972
Max.   :72.8     Max.   :1.0      Max.   :500928196
NA's   :456083   NA's   :775034   NA's   :1003024
      v                   q                  ooo                ddd
Min.   :    0.00   Min.   :       0   458.2019:  59882   360.2020:  65802
1st Qu.:    0.01   1st Qu.:       1   458.2018:  59084   360.2019:  64772
Median :    0.11   Median :      11   458.2020:  58944   360.2021:  64672
Mean   :    6.97   Mean   :    5709   458.2021:  58052   360.2018:  64286
3rd Qu.:    1.08   3rd Qu.:     150   458.2017:  55982   458.2019:  63506
Max.   :42892.73   Max.   :88344459   360.2019:  55798   458.2018:  63350
NA's   :225        NA's   :27377      (Other) :1534861   (Other) :1496215
     ppp              ldist             lgdpo            lgdpd
458.360:  69600   Min.   :2.3      Min.   :19.5     Min.   :19.5
360.458:  60084   1st Qu.:7.1      1st Qu.:19.7     1st Qu.:19.7
458.704:  51616   Median :7.7      Median :20.8     Median :20.0
704.458:  45440   Mean   :7.6      Mean   :21.0     Mean   :20.7
156.458:  45204   3rd Qu.:8.3      3rd Qu.:22.3     3rd Qu.:21.3
704.360:  44976   Max.   :9.6      Max.   :23.9     Max.   :23.9
(Other):1565683   NA's   :775034   NA's   :397835   NA's   :456083
    lgdpco           lgdpcd          logtrade
Min.   :1.1      Min.   :1.1      Min.   : 0.0000
1st Qu.:2.0      1st Qu.:1.4      1st Qu.: 0.0096
Median :2.3      Median :2.3      Median : 0.1011
Mean   :2.6      Mean   :2.5      Mean   : 0.6071
3rd Qu.:3.7      3rd Qu.:3.5      3rd Qu.: 0.7332
```

```
Max.   :4.3       Max.   :4.3       Max.   :10.6665
NA's   :397835    NA's   :456083    NA's   :225
```

Regression

```
ger1<-lm(logtrade~lgdpo+lgdpd+lgdpco+lgdpcd+ldist+contig+comcol+comlang_off+fta_wto)
ger2<-lm(logtrade~lgdpo+lgdpd+lgdpco+lgdpcd+ldist+contig+comcol+comlang_off+fta_wto+iso3_o
ger3<-ppml(data=gravity2,dependent_variable="tradeflow_baci",distance="distw_harmonic",add
ger4<-ppml(data=gravity2,dependent_variable="tradeflow_baci",distance="distw_harmonic",add
```

As you can see, the difference is apparent when we use HS-6-digit instead of total trade. This is of course the case since now we have wild, uncontrolled variability in the goods characteristics. Indeed, the gravity equation is much better suited predicting total trade where country and year characteristics dominates and industry/goods heterogeneity is absorbed by the total trade. Remember, I use only small number of countries with tons of HS 6 digit[2]. Moreover, PPML sometimes act funny where zeroes are abundant combined with many dummies. Convergence sometimes unachieved / converge to a very strange parameters.

UI students typically only interested in Indonesia, so country pair dummy and indonesia-time dummy often not needed.

Closing

OKay now you are ready to run regression yourself. Try to replicate what I do here and you prolly finished 50% of your thesis. You then can work to update this with your own hypothesis, adding more variable and more concentrated.

Running this on Stata is also excellent. I must confess that R is also speedy (these guys making the package is extremely good), but Stata is a bit more intuitive and compute you with important stats as well such as pseudo-R. Nevertheless, now you should be able to do both!

I cannot emphasize enough references in Yotov (2022). Whatever you want to do, a paper prolly covered it already. Learn from them and look for an insight to add. Work with your spv and you'll be fine.

[2] I added JPN, KOR, CHN and USA in this version. Previously it was only ASEAN6 and results were pretty funny since within-ASEAN trade isn't so large.

Table 3: Simple regression results

	OLS no dum	OLS with dum	PPML no dum	PPML with dum
(Intercept)	−5.901***	−5.665***	−17.604***	$3.278\,106 \times 10^9$
	(0.031)	(1.087)	(0.414)	$(1.844\,453 \times 10^{10})$
lgdpo	0.235***	0.386**	0.484***	$-4.985\,363 \times 10^8$
	(0.001)	(0.124)	(0.010)	$(1.889\,584 \times 10^9)$
lgdpd	0.124***	−0.012	0.521***	$2.892\,028 \times 10^8$
	(0.001)	(0.053)	(0.010)	$(1.350\,209 \times 10^9)$
lgdpco	−0.048***	−0.005	0.017	$-8.550\,758 \times 10^8$
	(0.001)	(0.062)	(0.015)	$(5.409\,681 \times 10^9)$
lgdpcd	−0.024***	−0.034	0.028+	$1.570\,095 \times 10^9$
	(0.001)	(0.032)	(0.015)	$(6.052\,223 \times 10^9)$
contig	0.114***	0.046*	0.523***	$1.491\,381 \times 10^9$
	(0.004)	(0.023)	(0.045)	$(7.040\,266 \times 10^9)$
comcol	0.125***	−0.282	0.523***	$-5.997\,149 \times 10^9$
	(0.008)	(0.343)	(0.100)	$(1.713\,353 \times 10^{10})$
comlang_off	0.067***	0.042	0.288***	$6.816\,366 \times 10^9$
	(0.004)	(0.078)	(0.041)	$(1.721\,904 \times 10^{10})$
fta_wto	−0.119***	0.481	0.275***	$1.141\,830 \times 10^9$
	(0.005)	(0.506)	(0.040)	$(9.983\,440 \times 10^9)$
ldist	−0.090***	−0.252		
	(0.002)	(0.164)		
dist_log			−0.273***	2.149 000
			(0.022)	(2.415 000)
Num.Obs.	1 107 423	1 107 423	1 107 423	1 107 423
R2	0.093	0.113		
R2 Adj.	0.093	0.113		
AIC	3 223 636.5	3 199 383.8		
BIC	3 223 767.6	3 201 076.1		
Log.Lik.	−1 611 807.266	−1 599 549.909		
RMSE	1.04	1.03	124.10	124.07

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

References

Anderson, James E., and Eric van Wincoop. 2003. "Gravity with Gravitas: A Solution to the Border Puzzle." Journal Article. *The American Economic Review* 93 (1): 24.

Conte, Madallena, Pierre Cotterlaz, and Thierry Mayer. 2022. "The CEPII Gravity Database." Working Papers 2022-05. CEPII. http://www.cepii.fr/CEPII/fr/publications/wp/abstract.asp?NoDoc=2726.

Gaulier, Guillaume, and Soledad Zignago. 2010. "BACI: International Trade Database at the Product-Level. The 1994-2007 Version." Working Papers 2010-23. CEPII. http://www.cepii.fr/CEPII/fr/publications/wp/abstract.asp?NoDoc=2726.

Gupta, Krisna. 2023. "The Heterogeneous Impact of Tariffs and Ntms on Total Factor Productivity for Indonesian Firms." Journal Article. *Bulletin of Indonesian Economic Studies* 59 (2): 269–300. https://doi.org/10.1080/00074918.2021.2016613.

Silva, Santos, and Silvana Tenreyro. 2006. "The Log of Gravity." Journal Article. *The Review of Economics and Statistics* 88 (4): 19.

Woelwer, Anna-Lena, Jan Pablo Burgard, Joshua Kunst, and Mauricio Vargas. 2023. *Gravity: Estimation Methods for Gravity Models*. http://pacha.dev/gravity/.

Yotov, Yoto. 2022. "Gravity at Sixty: The Workhorse Model of Trade." Journal Article. *CESifo Working Papers* 9584. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4037001.