

Codes for “A Neyman Orthogonalization Approach to the Incidental Parameter Problem in Likelihood Models”

Bonhomme, Jochmans, Weidner

Overview

The code in this replication package produces the results of the paper based on data from Ductor, Fafchamps, Goyal and van der Leij (2014).

Data availability statement

The file `networkdataac.csv` is freely available at the persistent archive indicated below.

The file `networkdataac.csv` is the raw data file exactly as downloaded from the Harvard Data-verse archive (<https://doi.org/10.7910/DVN/27138>). Access to it is immediate and requires no registration; no processing or editing was applied before it is read by the replication code.

We provide the file `networkdataac.csv` in the replication package.

We certify that the authors of the manuscript have legitimate access to and permission to use the data used in the manuscript. All data are publicly available.

References

The data set is the file `networkdataac.csv`, from Ductor, Fafchamps, Goyal and van der Leij (2014, *Review of Economics and Statistics*; https://doi.org/10.1162/REST_a_00430).

The data set can be freely downloaded at <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/27138>.

Computational requirements

The codes are run in Stata (version 19.5) and Matlab (version R2024b), including the commands `parfor` (from Parallel Computing), `fminunc` (from Optimization) and `quantile` (from Statistics and Machine Learning).

Random seeds are set at the beginning of each of the main codes. In practice, estimates may vary by a few decimals depending on the software used (for instance, `fminunc` uses finite-difference derivatives whose rounding differs across Matlab releases).

Running times for each of the main codes are documented below. They are based on the following machine (desktop):

- Dell XPS 8930, x64-based PC
- Processor: Intel(R) Core(TM) i9-9900K CPU @ 3.60 GHz, 3600 MHz, 8 Core(s), 16 Logical Processor(s)
- RAM: 64.0 GB

The parametric bootstrap (40 h) and simulation (6h30) are the computationally demanding scripts; both use `parfor` and benefit from multiple cores. They were run on 8 workers on a machine with 64 GB of RAM. `Code_estimation` and `Code_q_diagnostic` are moderate (a few hours on the same machine). `Code_descriptives`, `Code_prepare_data`, and `Code_GMM` run in minutes.

Description of codes

First run the Stata code `Code_descriptives.do`.

- To do so, you will need to enter your path on line 10 of the code.
- All files need to be in the same directory for the codes to run properly.

This code produces the descriptive statistics reported in the text, as well as the Stata file `data_research.dta` and the file `data_research.out` (for use in Matlab) that is subsequently used in the analysis.

Based on this, process the data for Matlab using the Matlab code `Code_prepare_data`.

- *Time: 1 min*

Then, run the Matlab code `Code_estimation`. This produces the point estimates for Tables 1 and 2, and Figures 1 and 2 in the supplement.

- *Time: 3h30*

Then, run the Matlab code `Code_q_diagnostic`. This produces the numbers in the last column of Table 1.

- *Time: 1h15*

Then, run the Matlab code `Code_parametric_bootstrap`. This produces the standard errors in Tables 1 and 2.

- *Time: 40 hours*

In addition, the Matlab code `Code_simulation` produces the numbers in Tables 3 and 4 in the supplement.

- *Time: 6h30*

And the Matlab code `Code_GMM` produces the GMM estimates reported in the supplement.

- *Time: 2 min 30 s (data) and 7 min 20 s (simulation)*

Note: the codes display the two figures in Matlab figure windows; the tables are generated inside Matlab, and LaTeX versions are available in the `output` folder.

Contact

For questions, contact Stéphane Bonhomme at sbonhomme@uchicago.edu.