

# README

---

This document describes how to reproduce all simulation and empirical results in "Adaptive Estimation and Uniform Confidence Bands for Nonparametric Structural Functions and Elasticities" by Chen, Christensen, and Kankanala.

## Data Availability Statement

---

The replication uses data from

Adao, R., C. Arkolakis, and S. Ganapati (2020). "Aggregate implications of firm heterogeneity: A nonparametric analysis of monopolistic competition trade models." NBER Working Paper no. 28081, dated November 2020.  
<https://www.nber.org/papers/w28081>

A description of the data can be accessed at the above link. The data is provided as part of this package and is located at `./data/2012.csv`.

## Computational Requirements

---

## Software Requirements

---

All files run in Matlab version 2020b+ (code was run with Matlab Release 2022a).

## Controlled Randomness

---

Random seeds are set in all files (including the empirical application, which requires bootstrapping).

## Memory and Runtime Requirements

---

Approximate time needed to reproduce the empirical application on a standard 2023 Desktop machine: < 10 minutes.

Approximate time needed to reproduce the simulations on a standard 2023 Desktop machine: 2-8 hours. These can be run in parallel.

## Details

---

The code was last run on a 2020 Macbook Pro with a M1 Chip and 16GB Memory.

## Replication Instructions

---

### Empirical Application (Section 3.3)

---

To reproduce the empirical results, run `AAG_setup.m`. This generates the files for Figure 1 in `./results/aag_1.eps` and `./results/aag_2.eps`.

### Simulations (Section 3.4 and Appendix B)

---

#### Tables 1, 6, 7, and 8

There are three main files, one for each design:

- `AAG_sim_NL.m` (lognormal design without fixed effects)
- `AAG_sim_NL_FE.m` (lognormal design with fixed effects)
- `AAG_sim_FE.m` (constant elasticity design with fixed effects)

Results in Table 1 are generated by `AAG_sim_NL_FE.m`. Results in Tables 6 and 7 are generated by `AAG_sim_NL.m`. Results in Table 8 are generated by `AAG_sim_FE.m`.

**Note:** Each file is set up to be run on a cluster in one of four configurations corresponding to the environment variable `SLURM_ARRAY_TASK_ID`, an integer between 1 (smallest sample size) and 4 (largest sample size). This can be modified by replacing

`array_value = str2double(getenv('SLURM_ARRAY_TASK_ID'));` with `array_value = int;` where `int` is an integer between 1 and 4.

Results are saved to `./results/aag_***_int.mat` where `***` matches the `.m` filename. Calling `make_tables.m` will tabulate results as reported in the paper.

## Figures 2, 5, and 6

Figures 2 and 5 (log normal design with and without fixed effects, respectively) are generated by running `make_plots_AAG_NL.m`. Figure 6 (pareto design with fixed effects) is generated by running `make_plots_AAG.m`. Figures are saved to `./results/aag_***_int.eps`, with the same naming convention as the above three `.m` files.

## Simulations (Section 5.1)

---

### Tables 2 and 3

Results in Table 2 are generated by `run_additional_npiv.m`. Results in Table 3 are generated by `run_additional_NPIV_A_table.m`.

**Note:** Each file is set up to be run on a cluster in one of eight configurations corresponding to the environment variable `SLURM_ARRAY_TASK_ID`, an integer between 1 (smallest sample size, losses trimmed to [0.01, 0.99]) and 8 (largest sample size, no trimming). This can be modified by replacing `array_value = str2double(getenv('SLURM_ARRAY_TASK_ID'));` with `array_value = int;` where `int` is an integer between 1 and 8.

Results are saved to `./results/npiv_2_int.mat` and `./results/npiv_2_a_int.mat`. Calling `make_tables.m` will tabulate results as reported in the paper.

### Figure 3

Figure 3 is generated by running `make_plots_additional_npiv.m`. Figures are saved to `./results/npiv_sim_int.eps`, with `int` an integer between 1 and 8.

# Simulations (Section 5.2)

---

## Tables 4 and 5

Results in Table 4 are generated by `run_nonlinear_regression.m`. Results in Table 3 are generated by `run_nonlinear_regression_A_table.m`.

**Note:** Each file is set up to be run on a cluster in one of eight configurations corresponding to the environment variable `SLURM_ARRAY_TASK_ID`, an integer between 1 (smallest sample size, losses trimmed to [0.01, 0.99]) and 8 (largest sample size, no trimming). This can be modified by replacing `array_value = str2double(getenv('SLURM_ARRAY_TASK_ID'))`; with `array_value = int`; where `int` is an integer between 1 and 8.

Results are saved to `./results/regression_int.mat` and `./results/regression_a_int.mat`. Calling `make_tables.m` will tabulate results as reported in the paper.

## Figure 4

Figure 3 is generated by running `make_plots_regression.m`. Figures are saved to `./results/reg_sim_int.eps`, with `int` an integer between 1 and 4.

# Simulations (Appendix C)

---

## Table 9

Results in Table 9 are generated by `run_engel_copula.m`.

**Note:** The file is set up to be run on a cluster in one of eight configurations corresponding to the environment variable `SGE_TASK_ID`, an integer between 1 (smallest sample size, losses trimmed to [0.01, 0.99]) and 8 (largest sample size, no trimming). This can be modified by replacing `array_value = str2double(getenv('SGE_TASK_ID'))`; with `array_value = int`; where `int` is an integer between 1 and 8.

Results are saved to `./results/engel_copula_int.mat`. Calling `make_tables.m` will tabulate results as reported in the paper.

## Figure 7

Figure 7 is generated by running `make_plots_engel_copula.m`. Figures are saved to `./results/engel_copula_int.eps`, with `int` an integer between 1 and 8.