Econometrics in R: Estimation of Gravity Models

Pachá

Dec 15, 2018

satRday Santiago

Before we begin

This is about a new R package

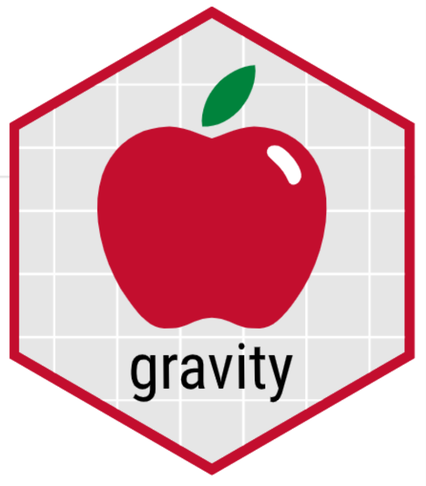

Figure 1: Hex sticker

Where to reach me

or : pachamaltese

: `m vargas at dcc dot uchile dot cl`

: +1 XXX XXX XX XX or +56 X X XXX XX XX

Question to the audience

Do you understand (a bit) linear algebra? Have you ever fitted a regression (in R) before today?

Linear regression

Let $\boldsymbol{y} \in \mathbb{R}^n$ be the outcome and $X \in \mathbb{R}^{n \times p}$ be the design matrix in the context of a general model with intercept:

$$\boldsymbol{y} = X\beta + \boldsymbol{e}$$

Being:

$$\underset{n\times 1}{\boldsymbol{y}} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \text{ and } \underset{n\times p}{X} = \begin{pmatrix} 1 & x_{11} & & x_{1p} \\ 1 & x_{21} & & x_{2p} \\ & & \ddots & \\ 1 & x_{n1} & & x_{np} \end{pmatrix} = (\boldsymbol{1}\ \boldsymbol{x}_1\ \ldots\ \boldsymbol{x}_p)$$

Linear regression

In linear models the aim is to minimize the error term by chosing $\hat{\beta}$. One possibility is to minimize the squared error by solving this optimization problem:

$$\min_{\beta} S = \|\boldsymbol{y} - X\beta\|^2 \tag{1}$$

Linear regression

Books such as Baltagi [2011] discuss how to solve (1) and other equivalent approaches that result in this optimal estimator:

$$\hat{\beta} = (X^t X)^{-1} X^t \boldsymbol{y} \tag{2}$$

Linear regression

With one independent variable and intercept, this is $y_i = \beta_0 + \beta_1 x_{i1} + e_i$, equation (2) means:

$$\hat{\beta}_1 = cor(\boldsymbol{y}, \boldsymbol{x}) \cdot \frac{sd(\boldsymbol{y})}{sd(\boldsymbol{x})} \text{ and } \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{\boldsymbol{x}} \tag{3}$$

Coding example with mtcars dataset

Consider the model:

$$mpg_i = \beta_1 wt_i + \beta_2 cyl_i + e_i$$

This is how to write that model in R notation:

```r
lm(mpg ~ wt + cyl, data = mtcars)
```

Coding example with mtcars dataset

Or written in matrix form:

```
y <- mtcars$mpg
x0 <- rep(1, length(y))
x1 <- mtcars$wt
x2 <- mtcars$cyl
X <- cbind(x0,x1,x2)
```

Coding example with mtcars dataset

It's the same to use `lm` or to perform a matrix multiplication because of equation (2):

```
fit <- lm(y ~ x1 + x2)
beta <- solve(t(X)%*%X) %*% (t(X)%*%y)
```

Coding example with mtcars dataset

It's the same to use `lm` or to perform a matrix multiplication because of equation (2):

```r
coefficients(fit)
```

```
## (Intercept)          x1          x2
##   39.686261   -3.190972   -1.507795
```

```r
beta
```

```
##         [,1]
## x0 39.686261
## x1 -3.190972
## x2 -1.507795
```

Coding example with Galton dataset

Equation (3) can be verified with R commands:

```r
if (!require(pacman)) install.packages("pacman")
p_load(HistData)

y <- Galton$child
x <- Galton$parent
beta1 <- cor(y, x) *  sd(y) / sd(x)
beta0 <- mean(y) - beta1 * mean(x)
c(beta0, beta1)
```

```
## [1] 23.9415302  0.6462906
```

Coding example with Galton dataset

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##     23.9415       0.6463
```

Simple gravity model

Simple gravity model

The main reference for this section is Woelwer et al. [2018] and the references therein.

Gravity models in their traditional form are inspired by Newton law of gravitation:

$$F_{ij} = G \frac{M_i M_j}{D_{ij}^2}.$$

The force F between two bodies i and j with $i \neq j$ is proportional to the masses M of these bodies and inversely proportional to the square of their geographical distance D. G is a constant and as such of no major concern.

Simple gravity model

The underlying idea of a traditional gravity model, shown for international trade, is equally simple:

$$X_{ij} = G \frac{Y_i^{\beta_1} Y_j^{\beta_2}}{D_{ij}^{\beta_3}}.$$

The trade flow X is explained by Y_i and Y_j that are the masses of the exporting and importing country (e.g. the GDP) and D_{ij} that is the distance between the countries.

This is also used to study urban policies and migration flows!

Simple gravity model

Dummy variables such as common borders *contig* or regional trade agreements *rta* can be added to the model. Let t_{ij} be the transaction cost defined as:

$$t_{ij} = D_{ij} \exp(contig_{ij} + rta_{ij})$$

Simple gravity model

So that the model with friction becomes:

$$X_{ij} = G \frac{Y_i^{\beta_1} Y_j^{\beta_2}}{t_{ij}^{\beta_3}}.$$

Simple gravity model

A logarithmic operator can be applied to form a log-linear model and use a standard estimation methods such as OLS:

$$\log X_{ij} = \beta_0 \log G + \beta_1 \log Y_i + \beta_2 \log Y_j + \beta_3 \log D_{ij} + \beta_4 contig_{ij} + \beta_5 rta_{ij}$$

Trade barriers model

Trade barriers model

Basically the model proposes that the exports X_{ij} from i to j are determined by the supply factors in i, Y_i, and the demand factors in j, Y_j, as well as the transaction costs t_{ij}.

Next to information on bilateral partners i and j, information on the rest of the world is included in the gravity model with $Y = \sum_i Y_i = \sum_j Y_j$ that represents the worldwide sum of incomes (e.g. the world's GDP).

Trade barriers model

In this model σ represents the elasticity of substitution between all goods. A key assumption is to take a fixed value $\sigma > 1$ in order to account for the preference for a variation of goods (e.g. in this model goods can be replaced for other similar goods).

Trade barriers model

The Multilateral Resistance terms are included via the terms P, Inward Multilateral Resistance, and Π, Outward Multilateral Resistance.

The Inward Multilateral Resistance P_i is a function of the transaction costs of i to all trade partners j.

The Outward Multilateral Resistance Π_j is a function of the transaction costs of j to all trade partners i and their demand.

The Multilateral Resistance terms dependent on each other. Hence, the estimation of structural gravity models becomes *complex*.

Trade barriers model

Figure 2: What?

Trade barriers model

The econometric literature proposes the Multilateral Resistance model defined by the equations:

$$X_{ij} = \frac{Y_i Y_j}{Y} \frac{t_{ij}^{1-\sigma}}{P_j^{1-\sigma} \Pi_i^{1-\sigma}}$$

with

$$P_i^{1-\sigma} = \sum_j \frac{t_{ij}^{1-\sigma}}{\Pi_j^{1-\sigma}} \frac{Y_j}{Y}; \; \Pi_j^{1-\sigma} = \sum_i \frac{t_{ij}^{1-\sigma}}{P_i^{1-\sigma}} \frac{Y_i}{Y}$$

Trade barriers model

Figure 3: Again, what?

Model estimation

Model estimation

To estimate gravity equations you need a square dataset including bilateral flows defined by the argument dependent_variable, a distance measure defined by the argument distance that is the key regressor, and other potential influences (e.g. contiguity and common currency) given as a vector in additional_regressors are required.

Some estimation methods require ISO codes or similar of type character variables to compute particular country effects. Make sure the origin and destination codes are of type “character”.

Model estimation

The rule of thumb for regressors or independent variables consists in:

- All dummy variables should be of type numeric (0/1).
- If an independent variable is defined as a ratio, it should be logged.

The user should perform some data cleaning beforehand to remove observations that contain entries that can distort estimates, notwithstanding the functions provided within gravity package will remove zero flows and distances.

Examples

Double Demeaning

Double Demeaning subtracts importer and exporter averages on the left and right hand side of the respective gravity equation, and all unilateral influences including the Multilateral Resistance terms vanish.

Therefore, no unilateral variables may be added as independent variables for the estimation.

Double Demeaning

Our ddm function first logs the dependent variable and the distance variable.

Afterwards, the dependent and independent variables are transformed in the following way (exemplary shown for trade flows, X_{ij}):

$$(\log X_{ij})_{\text{DDM}} = (\log X_{ij}) - (\log X_{ij})_{\text{Origin Mean}} - (\log X_{ij})_{\text{Destination Mean}} + (\log X_{ij})_{\text{Mean}}.$$

Double Demeaning

One subtracts the mean value for the origin country and the mean value for the destination country and adds the overall mean value to the logged trade flows.

This procedure is repeated for all dependent and independent variables. The transformed variables are then used for the estimation.

Double Demeaning

An example of how to apply the function ddm to an example dataset in gravity and the resulting output is shown in the following:

```r
p_load(gravity)

fit <- ddm(
    dependent_variable = "flow",
    distance = "distw",
    additional_regressors = c("rta", "comcur", "contig"),
    code_origin = "iso_o",
    code_destination = "iso_d",
    data = gravity_no_zeros
  )
```

Double Demeaning

The package returns lm or glm objects instead of summaries. Doing that allows to use our functions in conjunction with broom or other packages, for example:

```r
p_load(broom)
tidy(fit)
```

```
## # A tibble: 4 x 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 dist_log_ddm   -1.60     0.0331    -48.4  0.
## 2 rta_ddm         0.797    0.0700     11.4  6.54e-30
## 3 comcur_ddm      0.174    0.146       1.19 2.34e- 1
## 4 contig_ddm      1.00     0.120       8.36 6.62e-17
```

Double Demeaning

```
glance(fit)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC
## *     <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>   <dbl>  <dbl>
## 1     0.254         0.254  2.32     1455.       0     4 -38611. 77232.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <in
```

Double Demeaning

How does the internal code work?

```
ddm <- function(dependent_variable,
                distance,
                additional_regressors = NULL,
                code_origin,
                code_destination,
                robust = FALSE,
                data, ...) {
...
```

Double Demeaning

How does the internal code work?

```
# Checks
  stopifnot(is.data.frame(data))
  stopifnot(is.logical(robust))
...

# Transforming data, logging distances
  d <- d %>%
    mutate(
      dist_log = log(!!sym(distance))
    )
...
```

Double Demeaning

How does the internal code work?

```
# Transforming data, logging flows
  d <- d %>%
    mutate(
      y_log = log(!!sym(dependent_variable))
    )
...
```

Double Demeaning

How does the internal code work?

```
# Substracting the means
  d <- d %>%
    mutate(
      y_log_ddm = !!sym("y_log"),
      dist_log_ddm = !!sym("dist_log")
    ) %>%
    group_by(!!sym(code_origin), add = FALSE) %>%
    mutate(
      ym1 = mean(!!sym("y_log_ddm"), na.rm = TRUE),
      dm1 = mean(!!sym("dist_log_ddm"), na.rm = TRUE)
    ) %>%
...
```

Code and documentation

Code and documentation

github.com/pachamaltese/gravity

pacha.hk/gravity

Questions?

Questions?

Thanks for your attention!

References

References

Badi H. Baltagi. *Econometrics*. Number 978-3-642-20059-5 in Springer Texts in Business and Economics. Springer, June 2011. doi: 10.1007/978-1-4614-3169-5.

Anna Lenna Woelwer, Jan Pablo Burgard, Joshua Kunst, and Mauricio Vargas. Gravity: Estimation methods for gravity models in r. *Journal of Open Source Software*, 31(3): 1038, 2018. doi: 10.21105/joss.01038.