

README: Replication package for “Testable Implications of Parametric Assumptions in Ordered Random Utility Models”

Jose Apesteguia and Miguel A. Ballester

Econometrica MS 22294, RP 321 — August 2026 (revision 1)

Code author: Carlos Gonzalez-Perez.

Overview

The code in this replication package simulates all datasets used in the paper and produces every table and figure in the paper and its appendices (all computational exhibits appear in Appendix D, “Empirical Illustration”). **No external data are used:** all data are generated by the code itself under fixed random seeds.

A single master script, `code_final/00_main.R`, runs the entire pipeline: it installs any missing R packages, simulates the menu and choice datasets for the risk and altruism applications, performs the nonparametric (Bellman–Ford/WARP), semi-nonparametric (J-sample Anderson–Darling), and parametric (censored-logit MLE and parametric-bootstrap AD test) analyses, and saves all exhibits to `output_final/` and all simulated datasets to `data_simulated/`. The replicator should expect the code to run for approximately 5-10 minutes, even on a single core.

Data Availability and Provenance Statements

- ☒ **This paper does not involve analysis of external data.** The only data are generated by the authors via simulation in their code.

All datasets are simulated within `00_main.R` (through functions defined in `01_risk.R` and `02_altruism.R`) using fixed random seeds, so they are regenerated on every complete run of the master script. For convenience and transparency, a copy of every simulated dataset is nevertheless included in the package, in the folder `data_simulated/`, in non-proprietary CSV format. The data-generating processes are described in Appendix D.1 of the paper (“Data”) and are fully parameterized in the code.

Statement about Rights

- ☒ We certify that the authors of the manuscript have legitimate access to and permission to use all data in this manuscript (all data are generated by the authors’ own code).
- ☒ We certify that the authors have the right to redistribute and publish the data contained in this replication package.

License for Data and Code

See `LICENSE.txt`. The code is licensed under the MIT license and the simulated datasets under CC-BY 4.0.

Summary of Availability

- ☒ All data **are** publicly available (included in this package and regenerated by the code).

Dataset list

All files are in `data_simulated/` and are written by `00_main.R` (they are also shipped with the package). For the choice datasets, `<N>` takes the values 20, 80, 160, 480 (choices per menu).

Data file	Content	Generated by
<code>risk_menu_data.csv</code>	20 risk menus (linear budgets over state-contingent consumption)	<code>data_gen()</code> + <code>order_data()</code> , seed 12345
<code>risk_choice_data_n<N>.csv</code>	N CRRA-logistic simulated choices per risk menu	<code>choice_data_gen()</code> , seed 2
<code>altruism_menu_data.csv</code>	20 altruism menus (linear budgets over interpersonal allocations)	<code>data_gen_alt()</code> , seed 12345
<code>altruism_choice_data_n<N>.csv</code>	N CES-logistic simulated choices per altruism menu	<code>choice_data_alt_gen()</code> , seed 12345

Codebook. `menu`: menu identifier (menus are sorted by the price/probability ratio, see paper); `pi_1`, `pi_2`: prices of the two alternatives; `q_1`, `q_2`: state probabilities (risk only); `phi_1`, `phi_2` and `log_phi_1`, `log_phi_2`: price-probability ratios $\phi_j^s = \pi_j^s / q_j^s$ and their logs (risk only); `phi_j` / `pi_j`: menu index $\phi_j = \phi_j^2 / \phi_j^1$ (risk) or $\pi_j = \pi_j^2 / \pi_j^1$ (altruism); `t_real`: the true simulated type drawn from the logistic distribution; `c_1`, `c_2`: chosen consumption of each alternative; `r`: consumption ratio c_2 / c_1 ; `v_j`: transformation $1 - e^{-r}$ (altruism only); `n_obs`: dataset size N; `quantile`: within-menu percentile rank of the observation.

Computational requirements

Software Requirements

- ☒ The master script `00_main.R` installs all missing dependencies automatically from CRAN (internet connection required only if packages are missing) and creates the necessary directory structure.
- R (version 4.3 or later).
- CRAN packages, installed automatically by `00_main.R` if missing: `dplyr`, `ggplot2`, `stringr`, `tidyr`, `igraph`, `future`, `future.apply`, `kableExtra`, `latex2exp`, `textshape`, `kSamples` (which loads `SuppDists`). The Bellman–Ford negative-cycle detection used in the nonparametric risk analysis is implemented natively in `01_risk.R` (function `bellman()`) and requires no additional package.

- **Versions of the verified run.** The shipped outputs in `output_final/` were produced on the authors' machine under R 4.3.2 (Windows 11) with `dplyr` 1.1.2, `ggplot2` 3.5.1, `stringr` 1.5.0, `tidyr` 1.3.0, `igraph` 2.0.3, `future` 1.33.0, `future.apply` 1.11.0, `kableExtra` 1.4.0, `latex2exp` 0.9.6, `textshape` 1.7.5, `kSamples` 1.2-10, and `SuppDists` 1.1-9.9. The full session record of this run is shipped in `output_final/session_info.txt`, and its exact package environment is pinned in the lockfile `renv.lock` at the package root (see next point). The results were additionally verified under R 4.3.3 on Ubuntu 24.04, and the code has been checked to run correctly under substantially newer package versions (e.g., `ggplot2` 4.0.x, `igraph` 2.3.x).
- **Exact version pinning (optional).** By default, `00_main.R` installs whatever package versions are current on CRAN, which is sufficient to reproduce all exhibits. Replicators who wish to reproduce the exact package environment of the verified run can instead restore it from the shipped lockfile before running the master script:

```
install.packages("renv")
renv::restore(lockfile = "renv.lock")
```

Controlled Randomness

- ☒ Random seeds are set for every stochastic step:
 - Menu generation: seed 12345, set in `data_gen()` (`01_risk.R`) and `data_gen_alt()` (`02_altruism.R`).
 - Choice-data simulation: seed 2 for risk (passed in `00_main.R`, Section 1.A; see Step 3 in `01_risk.R` for the seed-selection discussion) and seed 12345 for altruism (default of `choice_data_alt_gen()`).
 - Menu subsampling for figures: seed 12345 (set inside the plotting functions).
 - CLA subsampling: seed 12345 (in both `cla_data()` and `cla_data_alt()`).
 - Parametric bootstrap ($B = 1000$): seed 12345, set in `bootstrap_ad_test()` / `bootstrap_ad_test_alt()`, with `future.seed = TRUE`, which assigns parallel-safe L'Ecuyer-CMRG streams to each bootstrap draw. **Bootstrap results are therefore invariant to the number of parallel workers used.**

Memory, Runtime, Storage Requirements

Approximate time needed to reproduce the analyses on a standard (2026) desktop machine:

- ☒ <10 minutes

Approximate storage space needed:

- ☒ < 25 MBytes

Details. The code was verified on: (i) a Windows 11 laptop with an 11th Gen Intel Core i5-1135G7 @ 2.40GHz (4 cores) and 16 GB RAM, running R 4.3.2 (the run that produced the shipped outputs; see `output_final/session_info.txt` and `output_final/run_log.txt`); and (ii) a 1-vCPU Ubuntu 24.04 (x86-64) container with R 4.3.3. Total runtime is approximately 7-8 minutes on (i) and 8 minutes on (ii). Both environments reproduce every exhibit of the paper to full reported precision; the raw simulated datasets may differ across operating systems in a handful of cells at machine precision (absolute deviations of order $1e-14$), owing to floating-point differences in the underlying mathematical libraries, without affecting any exhibit. The most time-consuming step is the recovery of the altruism meta-parameter α (approximately 3-5 minutes); all remaining steps take seconds (the Bellman-Ford

negative-cycle detection is implemented natively and runs in under a minute for all datasets). Memory requirements are negligible (< 1 GB). The number of parallel workers is set in `00_main.R`, Section 0c (default: `max(1, min(4, future::availableCores() - 1))`); changing it does not change any result. A timestamped log of each section is written to `output_final/run_log.txt`, and the exact package versions of a run are written to `output_final/session_info.txt`; the package ships with the log and session record of the authors' verified run.

Description of programs/code

All programs are in `code_final/`:

- `00_main.R` — **master script**. Installs/loads packages, detects the package root, creates `output_final/` and `data_simulated/`, sources the two function files, and executes the entire analysis in order: (1) risk application (Sections 1.A–1.D) and (2) altruism application (Sections 2.A–2.D). Every exhibit of the paper is saved to `output_final/` (LaTeX tables as `.tex`, underlying numbers as `.csv`, figures as `.png/.pdf`) and also displayed in the console.
- `01_risk.R` — functions for the risk application (data simulation, weighted-graph construction, native Bellman–Ford negative-cycle detection, seed selection discussion, induced-parameter recovery, J-sample AD test, mass analysis, CLA construction, censored-logit MLE, parametric-bootstrap AD test) plus general-purpose helpers shared with the altruism application. Sourced by `00_main.R`; not run directly.
- `02_altruism.R` — functions for the altruism application (data simulation, WARP verification, α recovery, induced-parameter recovery, AD test, mass analysis, CLA construction, MLE and bootstrap counterparts). Sourced by `00_main.R`; not run directly.
- `code_final.Rproj` — RStudio project file (optional convenience).
- `renv.lock` (package root) — lockfile pinning the exact package versions of the verified run (see Software Requirements).

The package ships with the outputs of the authors' verified run already in `output_final/` (including the corresponding `run_log.txt` and `session_info.txt`); executing `00_main.R` regenerates and overwrites every file in that folder.

Instructions to Replicators

1. Unzip the replication package to any location, preserving the folder structure.
2. Ensure an internet connection is available if R packages may be missing (they are installed automatically from CRAN). Optionally, restore the exact package versions of the verified run first via `renv::restore(lockfile = "renv.lock")` (see Software Requirements).
3. Run the master script from top to bottom in a fresh R session. Either:
 - from the command line, at the package root: `Rscript code_final/00_main.R`; or
 - in RStudio: open `code_final/code_final.Rproj`, open `00_main.R`, and run the entire file (the package root is detected automatically); or
 - in a plain R console: `setwd("<package root>")` and `source("code_final/00_main.R")`.
4. All exhibits are written to `output_final/` (see mapping below). No manual steps are required.

List of tables, figures and programs

The provided code reproduces:

- ☒ All tables and figures in the paper and approved appendices. (All computational exhibits of

the paper appear in Appendix D; there are no code-generated in-text numbers outside these exhibits. Table 2 of the paper is theoretical and involves no computation; its weight definitions are implemented in `matrix_function()` in `01_risk.R`.)

Exhibit (Appendix D)	Produced by (function, called from <code>00_main.R</code>)	Output file(s) in <code>output_final/</code>
Table 1(a)	<code>export_table_1a()</code> (§1.A)	<code>table1a_risk_menus.tex / .csv</code>
Table 1(b)	<code>export_table_1b()</code> (§2.A)	<code>table1b_altruism_menus.tex / .csv</code>
Table 2	n.a. (theoretical; implemented in <code>matrix_function()</code>)	n.a.
Table 3	<code>lapply_routine_bell()</code> (§1.B)	<code>table3_risk_nonrationalizable_quantiles</code>
Table 4	<code>altr_warp()</code> (§2.B)	<code>table4_altruism_warp_violations.csv</code>
Table 5	<code>recover_alpha() + median_alpha()</code> (§2.C)	<code>table5_altruism_median_alpha.csv</code>
Table 6 (risk cols)	<code>similarity_testing_wrap()</code> (§1.C)	<code>table6_risk_ad_test.tex</code>
Table 6 (altruism cols)	<code>similarity_testing_alt_wrap()</code> (§2.C)	<code>table6_altruism_ad_test.tex</code>
Table 7	<code>mass_analysis() + mass_output()</code> (§1.D)	<code>table7_risk_mass_at0.tex / .csv</code>
Table 8	<code>mass_analysis() + mass_output()</code> (§1.D)	<code>table8_risk_largest_r.tex / .csv</code>
Table 9	<code>mass_analysis_alt() + mass_output_alt()</code> (§2.D)	<code>table9_altruism_mass_at0.tex / .csv</code>
Table 10	<code>mass_analysis_alt() + mass_output_alt()</code> (§2.D)	<code>table10_altruism_largest_v.tex / .csv</code>
Table 11 (left, risk)	<code>parametric_estimation()</code> (§1.D)	<code>table11_risk_mle.csv</code>
Table 11 (left, altruism)	<code>parametric_estimation_alt()</code> (§2.D)	<code>table11_altruism_mle.csv</code>
Table 11 (right, risk)	<code>bootstrap_ad_test()</code> (§1.D)	<code>table11_risk_bootstrap_ad_test.csv</code>
Table 11 (right, altruism)	<code>bootstrap_ad_test_alt()</code> (§2.D)	<code>table11_altruism_bootstrap_ad_test.csv</code>
Figure 1(a)	<code>plot_cdf()</code> (§1.A)	<code>figure1a_risk.png</code>
Figure 1(b)	<code>plot_cdf_v()</code> (§2.A)	<code>figure1b_altruism.png</code>
Figure 2 (four panels)	<code>gen_weight_matrix(plot = T)</code> (§1.B)	<code>figure2_risk_n20.pdf, ..._n80.pdf, ..._n160.pdf, ..._n480.pdf</code>
Figure 3(a)	<code>plot_induced_param()</code> (§1.C)	<code>figure3a_risk.png</code>
Figure 3(b)	<code>plot_induced_param_alt()</code> (§2.C)	<code>figure3b_altruism.png</code>
Figure 4(a)	<code>plot_cla()</code> (§1.D)	<code>figure4a_risk.png</code>
Figure 4(b)	<code>plot_cla_alt()</code> (§2.D)	<code>figure4b_altruism.png</code>

Auxiliary files: `run_log.txt` (timestamped section log) and `session_info.txt` (exact package versions), both shipped for the authors’ verified run and regenerated on every execution.

References

No external data sources are used in this paper. Software references:

R Core Team (2024). *R: A Language and Environment for Statistical Computing* [software]. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>

Scholz, Fritz, and Angie Zhu (2024). *kSamples: K-Sample Rank Tests and their Combinations* [software], R package version 1.2-10. <https://CRAN.R-project.org/package=kSamples>

Bengtsson, Henrik (2021). “A Unifying Framework for Parallel and Distributed Processing in R using Futures,” *The R Journal*, 13(2), 208–227 (R packages `future` and `future.apply`).

Ushey, Kevin, and Hadley Wickham (2025). *renv: Project Environments* [software]. <https://CRAN.R-project.org/package=renv>