Gravity equation tutorial

Neil Rankin

March 22, 2017

1. Introduction

This handout goes through the main steps in empirical trade analysis by using the gravity model.

The first version of the code was written in Stata (by Malte Ehrich). For most this is likely to be the best choice of statistical software. However, you are free to use whatever package (provided it allows for scripts) you are most comfortable. I will try and add R snippets for those who are R users. [1]

Sometimes very tiny software-specific steps are ommited for convenience but should not affect your understanding. The full set of steps is present in the Stata do-file.

The tutorial has three parts:

1) Construction of a gravity data set;
2) Estimation of three gravity equations;
3) interpretation of output tables.

The main reference is Bacchetta *et al* (2012).

2. Research motivation

Question:

How does membership of the World Trade Organisation (WTO) affect trade?

Hypotheses:

1. Trade is higher if both countries are members of the WTO.
2. Trade is lower (than when both countries are WTO members) if only one country is a member.

3. Construction of a gravity dataset

Data sources

- Trade flows: Comtrade-database
- GDP: Worldbank, current US-$
- Gravity variables: CEPII Distance and other proxies for trade costs, e.g. dummies for colonial history etc.
- WTO membership: WTO
- Tariffs: World Integrated Trade Solution (WITS) Worldbank

[1] This document is written in RMarkdown which you should be able to fork from my GitHub account *(neilrankinza)*. There is also a "gravity" package in R (thanks to Ruan Erasmus for pointing this out). I am not really familar with this but it looks like it might be worth investigating if you are using R.

Loading data (part 1 in do file)

Most raw data are provided in .csv or Excel format.

Stata example:

```
insheet using "tradeflows.csv", clear delimiter(";") names*
```

Label your variables - you'll quickly forget what they mean if you don't

```
label var imports "Imports value in US-$"
```

Now, save the data set

```
save "tradeflows.dta", replace
```

Do the same for all the remaining data sets (GDP, CEPII,WTOmembership, tariffs,...).

R example:

```
tradeflows <- read.csv("Data/tradeflows.csv", sep = ";")
```

Create country-pair combinations (part 2 in do file)

Table: Mini-example of unbalanced panel

Importer	Exporter	Year	Imports
South Africa	Germany	2000	6543
South Africa	Germany	2001	23434
South Africa	Germany	2002	665462
South Africa	France	1999	5321
South Africa	France	2002	62134

Observations for the country-pairs South Africa-Germany for 1999 and South Africa-France for 2000 and 2001 are missing. We need to replace those with zeros to have a balanced dataset. The idea here is to take into account cases of 0 trade. In this example this is likely to create two challenges. The first is one of memory - this will create a dataset of 1,000,000 lines. Your computer should be able to handle this but size will increase exponentially as the country-pair numbers increase. The second is that if you take the ln(0) this vale will be NA and thus not included in the regression (thus making this exercise moot).

Stata code

```
fillin Importer Exporter Year
```

```
replace Imports=0 if Imports==.
```

```
/*could alternatively make Imports a very small number so that you could ln it*/
```

```
replace Imports=0.0001 if Imports==.
```

```
save "gravity_temp1.dta", replace
```

R code

```
vals <- expand.grid(importer = unique(tradeflows$importer),
                    exporter = unique(tradeflows$importer),
                    year = unique(tradeflows$year))
tradeflows1 <- dplyr::left_join(vals,tradeflows)
```

```
## Joining, by = c("importer", "exporter", "year")
```

```
tradeflows1$imports <- ifelse(is.na(tradeflows1$imports), 0.0001, tradeflows1$imports)
tradeflows1 <- tradeflows1 %>% filter(imports>0) %>% mutate(ln_imports=log(imports))
```

Table: Mini-example of balanced panel

Importer	Exporter	Year	Imports
South Africa	Germany	1999	0
South Africa	Germany	2000	6543
South Africa	Germany	2001	23434
South Africa	Germany	2002	665462
South Africa	France	1999	5321
South Africa	France	2000	0
South Africa	France	2001	0
South Africa	France	2002	62134

Reshape and merge country-specific data with trade flows

- GDP-data usually comes in wide-format

Table: GDP in wide-format

Country	Year_1960	Year_1961	Year_1962
South Africa	45646	456456	4564523
France	563456	456566	45556436
Denmark	56546	35321	96434

To merge GDP with trade flow-data, we need to reshape GDP from wide to long format. [2]

Stata

```
reshape long Year, i(countrycode) j(Year)
```

R

```
library(readstata13)
GDP <- read.dta13("Data/GDP.dta")

long_GDP <- melt(GDP)
```

```
## Using countryname, countrycode, indicatorname, indicatorcode as id variables
```

```
long_GDP <- dplyr::rename(long_GDP, year = variable, GDP = value)
#using stringr package (another package of the tidyverse)
library(stringr)
long_GDP$year <- as.numeric(str_sub(long_GDP$year, 6, 9))
```

Table: GDP in long-format

Country	Year	GDP
South Africa	1960	45646

[2] In the data science/R world 'tidy data' is often mentioned. This is a long format where every row is a specific observation (in this case country-year combination) and every column is a variable. There is also an 'extreme' tidy data version used often for packages like ggplot where there is only one column of values and the data is a 'stacked' set of separate varaibles.

Country	Year	GDP
South Africa	1961	456456
South Africa	1962	4564523
France	1960	563456
France	1961	456566
France	1962	45556436
Denmark	1960	56546
Denmark	1961	35321
Denmark	1962	96434

Save the reshaped GDP dataset

Stata

```
save "GDP_new.dta", replace
```

Or if you are using R then you can have multiple objects open at the same time (a **HUGE** benefit) so you don't have to save.

Now, we create GDP for importers and exporter

Stata

```
use "GDP_new.dta", clear

rename country Exporter

rename GDP GDP_Exporter

save "GDP_Exporter.dta", replace

use "GDP_new.dta", clear

rename country Importer

rename GDP GDP_Importer

save "GDP_Importer.dta", replace
```

R

```
#using dplyr here
GDP_exporter <- long_GDP %>% dplyr::rename(GDP_exp = GDP, exporter = countrycode) %>% filter(GDP_exp > 0
#now only keeping the variables we want
GDP_exporter <- GDP_exporter %>% select(exporter, year, ln_GDP_exp)

#importer
GDP_importer <- long_GDP %>% dplyr::rename(GDP_imp = GDP, importer = countrycode) %>% filter(GDP_imp > 0
#now only keeping the variables we want
GDP_importer <- GDP_importer %>% select(importer, year, ln_GDP_imp)
```

Merge trade and GDP datasets.

Stata

```
use "gravity_temp1.dta", clear

sort Exporter Year

merge m:1 Exporter Year using "GDP_exporter.dta"
```

We only keep exporter-year pairs for which we have observations in both datasets.

Stata

```
keep if _merge == 3
drop _merge
```

Now, we add GDP-importer.

Stata

```
sort Importer Year
merge m:1 Importer Year using "GDP_Importer.dta"
keep if _merge == 3
drop _merge
sort Exporter Importer Year
save "gravity_temp2.dta", replace
```

Now we add the standard gravity-type variables (e.g. language, border, etc.).

Stata

```
use "gravity_temp2.dta", clear
sort Exporter Importer Year
merge m:1 Exporter Importer using "CEPII.dta"
keep if _merge == 3
drop _merge
sort Exporter Importer Year
merge m:1 Exporter Importer using "Religion.dta"
keep if _merge == 1 | _merge == 3
drop _merge
replace Religion = 0 if Religion == .
sort Exporter Importer Year
save "gravity_temp3.dta", replace
```

Before we run our first regression, we need to take logs of imports, GDP, and distance to linearise our model.

Stata

```
gen ln_Imports = log(Imports)
label var ln_Import "Log of Imports value"
gen ln_GDP_Exporter = log(GDP_Exporter)
label var ln_GDP_Exporter "log of Exporter's GDP"
gen ln_GDP_Importer = log(GDP_Importer)
label var ln_GDP_Importer "log of Importer's GDP"
gen ln_Dist = log(dist)
label var ln_Dist "log of distance"
save "gravity_temp3.dta", replace
```

R

That was a lot of work in Stata, let's see what we can do in R.

```
data1 <- left_join(tradeflows1, GDP_exporter, by = c("exporter", "year"))
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## character vector and factor, coercing into character vector
```

```
data1 <- left_join(data1, GDP_importer, by = c("importer", "year"))
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## character vector and factor, coercing into character vector
```

```
# do for CEPII, religion etc
cepii <- readstata13::read.dta13("Data/dist_cepii224.dta")
cepii <- cepii %>% rename(importer = country, exporter = partner)

data1 <- left_join(data1, cepii, by = c("exporter", "importer"))

data1$ln_Dist <- log(data1$dist)
```

We run our first regression that we call "Tinbergen".

Stata

```
reg ln_Imports ln_GDP_Exporter ln_GDP_Importer ln_Dist
```

```
outreg2* creates our tables and needs to be performed after each regression.
```

```
outreg2 ln_GDP_Exporter ln_GDP_Importer ln_Dist using Stellenbosch, tex addtext(Year FE,
No, Year FE, No) e(N) ctitle("Tinbergen") replace
```

VARIABLES	Tinbergen
ln_GDP_Exporter	1.092***
	(0.00190)
ln_GDP_Importer	0.885***
	(0.00184)
ln_Dist	-1.340***
	(0.00514)
Constant	-28.42***
	(0.0798)
Observations	291,859
R-squared	0.630
Importer-Year FE	No
Exporter-Year FE	No
N	291859

Standard errors in parentheses *** p<0.01, ** p<0.05, * p<0.1

R

```
model1 <- lm(ln_imports ~ ln_GDP_exp + ln_GDP_imp + ln_Dist,  data1, na.action=na.omit)
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

```
stargazer(model1)
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Tue, Mar 28, 2017 - 12:46:04 PM

Table 6:

	Dependent variable:
	ln_imports
ln_GDP_exp	1.971***
	(0.003)
ln_GDP_imp	1.614***
	(0.003)
ln_Dist	−1.418***
	(0.010)
Constant	−70.207***
	(0.148)
Observations	526,440
R^2	0.530
Adjusted R^2	0.530
Residual Std. Error	5.944 (df = 526436)
F Statistic	198,044.900*** (df = 3; 526436)
Note:	*p<0.1; **p<0.05; ***p<0.01

Let's add further dummies for trade costs (column (2).

Stata

```
reg ln_Imports ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off
```

```
outreg2 ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off using Stellenbosch,
tex addtext(Year FE, No, Year FE, No) e(N) ctitle("Tinbergen-extended")
```

R

```
model2 <- lm(ln_imports ~ ln_GDP_exp + ln_GDP_imp + ln_Dist + colony + contig + comlang_off,  data1, na
stargazer(model1, model2)
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Tue, Mar 28, 2017 - 12:46:05 PM

Let's generate our target variables. The dummy "onein" equals one if one of both trading partners is member of the WTO in year t , "bothin" equals one if both are members of the WTO in year t and "nonein" equals one if both are not member of theWTO in year t.

Stata

```
use "joinWTO.dta", clear
```

```
rename country Exporter
```

```
rename join join_Exporter
```

Table 7:

	Dependent variable:	
	ln_imports	
	(1)	(2)
ln_GDP_exp	1.971***	2.013***
	(0.003)	(0.003)
ln_GDP_imp	1.614***	1.656***
	(0.003)	(0.003)
ln_Dist	−1.418***	−1.282***
	(0.010)	(0.010)
colony		1.157***
		(0.079)
contig		0.776***
		(0.069)
comlang_off		2.566***
		(0.023)
Constant	−70.207***	−73.769***
	(0.148)	(0.152)
Observations	526,440	526,440
R^2	0.530	0.543
Adjusted R^2	0.530	0.543
Residual Std. Error	5.944 (df = 526436)	5.866 (df = 526433)
F Statistic	198,044.900*** (df = 3; 526436)	104,048.900*** (df = 6; 526433)
Note:		*p<0.1; **p<0.05; ***p<0.01

```
save "joinWTO_Exporter.dta", replace
use "joinWTO.dta", clear
rename country Importer
rename join join_Importer
save "joinWTO_Importer.dta", replace
use "gravity_temp3.dta", clear
sort Exporter Year
merge m:1 Exporter using "joinWTO_Exporter.dta"
drop _merge sort Importer Year
merge m:1 Importer using "joinWTO_Importer.dta"
drop _merge
sort Exporter Importer Year
save "gravity_temp4.dta", replace
use "gravity_temp4.dta", clear
foreach var in onein bothin nonein {
gen 'var' = 0 }
replace onein = 1 if (join_Exporter <= Year & join_Importer > Year) |
(join_Importer <= Year & join_Exporter > Year)
label var onein "one of the country pair is member of the WTO"
replace bothin = 1 if (join_Exporter <= Year & join_Importer <= Year)
label var bothin "both countries is member of the WTO"
replace nonein = 1 if (join_Exporter > Year & join_Importer > Year)
label var nonein "none of the country pair is member of the WTO"
save "gravity.dta", replace
```

Lets run the next regression which includes our target variables (column (3),WTO).

Stata

```
reg ln_Imports ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off bothin
onein    outreg2 ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off bothin
onein using Stellenbosch, tex addtext(Year FE, No, Year FE, No) e(N) ctitle("WTO")
```

We have learned that one should control for 186 multilateral resistance. Therefore, we need country- and year-dummies.

Stata

```
use "gravity.dta", clear
tab Exporter, gen(Exporter_)
tab Importer, gen(Importer_)
tab Year, gen(Year_)
save "gravity.dta", replace
```

How does this affect our estimation results (column (4),MRT)?

```
reg ln_Imports ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off bothin
onein Exporter_* Importer_* Year_*  outreg2 ln_GDP_Exporter ln_GDP_Importer ln_Dist colony
contig comlang_off bothin onein using Stellenbosch, tex addtext(Year FE, Yes, Importer
FE, Yes, Exporter FE, Yes ) e(N) ctitle("MRT")
```

R example

```
#have not created WTO variables in here yet
model3 <- lm(ln_imports ~ ln_GDP_exp + ln_GDP_imp + ln_Dist + colony + contig + comlang_off + as.factor
stargazer(model1, model2, model3)
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Tue, Mar 28, 2017 - 12:48:38 PM

We will now make use of panel data.

$$\begin{aligned} lnX_{ijt} = \beta_0 + \beta_1 lnGDP_{it} + \beta_2 lnGDP_{jt} + \beta_3 lnDist_{ij} + \beta_4 Colony_{ij} + \beta_5 Language_{ij} + \beta_6 Contiguity_{ij} \\ + \beta_7 onein_{ijt} + \beta_8 bothin_{ijt} + \beta_9 nonein_{ijt} \\ + \mu_i + \eta_j + \nu_t + \epsilon_{ijt} \end{aligned}$$

Define the panel structure

Stata

```
egen pairid = group(Importer Exporter)
```

```
xtset pairid Year
```

Within-group estimation (fixed effects)

Stata

```
xtreg ln_Imports ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off onein
bothin nonein Exporter_* Importer_* Year_*, fe
```

```
outreg2 ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off bothin onein
using Stellenbosch, tex addtext(Year FE, Yes, Importer FE, Yes, Exporter Year, Yes) e(N)
ctitle("FE")
```

Random effects `xtreg ln_Imports ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off onein bothin nonein Exporter_* Importer_* Year_*, re outreg2 ln_GDP_Exporter ln_GDP_Importer ln_Dist colony contig comlang_off bothin onein using Stellenbosch, tex addtext(Year FE, Yes, Importer FE, Yes, Exporter Year, Yes) e(N) ctitle("RE")`

R

```
#here it is probably useful to invetsigate the gravity package
```

Table 8:

	Dependent variable:		
	ln_imports		
	(1)	(2)	(3)
ln_GDP_exp	1.971***	2.013***	1.640***
	(0.003)	(0.003)	(0.027)
ln_GDP_imp	1.614***	1.656***	1.797***
	(0.003)	(0.003)	(0.027)
ln_Dist	−1.418***	−1.282***	−1.458***
	(0.010)	(0.010)	(0.011)
colony		1.157***	0.882***
		(0.079)	(0.078)
contig		0.776***	1.521***
		(0.069)	(0.063)
comlang_off		2.566***	2.244***
		(0.023)	(0.025)
as.factor(exporter)AFG			0.361*
			(0.212)
as.factor(exporter)AGO			−2.605***
			(0.158)
as.factor(exporter)ALB			0.096
			(0.152)
as.factor(exporter)ARE			1.259***
			(0.179)
as.factor(exporter)ARG			3.087***
			(0.201)
as.factor(exporter)ARM			−0.789***
			(0.151)
as.factor(exporter)ATG			1.340***
			(0.153)
as.factor(exporter)AUS			4.079***
			(0.212)
as.factor(exporter)AUT			4.047***
			(0.201)
as.factor(exporter)AZE			−1.365***
			(0.155)
as.factor(exporter)BDI			0.591***
			(0.152)
as.factor(exporter)BEN			0.693***
			(0.151)