Overview

This replication package consists of Matlab programs for the Monte Carlo exercise in the paper "Deconvolution from two order statistics." The main program runs from the data generating procedure to the reproduction of Figure 1 in the paper and Figures 4 and 5 in Appendix C. The figures generated from the estimates will be automatically saved as pdf with the appropriate names in the output folder.

The replicator should expect the code to run for a long time. The Monte Carlo simulation consists of replicating the same estimation 500 times for different sample sizes, where the runtime depends on the sample size. Specifically, one-time estimation with a sample size of 1000 runs for 67 minutes, so a replication of 500 times with 64 cores parallel computing runs around 67*500/64/60=9 hours; One-time estimation with a sample size of 2000 runs for 3 hours, so a replication of 500 times with 64 cores parallel computing runs around 3*500/64=24 hours; One-time estimation with a sample size of 4000 runs for 4 hours, so a replication of 500 times with 64 cores parallel computing runs around 4*500/64=32 hours.

Data Availability and Provenance Statements

This paper does not involve analysis of external data (i.e., no data are used, or the only data are generated by the authors via simulation in their code).

Computational requirements

Global Optimization Toolbox (Matlab).

Statistics and Machine Learning Toolbox (Matlab).

Parallel Computing Toolbox (Matlab) if using `parfor` loop (see "Controlled Randomness" and "Instructions to Replicators" below).

Software Requirements

Matlab Release 2021a

Controlled Randomness

Random seed is set at line 10 and 112 of `programs/main.m`. Line 10 controls the randomness in the samples drawn from the DGP so that the estimates can be exactly reproduced using a `parfor` loop. Line 112 controls the randomness in the subset of estimates from the simulation exercise to plot in Figures 1, 4, and 5.

Random seed is set at line 17 of `programs/calibrate_hqt2020.m`. It controls the optimization routine for calibrating the true value of the DGP parameters for the Monte Carlo exercise.

We implemented the code using parallel processing (with `parfor`). Despite the random seeds, the exact output may thus differ if the code is run using a `for` loop instead of `parfor`.

Memory and Runtime Requirements

Summary

Approximate time needed to reproduce the analyses on a standard desktop machine:

- Not feasible to run on a desktop machine, as described below.

Details

The code was last run on **64-core supercomputers.** Computation took around 65 hours for Figure 1 in the main text and Figures 4 and 5 in Appendix C. `programs/main.m` and `calibrate_hqt2020rand.m` are the two primary programs and the remaining programs are helper functions.

- `calibrate_hqt2020rand.m`: 1 minute
- `main.m`: 9—32 hours depending on the sample size

Description of programs/code

The Matlab programs in `programs` generate Figure 1 in the main body of the article and Figures 4 and 5 in Appendix C. The program `programs/main.m` runs them all from scratch. Output files are called appropriate names (see details below) and should be easy to correlate with the manuscript.

Instructions to Replicators

- Adjust the default path to `programs/`.
- To replicate Figures 1, 4, and 5, the replicator needs to run `main.m`, which involves a `parfor` loop. The replicator should specify the number of workers in line 67. The default is set to 64 workers.
  If the replicator wishes to use a `for` loop instead, the replicator may change the `parfor` loop in line 68 to `for` and comment out lines `67 %parpool(64)` and 103 `%delete(gcp)`. We note that despite the random seeds, results may differ if one uses the `for` loop.
- To replicate Figure 1 in Section 4.2, run `main.m` three times, each time changing the sample size and sieve order in lines 18 and 19, leaving all other parameters fixed (in particular, in line 20, set `kappa = 1`).

N	1000	2000	4000
k	4	5	6
kappa	1	1	1

- To replicate Figures 4 and 5 in Appendix C.3, run `main.m` nine times, each time changing the sample size, sieve order, and kappa in lines 18—20. (The first three specifications are the same as above, so the replicator needs to run the program six more times). [Note: The simulation results stored in this replication package are computed with a different random seed in line 10 (using `rng(1618)` instead of `sc` in the current package). Thus the estimates may differ, but the figures are expected to be similar.]

N	1000	2000	4000	1000	2000	4000	1000	2000	4000
k	4	5	6	4	5	6	4	5	6
kappa	1	1	1	3.14	3.14	3.14	5	5	5

Details

- `programs/calibrate_hqt2020rand.m` calibrates the true DGP and saves the result in `dgp_params.txt`. The true DGP is obtained from Figure 4 of Hernández et al. (2020), stored in `HQT2020RAND_fig4a.txt` (for the latent variable) and `HQT2020RAND_fig4b.txt` (for measurement error). These text files are extracted from the original article using `WebPlotDigitizer (https://automeris.io/WebPlotDigitizer)`.
  The program is also used to calibrate initial values of the optimization procedure in `main.m`. The results are stored in `dgp_params_approx_k=3.txt` (with `k=3` in line 33), `dgp_params_approx_k=4. txt` (with `k=4` in line 33), and `dgp_params_approx_k=5. txt` (with `k=5` in line 33). Because the calibrated outputs are stored in the folder `programs`, the replicator does not need to run this program to reproduce the figures.
- `programs/main.m` generates all figures in the article. There are three parameters to change for different specifications: `N` (Sample size) on line 18, `k` (the number of sieve bases) on line 19, `kappa` (tuning parameter) on line 20. Output files are given appropriate names:
  - `main_out_N=#_k=#_kappa=#_R=#.mat`: stores all objects from the simulation
  - `cdfepsilon_N=#_k=#_kappa=#_R=#.pdf`: figure for c.d.f. of measurement error
  - `cdfxi_N=#_k=#_kappa=#_R=#.pdf`: figure for c.d.f. of the latent variable
  - `main_out.xlsx`: stores estimated values of the sieve parameter

List of tables and programs

The provided code reproduces Figure 1 in the paper and Figures 4-5 in Appendix C. The figures generated from the estimates will be automatically saved as pdf with the appropriate names in the output folder.

References

Hernández, Cristián, Daniel Quint, and Christopher Turansick. "Estimation in English auctions with unobserved heterogeneity." The RAND Journal of Economics 51, no. 3 (2020): 868-904.

Acknowledgements

We thank Cristián Hernández, Daniel Quint, and Christopher Turansick for sharing their code.