GRAVITY MODELS USING EUROPEAN UNION PANEL DATA

ECONOMETRICS II: PROJECT IV

NANA OSEI SARPONG
11371873

Contents

INTRODUCTION

The gravity model has been a longstanding empirical model in international economics. In this project, we estimate the gravity model in a similar fashion as Serlanga & Shin (2007 using data from the European Union. The data spans 1960-2001, covering the bilateral trade flows between 15 European Union member states.

THEORETICAL MODEL

In recent times, the gravity model has become the benchmark model for assessing various flavours of issues relating to regional trading groups, currency unions, trade distortions, among other as seen in papers by various authors such as Bougheas et al., (1999); Frankel and Rose, (2002); Glick and Rose, (2002); Martinez-Zaroso and Nowak-Lehmann, (2003). Sir Isaac Newton's law of universal gravitation, formulated in the 17th century later found its grounding in economics when Jan Tinbergen and Ragnar Nurkse gave the model a new home in the 1960s. To this very day, the model's usefulness is unparalleled and it turns out to be one of the few economic models that really fits well empirically and as it turns out, it also has some deeply ingrained underpinnings in economics, specifically in Ricardian models, Hecksher-Olin and increasing returns to scale new trade models (Bergstrand, 1990; Leamer, 1992; Deardorff, 1998; Eaton and Kortum, 2002)

$$T_{ij} = \frac{Y_i Y_j}{D_{ij}}$$

T_{ij} represents trade flows between country j and country i

Y_i and Y_j represent the respective GDPs of countries i and j

D_{ij} represents the distance between country i and j

The model thereby posits that countries with larger GDP and smaller distances between them will engage in more trade. This simple model has been further extended to include other variables that capture other aspects that could affect trade volumes between countries that would not have been included in the traditional model. Its accuracy in predictions of trade patterns is what has been its biggest strengths and the reason why it is so widely used.

MODEL SPECIFICATION

Trade data collected across different countries over time lends itself well to panel data estimation due to its ability to capture both cross-sectional and time-series variations simultaneously. Panel data techniques offer several advantages over traditional cross-sectional or time-series analyses. By incorporating data from multiple countries and time periods, panel data models increase the efficiency and reliability of estimates, allowing for better control of unobserved heterogeneity and time-specific effects. In this paper, the main models we will be estimating are the pooled model, the fixed effect model, and the random effects model.

The pooled model is typically the best starting point. It assumes homogeneity and attempts to estimate the effect of various variables on trade. In the event where there is evidence of homogeneity the estimates from this model may be biased.

The random effects model is usually a good alternative when there is evidence of heterogeneity. This model accounts for possible heterogeneity by varying the intercept on a country specific basis. Its downside is that it assumes that unobserved heterogeneity is uncorrelated with the included explanatory variables.

The fixed effects model serves as an improvement over the random effects model if the unobserved heterogeneity is correlated with the included explanatory variables. It however is unable to estimate the effect of time varying variables and may not be as efficient as the random effects model.

Further, the country pairs lead to the possibility of correlation at the country level. This structure is undesirable for the classical assumptions of the classical linear regression model. We shall account for this by clustering standard errors for each country pair since this has the potential to improve inference.

VARIABLE DESCRIPTION

The table below gives a description of the variables included in the models that will be estimated.

Variable name	**Description**
trade	sum of logged exports and imports, bilateral trade flow
gdp	sum of the logged real GDP
sim	measure of similarity between two trading countries
rlf	measure of relative factor endowments
rer	logged bilateral real exchange rate
cee	Indicator for a country in the European Community
emu	Indicator for countries that have adopted a common currency
dist	geographical distance between capital cities
bor	Indicator for trading partners with a common border
lan	Indicator for trading partners that speak a common language

ESTIMATION RESULTS

The table below shows the results of our gravity model using the three different earlier mentioned techniques, the pooled model, the fixed effects model and the random effects model.

Model Comparison

	Pooled	FE	RE
Dep. Variable	trade	trade	trade
Estimator	PooledOLS	PanelOLS	RandomEffects
No. Observations	3822	3822	3822
Cov. Est.	Clustered	Clustered	Clustered
R-squared	0.9049	0.9513	0.8975
R-Squared (Within)	0.8881	0.5921	0.8977
R-Squared (Between)	0.9098	0.9622	0.8904
R-Squared (Overall)	0.9049	0.9513	0.8921
F-statistic	4027.9	1.241e+04	3707.7
P-value (F-stat)	0.0000	0.0000	0.0000
const	-10.947*		-13.930*
	(-7.9226)		(-9.4027)
gdp	1.5792*	0.6745*	1.7949*
	(24.707)	(15.205)	(27.368)
sim	0.8849*	0.3878*	1.1426*
	(9.9992)	(2.3901)	(6.6144)
rlf	0.0317	-0.4839*	0.0334*
	(0.8375)	(-7.3599)	(2.3947)
rer	0.0987*	0.1103*	0.0690*
	(5.8459)	(3.1231)	(2.3864)
cee	0.3178*	1.3237*	0.3182*
	(4.1036)	(10.632)	(7.4602)
emu	0.2043*	0.2467	0.0927*
	(3.2848)	(1.6957)	(2.0899)
dist	-0.6456*		-0.5909*
	(-4.4816)		(-3.6628)
bor	0.5247*		0.4415
	(2.7227)		(1.7575)
lan	0.2336		0.4172
	(1.2184)		(1.9403)

T-stats reported in parentheses; * represents 5% significance

As can be seen from the table above, the size of a country's GDP is positively related to the amount of trade it engages in and this effect is statistically significant at the 5% level across all model. Similar countries also tend to engage in more trade as their similarity increases. This effect is positive and statistically significant across all models although the impact is smaller in the case of the fixed effects model. The effect of relative factor endowment is ambiguous since it is significant in only the fixed effects and the random effects models but has opposite signs. The bilateral real exchange rate also shows that as the exchange rate increases, more trade will occur and this is significant across all three models. Countries that both belong to the European Community have a statistically significant higher trade among themselves as opposed to otherwise. Having a common currency is associated with higher trade than otherwise in the pooled and random effects model where it is statistically significant but this effect is not statistically significant in the fixed effects model. Having a common border only has a statistically significant positive relationship with trade in the pooled model. Having a common language which is typically included as a proxy for cultural similarities does not have a statistically significant relationship across all three models.

CONCLUSION

The gravity model, as estimated through pooled, fixed effects, and random effects models, reveals interesting insights regarding the relationship between various factors and bilateral trade flows within the European Union. The findings underscore the importance of GDP size and similarity between countries in driving trade volumes, with real exchange rates also playing a significant role. Also, membership in the European Community and sharing a common currency positively impact trade, though some models showed the effect to be insignificant. Overall, these results reinforce the gravity model's empirical robustness and its usefulness in understanding trade dynamics within regional contexts like that of the European Union.

appendix

April 2, 2024

[1]:
```
# Importing relevant libraries
import pandas as pd
import statsmodels.api as sm
from linearmodels.panel import PooledOLS
from linearmodels.panel import RandomEffects
from linearmodels.panel import PanelOLS
from linearmodels.panel import compare
```

[2]:
```
# Importing data
data = pd.read_csv('ss-data.txt',sep='\s+',header=None)
data
```

[2]:
```
              0        1         2         3         4        5         6
0     1960.0000   1.0000  2.927572  12.07024 -0.722535  4.08366 -4.204118
1        0.0000   0.0000  6.816736   0.00000  0.000000  3.28350  3.272900
2       12.6399  -1.3927  8.052400  -1.53520       NaN      NaN       NaN
3     1961.0000   1.0000  2.945885  12.12012 -0.722588  3.38687 -4.193465
4        0.0000   0.0000  6.816736   0.00000  0.000000  3.32130  3.407600
…            …        …         …         …         …        …
11461    1.0000   0.0000  7.267525   0.00000  1.000000  4.48550  6.110400
11462   13.9733  -1.2471  8.632900  -0.95490       NaN      NaN       NaN
11463 2001.0000  91.0000  6.985576  14.29613 -1.245710  9.12933  2.786375
11464    1.0000   0.0000  7.267525   0.00000  1.000000  4.44840  6.096500
11465   13.9851  -1.2521  8.523000  -1.00510       NaN      NaN       NaN

[11466 rows x 7 columns]
```

[3]:
```
# Formatting the data
all_rows = []

counter = 0
for j in range(int(len(data)/3)):
    row = []
    for i in range(3):
        row.extend(data.iloc[counter +i])

    counter += 3
```

```
    all_rows.append(row)

dataset = pd.DataFrame(all_rows)
dataset
```

[3]:

```
           0     1         2         3         4         5         6    7  \
0     1960.0   1.0  2.927572  12.07024 -0.722535  4.083660 -4.204118  0.0
1     1961.0   1.0  2.945885  12.12012 -0.722588  3.386870 -4.193465  0.0
2     1962.0   1.0  3.023661  12.16015 -0.727201  5.849378 -4.181783  0.0
3     1963.0   1.0  2.986535  12.20124 -0.727539  5.928937 -4.169314  0.0
4     1964.0   1.0  3.081419  12.26458 -0.729358  6.144835 -4.156189  0.0
…        …    …        …         …         …         …        …   …
3817  1997.0  91.0  7.139905  14.18990 -1.257982  8.950009  3.118567  1.0
3818  1998.0  91.0  7.173138  14.21853 -1.255940  8.996798  3.104255  1.0
3819  1999.0  91.0  7.151108  14.24670 -1.242226  9.082217  3.064478  1.0
3820  2000.0  91.0  7.091919  14.27779 -1.240669  9.113317  2.931588  1.0
3821  2001.0  91.0  6.985576  14.29613 -1.245710  9.129330  2.786375  1.0

        8         9  …   11      12      13       14      15      16  \
0     0.0  6.816736  …  0.0  3.2835  3.2729  12.6399 -1.3927  8.0524
1     0.0  6.816736  …  0.0  3.3213  3.4076  12.6952 -1.3874  8.0404
2     0.0  6.816736  …  0.0  3.3181  3.5134  12.7435 -1.3917  8.1452
3     0.0  6.816736  …  0.0  3.3159  3.6078  12.7893 -1.3884  8.1614
4     0.0  6.816736  …  0.0  3.3266  3.7607  12.8565 -1.3871  8.2114
…   …        …  …  …   …      …       …      …       …
3817  0.0  7.267525  …  1.0  4.7184  6.0167  13.8661 -1.2746  8.5992
3818  0.0  7.267525  …  1.0  4.6709  6.1389  13.8963 -1.2668  8.6213
3819  0.0  7.267525  …  1.0  4.6139  6.1737  13.9341 -1.2563  8.6397
3820  0.0  7.267525  …  1.0  4.4855  6.1104  13.9733 -1.2471  8.6329
3821  0.0  7.267525  …  1.0  4.4484  6.0965  13.9851 -1.2521  8.5230

          17  18  19  20
0    -1.5352 NaN NaN NaN
1    -1.5130 NaN NaN NaN
2    -1.4990 NaN NaN NaN
3    -1.4865 NaN NaN NaN
4    -1.4734 NaN NaN NaN
…        …  ..  ..  ..
3817 -0.6152 NaN NaN NaN
3818 -0.6500 NaN NaN NaN
3819 -0.7173 NaN NaN NaN
3820 -0.9549 NaN NaN NaN
3821 -1.0051 NaN NaN NaN

[3822 rows x 21 columns]
```

```
[4]: # Removing colums with NaNs
     dataset = dataset.iloc[:,:18]
     dataset
```

```
[4]:          0     1         2         3         4         5         6    7  \
     0     1960.0   1.0  2.927572  12.07024 -0.722535  4.083660 -4.204118  0.0
     1     1961.0   1.0  2.945885  12.12012 -0.722588  3.386870 -4.193465  0.0
     2     1962.0   1.0  3.023661  12.16015 -0.727201  5.849378 -4.181783  0.0
     3     1963.0   1.0  2.986535  12.20124 -0.727539  5.928937 -4.169314  0.0
     4     1964.0   1.0  3.081419  12.26458 -0.729358  6.144835 -4.156189  0.0
     …        …     …        …         …         …        …         …    …
     3817  1997.0  91.0  7.139905  14.18990 -1.257982  8.950009  3.118567  1.0
     3818  1998.0  91.0  7.173138  14.21853 -1.255940  8.996798  3.104255  1.0
     3819  1999.0  91.0  7.151108  14.24670 -1.242226  9.082217  3.064478  1.0
     3820  2000.0  91.0  7.091919  14.27779 -1.240669  9.113317  2.931588  1.0
     3821  2001.0  91.0  6.985576  14.29613 -1.245710  9.129330  2.786375  1.0

             8         9   10   11       12       13       14       15       16       17
     0     0.0  6.816736  0.0  0.0  3.2835  3.2729  12.6399 -1.3927  8.0524 -1.5352
     1     0.0  6.816736  0.0  0.0  3.3213  3.4076  12.6952 -1.3874  8.0404 -1.5130
     2     0.0  6.816736  0.0  0.0  3.3181  3.5134  12.7435 -1.3917  8.1452 -1.4990
     3     0.0  6.816736  0.0  0.0  3.3159  3.6078  12.7893 -1.3884  8.1614 -1.4865
     4     0.0  6.816736  0.0  0.0  3.3266  3.7607  12.8565 -1.3871  8.2114 -1.4734
     …     …        …    …    …      …       …       …        …        …      …
     3817  0.0  7.267525  0.0  1.0  4.7184  6.0167  13.8661 -1.2746  8.5992 -0.6152
     3818  0.0  7.267525  0.0  1.0  4.6709  6.1389  13.8963 -1.2668  8.6213 -0.6500
     3819  0.0  7.267525  0.0  1.0  4.6139  6.1737  13.9341 -1.2563  8.6397 -0.7173
     3820  0.0  7.267525  0.0  1.0  4.4855  6.1104  13.9733 -1.2471  8.6329 -0.9549
     3821  0.0  7.267525  0.0  1.0  4.4484  6.0965  13.9851 -1.2521  8.5230 -1.0051

     [3822 rows x 18 columns]
```

```
[5]: # Naming columns as per the data description
     column_names =␣
      ↪['year','country','trade','gdp','sim','rlf','rer','cee','emu','dist','bor','lan','rert','ft
     dataset.columns = column_names
     dataset
```

```
[5]:          year  country     trade       gdp       sim       rlf       rer  cee  \
     0     1960.0      1.0  2.927572  12.07024 -0.722535  4.083660 -4.204118  0.0
     1     1961.0      1.0  2.945885  12.12012 -0.722588  3.386870 -4.193465  0.0
     2     1962.0      1.0  3.023661  12.16015 -0.727201  5.849378 -4.181783  0.0
     3     1963.0      1.0  2.986535  12.20124 -0.727539  5.928937 -4.169314  0.0
     4     1964.0      1.0  3.081419  12.26458 -0.729358  6.144835 -4.156189  0.0
     …        …        …        …         …         …        …         …    …
     3817  1997.0     91.0  7.139905  14.18990 -1.257982  8.950009  3.118567  1.0
     3818  1998.0     91.0  7.173138  14.21853 -1.255940  8.996798  3.104255  1.0
```

```
3819  1999.0     91.0  7.151108  14.24670 -1.242226  9.082217  3.064478  1.0
3820  2000.0     91.0  7.091919  14.27779 -1.240669  9.113317  2.931588  1.0
3821  2001.0     91.0  6.985576  14.29613 -1.245710  9.129330  2.786375  1.0

      emu      dist  bor  lan    rert  ftrade     fgdp    fsim    frlf    frer
0     0.0  6.816736  0.0  0.0  3.2835  3.2729  12.6399 -1.3927  8.0524 -1.5352
1     0.0  6.816736  0.0  0.0  3.3213  3.4076  12.6952 -1.3874  8.0404 -1.5130
2     0.0  6.816736  0.0  0.0  3.3181  3.5134  12.7435 -1.3917  8.1452 -1.4990
3     0.0  6.816736  0.0  0.0  3.3159  3.6078  12.7893 -1.3884  8.1614 -1.4865
4     0.0  6.816736  0.0  0.0  3.3266  3.7607  12.8565 -1.3871  8.2114 -1.4734
…     …         …    …    …       …       …        …       …       …
3817  0.0  7.267525  0.0  1.0  4.7184  6.0167  13.8661 -1.2746  8.5992 -0.6152
3818  0.0  7.267525  0.0  1.0  4.6709  6.1389  13.8963 -1.2668  8.6213 -0.6500
3819  0.0  7.267525  0.0  1.0  4.6139  6.1737  13.9341 -1.2563  8.6397 -0.7173
3820  0.0  7.267525  0.0  1.0  4.4855  6.1104  13.9733 -1.2471  8.6329 -0.9549
3821  0.0  7.267525  0.0  1.0  4.4484  6.0965  13.9851 -1.2521  8.5230 -1.0051

[3822 rows x 18 columns]
```

[6]:
```python
# Formatting the data for panel estimation
dataset = dataset[['country','year','trade','gdp','sim','rlf','rer','cee','emu','dist','bor','lan','r
dataset.year = dataset.year.astype(int)
dataset.set_index(['country', 'year'],inplace=True)
dataset
```

[6]:
```
                 trade       gdp       sim       rlf       rer  cee  emu  \
country year
1.0     1960  2.927572  12.07024 -0.722535  4.083660 -4.204118  0.0  0.0
        1961  2.945885  12.12012 -0.722588  3.386870 -4.193465  0.0  0.0
        1962  3.023661  12.16015 -0.727201  5.849378 -4.181783  0.0  0.0
        1963  2.986535  12.20124 -0.727539  5.928937 -4.169314  0.0  0.0
        1964  3.081419  12.26458 -0.729358  6.144835 -4.156189  0.0  0.0
…                  …         …         …         …         …    …    …
91.0    1997  7.139905  14.18990 -1.257982  8.950009  3.118567  1.0  0.0
        1998  7.173138  14.21853 -1.255940  8.996798  3.104255  1.0  0.0
        1999  7.151108  14.24670 -1.242226  9.082217  3.064478  1.0  0.0
        2000  7.091919  14.27779 -1.240669  9.113317  2.931588  1.0  0.0
        2001  6.985576  14.29613 -1.245710  9.129330  2.786375  1.0  0.0

                  dist  bor  lan    rert  ftrade     fgdp    fsim    frlf  \
country year
1.0     1960  6.816736  0.0  0.0  3.2835  3.2729  12.6399 -1.3927  8.0524
        1961  6.816736  0.0  0.0  3.3213  3.4076  12.6952 -1.3874  8.0404
        1962  6.816736  0.0  0.0  3.3181  3.5134  12.7435 -1.3917  8.1452
        1963  6.816736  0.0  0.0  3.3159  3.6078  12.7893 -1.3884  8.1614
        1964  6.816736  0.0  0.0  3.3266  3.7607  12.8565 -1.3871  8.2114
```

```
…                    …   …   …     …     …     …      …      …
91.0     1997  7.267525  0.0  1.0  4.7184  6.0167  13.8661 -1.2746  8.5992
         1998  7.267525  0.0  1.0  4.6709  6.1389  13.8963 -1.2668  8.6213
         1999  7.267525  0.0  1.0  4.6139  6.1737  13.9341 -1.2563  8.6397
         2000  7.267525  0.0  1.0  4.4855  6.1104  13.9733 -1.2471  8.6329
         2001  7.267525  0.0  1.0  4.4484  6.0965  13.9851 -1.2521  8.5230

                 frer
country year
1.0      1960 -1.5352
         1961 -1.5130
         1962 -1.4990
         1963 -1.4865
         1964 -1.4734
…                …
91.0     1997 -0.6152
         1998 -0.6500
         1999 -0.7173
         2000 -0.9549
         2001 -1.0051

[3822 rows x 16 columns]
```

[11]:

```python
# Pooled regression model
exog_vars = ['gdp','sim','rlf','rer','cee','emu','dist','bor','lan']
exog = sm.add_constant(dataset[exog_vars])
pooled_model = PooledOLS(dataset.trade,exog).fit(cov_type='clustered',
 ↪cluster_entity=True)
print(pooled_model)
```

```
                          PooledOLS Estimation Summary
================================================================================
Dep. Variable:                  trade   R-squared:                        0.9049
Estimator:                  PooledOLS   R-squared (Between):              0.9098
No. Observations:                3822   R-squared (Within):               0.8881
Date:                Tue, Apr 02 2024   R-squared (Overall):              0.9049
Time:                        12:44:44   Log-likelihood                   -3379.3
Cov. Estimator:             Clustered
                                        F-statistic:                      4027.9
Entities:                          91   P-value                           0.0000
Avg Obs:                       42.000   Distribution:                  F(9,3812)
Min Obs:                       42.000
Max Obs:                       42.000   F-statistic (robust):             185.72
                                        P-value                           0.0000
Time periods:                      42   Distribution:                  F(9,3812)
Avg Obs:                       91.000
Min Obs:                       91.000
```

```
Max Obs:                       91.000

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
const          -10.947     1.3818    -7.9226     0.0000     -13.657     -8.2383
gdp             1.5792     0.0639     24.707     0.0000      1.4539      1.7045
sim             0.8849     0.0885     9.9992     0.0000      0.7114      1.0584
rlf             0.0317     0.0379     0.8375     0.4024     -0.0425      0.1060
rer             0.0987     0.0169     5.8459     0.0000      0.0656      0.1317
cee             0.3178     0.0775     4.1036     0.0000      0.1660      0.4697
emu             0.2043     0.0622     3.2848     0.0010      0.0824      0.3263
dist           -0.6456     0.1440    -4.4816     0.0000     -0.9280     -0.3632
bor             0.5247     0.1927     2.7227     0.0065      0.1469      0.9026
lan             0.2336     0.1917     1.2184     0.2231     -0.1423      0.6096
==============================================================================
```

[12]:

```python
# Fixed effects regression model
fixed_effects_model = PanelOLS(dataset.trade,␣
 ↪dataset[['gdp','sim','rlf','rer','cee','emu']]).fit(cov_type='clustered',␣
 ↪cluster_entity=True)
print(fixed_effects_model)
```

```
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:                  trade   R-squared:                        0.9513
Estimator:                   PanelOLS   R-squared (Between):              0.9622
No. Observations:                3822   R-squared (Within):               0.5921
Date:                Tue, Apr 02 2024   R-squared (Overall):              0.9513
Time:                        12:44:55   Log-likelihood                   -5987.3
Cov. Estimator:             Clustered
                                        F-statistic:                   1.241e+04
Entities:                          91   P-value                           0.0000
Avg Obs:                       42.000   Distribution:                  F(6,3816)
Min Obs:                       42.000
Max Obs:                       42.000   F-statistic (robust):             518.06
                                        P-value                           0.0000
Time periods:                      42   Distribution:                  F(6,3816)
Avg Obs:                       91.000
Min Obs:                       91.000
Max Obs:                       91.000

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
gdp             0.6745     0.0444     15.205     0.0000      0.5875      0.7615
```

```
sim             0.3878     0.1623     2.3901     0.0169     0.0697     0.7060
rlf            -0.4839     0.0658    -7.3599     0.0000    -0.6129    -0.3550
rer             0.1103     0.0353     3.1231     0.0018     0.0411     0.1796
cee             1.3237     0.1245     10.632     0.0000     1.0796     1.5678
emu             0.2467     0.1455     1.6957     0.0900    -0.0385     0.5320
==============================================================================
```

[14]:
```python
# Random effects regression model
random_effects_model = RandomEffects(dataset.trade,exog).
 ↪fit(cov_type='clustered', cluster_entity=True)
print(random_effects_model)
```

```
                        RandomEffects Estimation Summary
================================================================================
Dep. Variable:                  trade   R-squared:                        0.8975
Estimator:              RandomEffects   R-squared (Between):              0.8904
No. Observations:                3822   R-squared (Within):               0.8977
Date:                Tue, Apr 02 2024   R-squared (Overall):              0.8921
Time:                        12:48:33   Log-likelihood                   -734.59
Cov. Estimator:             Clustered
                                        F-statistic:                      3707.7
Entities:                          91   P-value                           0.0000
Avg Obs:                       42.000   Distribution:                  F(9,3812)
Min Obs:                       42.000
Max Obs:                       42.000   F-statistic (robust):             241.42
                                        P-value                           0.0000
Time periods:                      42   Distribution:                  F(9,3812)
Avg Obs:                       91.000
Min Obs:                       91.000
Max Obs:                       91.000

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
const         -13.930     1.4815    -9.4027     0.0000    -16.835     -11.026
gdp            1.7949     0.0656     27.368     0.0000     1.6663      1.9234
sim            1.1426     0.1728     6.6144     0.0000     0.8039      1.4813
rlf            0.0334     0.0140     2.3947     0.0167     0.0061      0.0608
rer            0.0690     0.0289     2.3864     0.0171     0.0123      0.1257
cee            0.3182     0.0427     7.4602     0.0000     0.2346      0.4018
emu            0.0927     0.0443     2.0899     0.0367     0.0057      0.1796
dist          -0.5909     0.1613    -3.6628     0.0003    -0.9072     -0.2746
bor            0.4415     0.2512     1.7575     0.0789    -0.0510      0.9339
lan            0.4172     0.2150     1.9403     0.0524    -0.0044      0.8387
==============================================================================
```

[15]:
```python
# Comparison from the various estimation techniques
print(compare({'Pooled': pooled_model,
               'FE':fixed_effects_model,
              'RE': random_effects_model,
              }))
```

```
                       Model Comparison
=============================================================
                             Pooled           FE           RE
-------------------------------------------------------------
Dep. Variable                 trade        trade        trade
Estimator                 PooledOLS     PanelOLS RandomEffects
No. Observations               3822         3822         3822
Cov. Est.                 Clustered    Clustered    Clustered
R-squared                    0.9049       0.9513       0.8975
R-Squared (Within)           0.8881       0.5921       0.8977
R-Squared (Between)          0.9098       0.9622       0.8904
R-Squared (Overall)          0.9049       0.9513       0.8921
F-statistic                  4027.9    1.241e+04       3707.7
P-value (F-stat)             0.0000       0.0000       0.0000
===================== =========== =========== ===============
const                       -10.947                    -13.930
                          (-7.9226)                  (-9.4027)
gdp                          1.5792       0.6745       1.7949
                           (24.707)     (15.205)     (27.368)
sim                          0.8849       0.3878       1.1426
                           (9.9992)     (2.3901)     (6.6144)
rlf                          0.0317      -0.4839       0.0334
                           (0.8375)    (-7.3599)     (2.3947)
rer                          0.0987       0.1103       0.0690
                           (5.8459)     (3.1231)     (2.3864)
cee                          0.3178       1.3237       0.3182
                           (4.1036)     (10.632)     (7.4602)
emu                          0.2043       0.2467       0.0927
                           (3.2848)     (1.6957)     (2.0899)
dist                        -0.6456                    -0.5909
                          (-4.4816)                  (-3.6628)
bor                          0.5247                     0.4415
                           (2.7227)                   (1.7575)
lan                          0.2336                     0.4172
                           (1.2184)                   (1.9403)
-------------------------------------------------------------

T-stats reported in parentheses
```