Replicating an Empirical Example of International Trade

July 31, 2018

Abstract

alpaca is an R package providing the Newton-Raphson pseudo-demeaning algorithm proposed by Stammann (2018). To introduce the usage, we replicate parts of a paper by Glick & Rose (2016) who investigate the effect of currency unions on trade flows using a data set of about 200 countries over 65 years. The replication is inspired by a recent working paper of Larch et al. (2017).

1 Introduction

In econometrics, fixed effects models are popular to control for unobserved heterogeneity in data sets with a panel like structure. In non-linear models this is usually done by including a dummy variable for each level of a fixed effects category. This approach can quickly become infeasible if the number of levels and/or fixed effects categories increases (either due to memory or time limitations).

One well known example where those limits take place are so called structural gravity models of trade. In this replication example we will use data of 879,794 bilateral trade flows. Estimating a theory consistent gravity model with this data set requires to specify a three-way error component with roughly 55,000 levels.

The replication is organized as follows. First we show how to prepare the data. Afterwards we replicate table 5 of Glick & Rose (2016) using a poisson model instead of a linear. Finally we replicate table 6 and demonstrate how to perform joint hypotheses tests.

2 Preparing the Data Set

The data set is available either from Andrew Rose's website[1] or from *sciencedirect*[2]. We use the same variable names as Glick & Rose (2016) so that we can compare summary statistics with the one provided in their *Stata* log-files. This ensures that we use the same data set.

```
# Import the data set
library(haven)
library(data.table)
cudata <- read_dta("appl/dataaxj1.dta")
cudata <- as.data.table(cudata)

# Subsetting relevant variables
var.nms <- c("exp1to2", "custrict11", "ldist", "comlang", "border", "regional",
    "comcol", "curcol", "colony", "comctry", "cuwoemu", "emu", "cuc", "cty1",
    "cty2", "year", "pairid")
cudata <- cudata[, ..var.nms]

# Generate identifiers required for structural gravity
```

[1] http://faculty.haas.berkeley.edu/arose/RecRes.htm
[2] https://www.sciencedirect.com/science/article/pii/S0014292116300630#ec0005

```r
cudata[, `:=`(pairid, factor(pairid))]
cudata[, `:=`(exp.time, interaction(cty1, year))]
cudata[, `:=`(imp.time, interaction(cty2, year))]

# Generate dummies for disaggregated currency unions
cudata[, `:=`(cuau, as.integer(cuc == "au"))]
cudata[, `:=`(cube, as.integer(cuc == "be"))]
cudata[, `:=`(cuca, as.integer(cuc == "ca"))]
cudata[, `:=`(cucf, as.integer(cuc == "cf"))]
cudata[, `:=`(cucp, as.integer(cuc == "cp"))]
cudata[, `:=`(cudk, as.integer(cuc == "dk"))]
cudata[, `:=`(cuea, as.integer(cuc == "ea"))]
cudata[, `:=`(cuec, as.integer(cuc == "ec"))]
cudata[, `:=`(cuem, as.integer(cuc == "em"))]
cudata[, `:=`(cufr, as.integer(cuc == "fr"))]
cudata[, `:=`(cugb, as.integer(cuc == "gb"))]
cudata[, `:=`(cuin, as.integer(cuc == "in"))]
cudata[, `:=`(cuma, as.integer(cuc == "ma"))]
cudata[, `:=`(cuml, as.integer(cuc == "ml"))]
cudata[, `:=`(cunc, as.integer(cuc == "nc"))]
cudata[, `:=`(cunz, as.integer(cuc == "nz"))]
cudata[, `:=`(cupk, as.integer(cuc == "pk"))]
cudata[, `:=`(cupt, as.integer(cuc == "pt"))]
cudata[, `:=`(cusa, as.integer(cuc == "sa"))]
cudata[, `:=`(cusp, as.integer(cuc == "sp"))]
cudata[, `:=`(cuua, as.integer(cuc == "ua"))]
cudata[, `:=`(cuus, as.integer(cuc == "us"))]
cudata[, `:=`(cuwa, as.integer(cuc == "wa"))]
cudata[, `:=`(cuwoo, custrict11)]
cudata[cuc %in% c("em", "au", "cf", "ec", "fr", "gb", "in", "us"), `:=`(cuwoo,
    0L)]

# Set missing trade flows to zero
cudata[is.na(exp1to2), `:=`(exp1to2, 0)]

# Generate lead and lags of any currency union entry and exit
setkey(cudata, pairid, year)
cudata[, `:=`(lcu11, shift(custrict11)), by = pairid]
cudata[, `:=`(tranex, as.integer(!is.na(lcu11) & custrict11 - lcu11 == -1))]
cudata[, `:=`(tranen, as.integer(!is.na(lcu11) & custrict11 - lcu11 == 1))]
for (i in seq(14L)) {
    cudata[, `:=`(paste0("acuex", i), shift(tranex, i, 0L, "lag")), by = pairid]
    cudata[, `:=`(paste0("acuen", i), shift(tranen, i, 0L, "lag")), by = pairid]
    cudata[, `:=`(paste0("bcuex", i), shift(tranex, i, 0L, "lead")), by = pairid]
    cudata[, `:=`(paste0("bcuen", i), shift(tranen, i, 0L, "lead")), by = pairid]
}

# Generate lead and lags for EMU entry
cudata[, `:=`(lemu, shift(emu)), by = pairid]
cudata[, `:=`(tranemuen, as.integer(!is.na(lemu) & emu - lemu == 1 & comctry ==
    0))]
# Guadeloupe, French Guiana, Reunion, and Martinique
cudata[pairid == shift(pairid) & custrict11 == 1L & lcu11 == 1L & comctry ==
    0 & tranemuen != 0L, `:=`(tranemuen, 0L)]
```

```r
for (i in seq(14L)) {
    cudata[, `:=`(paste0("aemuen", i), shift(tranemuen, i, 0L, "lag")), by = pairid]
    cudata[, `:=`(paste0("bemuen", i), shift(tranemuen, i, 0L, "lead")), by = pairid]
}

# Generate lead and lags for non-EMU entry
cudata[, `:=`(trannoeen, as.integer(!is.na(lcu11) & custrict11 - lcu11 == 1))]
cudata[tranemuen == 1L, `:=`(trannoeen, 0L)]
for (i in seq(14L)) {
    cudata[, `:=`(paste0("anoeen", i), shift(trannoeen, i, 0L, "lag")), by = pairid]
    cudata[, `:=`(paste0("bnoeen", i), shift(trannoeen, i, 0L, "lead")), by = pairid]
}

# Remove auxiliary variables
cudata[, `:=`(c("lemu", "lcu11"), NULL)]
```

3 Replicating Table 5

Glick & Rose (2016) propose the following "theory consistent" gravity specification

$$x_{ijt} = \exp\left(\gamma \text{cu}_{ijt} + \mathbf{z}'_{ijt}\boldsymbol{\beta} + \lambda_{it} + \psi_{jt} + \phi_{ij}\right) \epsilon_{ijt} \,,$$

where x_{ijt} denotes the nominal value of bilateral export from i to j at time t, cu_{ijt} is one if i and j use the same currency at time t, and $\mathbf{z}_{ijt}$ is a vector of further control variables. λ_{it}, ψ_{jt}, and ϕ_{ij} are three complete sets of time-varying exporter, time-varying importer, and dyadic fixed effects.

As Glick & Rose (2016) we are mainly interested in how being in a currency union (CU) affects the export flows (γ). We investigate three different measures: all CUs aggregated, distinguish between European Monetary Union (EMU) and others, and further distinguish the others. Table 1 shows poisson gravity estimates for the six different model specifications. Usually the point estimates are interpreted as semi elasticities. For instance, being in the EMU raises exports by roughly three percent (statistically insignificantly different from zero) if we look at the estimates in column 5 and 6. Overall the results differ significantly from the ones reported by Glick & Rose (2016) as already mentioned by Larch et al. (2017).

Table 1: Panel gravity estimates for bilateral exports

Fixed Effects:	λ_{it} and ψ_{jt}			λ_{it}, ψ_{jt}, and ϕ_{ij}		
	All CUs	Disagg. EMU	Disagg. CUs	All CUs	Disagg. EMU	Disagg. CUs
All CUs	- 0.13 (0.07)			0.14 (0.04)		
EMU		- 0.20 (0.07)	- 0.19 (0.07)		0.03 (0.04)	0.03 (0.04)

Further control variables: log distance, common language common land border, regional FTA membership, common colonizer, current colony/colonizer, ever colony/colonizer, common country. Clustered standard errors in parentheses (by distance).

Next we show how to replicate the results in column 6 using *alpaca*.

```r
# Three fixed effects categories (exporter x year, importer x year, dyadic)
# Disaggregated Currency Unions
formula <- exp1to2 ~ emu + cuwoo + cuau + cucf + cuec + cufr + cugb + cuin +
    cuus + regional + curcol | exp.time + imp.time + pairid
mod <- feglm(formula, cudata[exp1to2 > 0], family = poisson())
```

```
# Summarize estimates (standard errors clustered by distance)
summary(mod, "clustered", "pairid")
```

4 Replicating Table 6

Glick & Rose (2016) are also interested to test whether the symmetry assumption they imposed in an earlier paper is justified by the data. Therefore they estimate the following unrestricted model

$$x_{ijt} = \exp\left(\sum_{k=-14}^{14} \theta_k \text{cuentry}_{ij(t-k)} + \sum_{k=-14}^{14} \varphi_k \text{cuexit}_{ij(t-k)} + \mathbf{z}'_{ijt}\boldsymbol{\beta} + \lambda_{it} + \psi_{jt} + \phi_{ij}\right) \epsilon_{ijt} \,,$$

where $\text{cuentry}_{ij(t-k)}$ and $\text{cuexit}_{ij(t-k)}$ are one if i and j entered or exited a currency union at time $t-k$ respectively. Afterwards they test jointly for symmetry between the effects of entry and/or exit.

Table 2 shows different Wald-tests based on poisson gravity estimates. Contrary to Glick & Rose (2016) we reject the null in almost all cases, questioning the validity of the symmetry assumption.

Table 2: Symmetry tests for bilateral exports

Fixed Effects:	λ_{it} and ψ_{jt}	λ_{it}, ψ_{jt}, and ϕ_{ij}
After any CU Entry = - After any CU Exit?	28.6 (0.01)	18.7 (0.18)
After any CU Entry = - After any CU Exit?	46.1 (0.00)	39.6 (0.00)
Both	79.7 (0.00)	60.0 (0.00)
After any CU Entry = - After any CU Exit?	35.2 (0.00)	31.7 (0.00)
After any CU Entry = - After any CU Exit?	44.2 (0.00)	63.8 (0.00)
Both	91.9 (0.00)	102.0 (0.00)
After any CU Entry = - After any CU Exit?	32.4 (0.00)	20.8 (0.11)

Further control variables: log distance, common language common land border, regional FTA membership, common colonizer, current colony/colonizer, ever colony/colonizer, common country. Reported Wald-statistic and p-values in parentheses. Computation based on clustered covariance estimates (by distance).

Next we show how to replicate some of the results in table 2 using *alpaca*.

```
# Wald-test
wald.test <- function(R, theta, q, V) {
    V.wald <- R %*% V %*% t(R)
    lr <- R %*% theta - q
    W <- as.double(t(lr) %*% solve(V.wald) %*% lr)
    list(W = W, p.value = 1 - pchisq(W, nrow(R)))
}

# Three fixed effects categories (exporter x year, importer x year, dyadic)
# Any CU
formula1.3way <- as.formula(paste0("exp1to2~", paste0("bcuen", seq(14L), collapse = "+"),
    "+tranen+", paste0("acuen", seq(14L), collapse = "+"), "+", paste0("bcuex",
        seq(14L), collapse = "+"), "+tranex+", paste0("acuex", seq(14L), collapse = "+"),
    "+regional+curcol|exp.time+imp.time+pairid"))
mod <- feglm(formula1.3way, cudata[exp1to2 > 0], family = poisson())
theta1.3way <- coef(mod)
V1.3way <- vcov(mod, "clustered", cluster.vars = "pairid")
```

```r
rm(mod)

# After any CU Entry = - After any CU Exit?
R1.3way <- matrix(0, 14L, length(theta1.3way))
colnames(R1.3way) <- names(theta1.3way)
R1.3way[, "tranen"] <- -1
R1.3way[, "tranex"] <- -1
for (i in seq(14L)) {
    R1.3way[i, paste0("acuen", i)] <- 1
    R1.3way[i, paste0("acuex", i)] <- 1
}
q1.3way <- numeric(nrow(R1.3way))
w1.3way <- wald.test(R1.3way, theta1.3way, q1.3way, V1.3way)

# Before any CU Entry = - Before any CU Exit?
R2.3way <- matrix(0, 14L, length(theta1.3way))
colnames(R2.3way) <- names(theta1.3way)
R2.3way[, "tranen"] <- -1
R2.3way[, "tranex"] <- -1
for (i in seq(14L)) {
    R2.3way[i, paste0("bcuen", i)] <- 1
    R2.3way[i, paste0("bcuex", i)] <- 1
}
q2.3way <- numeric(nrow(R2.3way))
w2.3way <- wald.test(R2.3way, theta1.3way, q2.3way, V1.3way)

# Both (any CU)
R3.3way <- rbind(R1.3way, R2.3way)
q3.3way <- numeric(nrow(R3.3way))
w3.3way <- wald.test(R3.3way, theta1.3way, q3.3way, V1.3way)

# EMU entry
formula2.3way <- as.formula(paste0("exp1to2~", paste0("bnoeen", seq(14L), collapse = "+"),
    "+trannoeen+", paste0("anoeen", seq(14L), collapse = "+"), "+", paste0("bcuex",
        seq(14L), collapse = "+"), "+tranex+", paste0("acuex", seq(14L), collapse = "+"),
    "+", paste0("bemuen", seq(14L), collapse = "+"), "+tranemuen+", paste0("aemuen",
        seq(14L), collapse = "+"), "+", "+regional+curcol|exp.time+imp.time+pairid"))
mod <- feglm(formula2.3way, cudata[exp1to2 > 0], family = poisson())
theta2.3way <- coef(mod)
V2.3way <- vcov(mod, "clustered", cluster.vars = "pairid")
rm(mod)

# After non-EMU CU Entry = After EMU Entry?
R4.3way <- matrix(0, 14L, length(theta2.3way))
colnames(R4.3way) <- names(theta2.3way)
R4.3way[, "trannoeen"] <- -1
R4.3way[, "tranemuen"] <- -1
for (i in seq(14L)) {
    R4.3way[i, paste0("anoeen", i)] <- 1
    R4.3way[i, paste0("aemuen", i)] <- 1
}
q4.3way <- numeric(nrow(R4.3way))
w4.3way <- wald.test(R4.3way, theta2.3way, q4.3way, V2.3way)
```

```r
# Before non-EMU CU Entry = Before EMU Entry?
R5.3way <- matrix(0, 14L, length(theta2.3way))
colnames(R5.3way) <- names(theta2.3way)
R5.3way[, "trannoeen"] <- -1
R5.3way[, "tranemuen"] <- -1
for (i in seq(14L)) {
    R5.3way[i, paste0("bnoeen", i)] <- 1
    R5.3way[i, paste0("bemuen", i)] <- 1
}
q5.3way <- numeric(nrow(R5.3way))
w5.3way <- wald.test(R5.3way, theta2.3way, q5.3way, V2.3way)

# Both (EMU Entry)
R6.3way <- rbind(R4.3way, R5.3way)
q6.3way <- numeric(nrow(R6.3way))
w6.3way <- wald.test(R6.3way, theta2.3way, q6.3way, V2.3way)

# After non-EMU CU Exit = - After EMU Entry
R7.3way <- matrix(0, 14L, length(theta2.3way))
colnames(R7.3way) <- names(theta2.3way)
R7.3way[, "tranex"] <- -1
R7.3way[, "tranemuen"] <- -1
for (i in seq(14L)) {
    R7.3way[i, paste0("acuex", i)] <- 1
    R7.3way[i, paste0("aemuen", i)] <- 1
}
q7.3way <- numeric(nrow(R7.3way))
w7.3way <- wald.test(R7.3way, theta2.3way, q7.3way, V2.3way)

# Display test results
W <- round(c(w1.3way$W, w2.3way$W, w3.3way$W, w4.3way$W, w5.3way$W, w6.3way$W,
    w7.3way$W), 1L)
P <- round(c(w1.3way$p.value, w2.3way$p.value, w3.3way$p.value, w4.3way$p.value,
    w5.3way$p.value, w6.3way$p.value, w7.3way$p.value), 1L)
cbind(W, P)
```

References

Glick, R. & Rose, A. K. (2016), 'Currency unions and trade: A post-emu reassessment', *European Economic Review* **87**, 78 – 91.

Larch, M., Wanner, J., Yotov, Y. & Zylkin, T. (2017), 'The currency union effect: A ppml re-assessment with high-dimensional fixed effects', *School of Economics Working Paper Series* .

Stammann, A. (2018), 'Fast and feasible estimation of generalized linear models with high-dimensional k-way fixed effects', *ArXiv e-prints* .