Reproduction Instructions for ‘Estimating the size of a population through repeated sampling: a survey sampling view on capture-recapture procedures’ by N.Sapargali and G.Kauermann

Code by N.Sapargali[*]

1 Software Environment

1.1 Operating System Info

```
Distributor ID: Debian
Description:    Debian GNU/Linux 13 (trixie)
Release:        13
Codename:       trixie
Kernel version: 6.12.57+deb13-amd64
CPU:            Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz
```

1.2 Julia Environment

```
Julia Version 1.12.3
Commit 966d0af0fdf (2025-12-15 11:20 UTC)
Build Info:
  Official https://julialang.org release
Platform Info:
  OS: Linux (x86_64-linux-gnu)
  CPU: 8 × Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz
  WORD_SIZE: 64
  LLVM: libLLVM-18.1.7 (ORCJIT, skylake)
  GC: Built with stock GC

# Direct Dependencies (Pkg.status())
Status ‘./Project.toml‘
    [336ed68f] CSV v0.10.15
    [a93c6f00] DataFrames v1.8.1
    [31c24e10] Distributions v0.25.122
    [2ab3a3ac] LogExpFunctions v0.3.29
    [429524aa] Optim v1.13.3
    [91a5bcdd] Plots v1.41.3
    [6f49c342] RCall v0.14.10
    [276daf66] SpecialFunctions v2.6.1
```

[*] nurzhan.sapargali@stat.uni-muenchen.de

```
    [2913bbd2] StatsBase v0.34.9
    [37e2e46d] LinearAlgebra v1.12.0
    [9a3f8284] Random v1.11.0
```

For the full list of installed packages including indirect dependencies, please refer to the `Manifest.toml` file included in the code supplement.

1.3 R Environment

```
R version 4.5.0 (2025-04-11)
Platform: x86_64-pc-linux-gnu
Running under: Debian GNU/Linux 13 (trixie)

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.29.so;
LAPACK version 3.12.0

locale:
 [1] LC_CTYPE=en_GB.UTF-8     LC_NUMERIC=C                  LC_TIME=en_GB.UTF-8
 [4] LC_COLLATE=en_GB.UTF-8   LC_MONETARY=en_GB.UTF-8       LC_MESSAGES=en_GB.UTF-8
 [7] LC_PAPER=en_GB.UTF-8     LC_NAME=C                     LC_ADDRESS=C
 [10] LC_TELEPHONE=C          LC_MEASUREMENT=en_GB.UTF-8    LC_IDENTIFICATION=C

time zone: Europe/Berlin
tzcode source: system (glibc)

attached base packages:
 [1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] lubridate_1.9.4        forcats_1.0.1          stringr_1.6.0          purrr_1.2.0
 [5] readr_2.1.5            tidyr_1.3.1            tibble_3.3.0           tidyverse_2.0.0
 [9] directlabels_2025.6.24 ggpubr_0.6.2           ggplot2_4.0.0          dplyr_1.1.4

loaded via a namespace (and not attached):
 [1]  generics_0.1.4     rstatix_0.7.3      stringi_1.8.7      hms_1.1.4
 [5]  magrittr_2.0.4     grid_4.5.0         timechange_0.3.0   RColorBrewer_1.1-3
 [9]  backports_1.5.0    Formula_1.2-5      gridExtra_2.3      scales_1.4.0
 [13] textshaping_1.0.4  abind_1.4-8        cli_3.6.5          rlang_1.1.6
 [17] cowplot_1.2.0      withr_3.0.2        tools_4.5.0        tzdb_0.5.0
 [21] ggsignif_0.6.4     broom_1.0.10       vctrs_0.6.5        R6_2.6.1
 [25] lifecycle_1.0.4    car_3.1-3          ragg_1.5.0         pkgconfig_2.0.3
 [29] pillar_1.11.1      gtable_0.3.6       glue_1.8.0         systemfonts_1.3.1
 [33] tidyselect_1.2.1   rstudioapi_0.17.1  farver_2.1.2       labeling_0.4.3
 [37] carData_3.0-5      compiler_4.5.0     S7_0.2.0           quadprog_1.5-8
```

2 Folder Structure

```
/
├── _100_input/ .......................... Root directory for all input data
│   ├── datasets/ ........ Real-world datasets and Table 2 in supplementary materials
│   └── simulated/ ................ Target folder for generated simulation inputs
├── _900_output/ ................... Target folder for all estimations and figures
│   ├── data/ ...... Estimation from simulations (contains pre-computed full results)
│   │   ├── appendix/ ......................... Results for supplementary materials
│   │   │   ├── beta_bin/. ............................... Beta-binomial estimation
│   │   │   │   ├── estimates_0.5_betabin.csv ........ Full simulation results (α = 0.5)
│   │   │   │   └── estimates_2.0_betabin.csv ........ Full simulation results (α = 2.0)
│   │   │   ├── low_pop/ ................. Estimation under low population sizes
│   │   │   │   ├── estimates_0.5_low_pop.csv. ....... Full simulation results (α = 0.5)
│   │   │   │   └── estimates_2.0_low_pop.csv. ....... Full simulation results (α = 2.0)
│   │   │   ├── low_sample/ .................. Estimation under low sample sizes
│   │   │   │   ├── estimates_0.5_low_sample.csv ..... Full simulation results (α = 0.5)
│   │   │   │   └── estimates_2.0_low_sample.csv ..... Full simulation results (α = 2.0)
│   │   │   └── one_inflation_equiv/ ..... Target folder for Table 1 in supplementary materials
│   │   ├── datasets/ ................. Target folder for Tables 2 and 3 in main text
│   │   └── simulated/ ............................................. Main results
│   │       ├── estimates_0.5.csv ................ Full simulation results (α = 0.5)
│   │       └── estimates_2.0.csv ................ Full simulation results (α = 2.0)
│   └── figures/ ....... Target folder for figures from manuscript and supplementary materials
│       ├── simulated
│       └── appendix
├── _100_utils.jl ............................................. Utility functions
├── _110_beta_estimator.jl ............................... Beta-binomial estimator
├── _120_one_nbin.jl ................................. Negative Binomial estimator
├── _130_benchmarks.jl .................... Alternative estimators as benchmarks
├── _140_simulation_functions.jl .............. Core logic for Sampford sampling
├── _150_plot_functions.R ............................. Shared R plotting functions
├── _200_simulate_samples.jl .............. Simulation: standard sample generation
├── _210_simulate_low_pop.jl ................ Simulation: low population scenarios
├── _220_simulate_low_sample.jl ........... Simulation: low sample size scenarios
├── _299_simulate_all.jl ................... Master Script: Runs all simulations
├── _300_estimate_simulated.jl .............. Estimation: standard simulated data
├── _310_estimate_low_pop.jl ..................... Estimation: low population data
├── _320_estimate_low_sample.jl ................ Estimation: low sample size data
├── _330_estimate_beta.jl ...................... Estimation: Beta-binomial model
├── _399_estimate_all_simulated.jl ........ Master Script: Runs all or partial estimations
├── _400_estimate_data.jl ................... Constructs Tables 2 and 3 in main text
├── _500_plot_simulated.R ......................... Plots Figures 1 to 5 in main text
├── _510_plot_low_pop.R ............. Plots Figures 7 to 11 in supplementary materials
├── _520_plot_low_sample.R ......... Plots Figures 12 to 16 in supplementary materials
├── _600_poisson_equivalence.jl ........ Plots Figure 1 in supplementary materials
└── _610_instability_plots.R ........ Plots Figures 2 and 3 in supplementary materials
```

- _620_one_inflation_equivalence.jl .... Produces Table 1 and Figure 4 in supplementary materials
- _630_failure_rates.R .... Plots Figures 5 and 6 in supplementary materials
- _999_verify_results.jl .... Automated verification of ”Quick Spot Check” results
- estimateN.R .... Morgan-Ridout estimator as implemented by original author
- Project.toml .... Julia environment dependencies
- Manifest.toml .... Julia environment full snapshot

3 Execution Instructions

Note: All runtime estimates are based on execution on a machine with Intel i7-8665U CPU 1.90GHz and 16GB RAM.

To reproduce the results in the manuscript, follow this sequence:

3.1 Environment Setup

Ensure Julia and R are installed as per the versions in Section 1 or similar. In the root directory, run:

```
julia --project=. -e 'using Pkg; Pkg.instantiate()'
```

This ensures all UUID-locked dependencies are installed.

3.2 Data Generation

Execute the master simulation script:

```
julia --project=. _299_simulate_all.jl
```

This will populate the `_100_input/simulated/` directory with the raw simulated data amounting to approx. 90MB. Approximate runtime: 20 minutes.

3.3 Estimation (Master Script: `_399_estimate_all_simulated.jl`)

The estimation process is the most computationally intensive step. You may choose between two modes by editing the `CONFIG` variable in `_399_estimate_all_simulated.jl` at lines 15–18. This variable is an array of tuples; the second element of each tuple represents the `Intermediate Count`.

- **Mode A: Quick Spot Check (Recommended for Reviewers, Default Mode)**

  This mode considers a random subset of simulated datasets for each simulation setting, allowing for a faster verification of results.

  - *Setup:* Ensure the second element of each tuple in `CONFIG` (lines 15–18) is set to a desired number of intermediate datasets (default is `1` random dataset per setting).
  - *Output:* Saves files with an `_intermediate.csv` suffix (e.g. `estimates_0.5_intermediate.csv`).
  - *Runtime:* Approx. 2 hours at `1` dataset per setting.

- **Mode B: Full Reproduction**

  This mode processes all simulated datasets to fully replicate the results in the manuscript.

  - *Setup:* Change the second element of each tuple in `CONFIG` (lines 15–18) to `0`.
  - *Output:* Creates the full result `.csv` files used in the final figures.
  - *Runtime:* Approx. 96 hours.

  Note on Randomization: The selection of datasets for the Quick Spot Check is **not seeded**. Successive runs of Mode A will process different random subsets of the simulated data.

  Warning: Running `_399_estimate_all_simulated.jl` will overwrite existing files in the `_900_output/data/` directory. This includes the pre-computed full results provided in this supplement!

3.4 Empirical Analysis

To reproduce the results for the real-world datasets, run:

```
julia --project=. _400_estimate_data.jl
```

This will generate the results for Tables 2 and 3 in the main text. Approximate runtime: 1 hour.

3.5 Visualization (R Scripts)

All R scripts (of `_500_` and `_600_` series) generate plots and figures from the estimation results. These scripts can be executed in any order, provided the required `.csv` files exist in the `_900_output/data/` subdirectories.

By default, the R scripts are configured to process the full simulation results provided in the supplement. To visualize results from your own "Quick Spot Check" (Mode A), you must modify the `INTERMEDIATE` constant in each R script:

- `INTERMEDIATE <- FALSE` (Processes full results, default).
- `INTERMEDIATE <- TRUE`.

  This instructs the script to look for files with the `_intermediate.csv` suffix. The plot names will also have an `_intermediate` suffix to avoid overwriting existing figures.

  Note on Visual Discrepancies: If the `Intermediate Count` in script `_399_` was set to a low value (e.g., 1), the resulting boxplots, RMSE plots, and relative bias figures will be based only on those limited replicates. Consequently, these visualizations will look significantly different from the final figures presented in the manuscript.

3.6 Additional Verifications (Julia)

The following scripts verify theoretical equivalences discussed in the supplementary materials. These can be executed in any order and are entirely independent of the simulation and estimation results described in the previous sections.

- **Poisson Equivalence:**

  ```
  julia --project=. _600_poisson_equivalence.jl.
  ```

- **One-Inflation Equivalence:**

  ```
  julia --project=. _620_one_inflation_equivalence.jl.
  ```

Approximate runtime for each: 5 minutes.

3.7 Automated Spot-Check Verification

When comparing your ”Quick Spot Check” (Mode A) to the provided full simulation results, exact numerical equality is expected for all estimators except in rare cases for the proposed `MPLE-NB` estimator.

Due to the nature of the likelihood surface for the Negative Binomial model discussed in the main text, certain simulated datasets may result in:

1. **Collapsed Model:** Where $\hat{a}$ tends toward infinity (indicating limiting case of simple random sampling). Numerically, this manifests as large values of $\hat{a}$ (e.g., > 10) and $\hat{N}$ very close to the observed number of unique individuals sampled.

2. **Low Identifiability:** Flat likelihood surface (low information in data), resulting in low values of $\hat{a}$ and extremely large values for $\hat{N}$.

In these instances, small differences in optimization paths may lead to different terminal values. These should be considered theoretically equivalent to the ”limiting case” and ”boundary estimate” scenarios, respectively.

Additionally, the estimator `Morgan-Ridout` is also known to frequently produce boundary solutions and so may also exhibit extremely large $\hat{N}$ values in some cases.

To automate the comparison between your intermediate results and the provided full simulation results, run:

```
julia --project=. _999_verify_results.jl
```

The script follows a two-step verification logic:

1. **Standard Check:** Verify that $\hat{N}$ in the intermediate results data is within ± 2 of the full simulation results.

2. **Divergence Reporting:** Any cases that fall outside this margin are flagged. The script will print the `trial`, `T`, `N`, `a_hat`, and `N_hat` for both runs side-by-side, allowing for manual inspection of boundary/instability equivalence.

4 Data Provenance

4.1 Data Sources

- **Arboreal geckos** (`arboreal_geckos.txt`):

  Grimm, A., Gruber, B., and Henle, K. (2014). Reliability of different mark-recapture methods for population size estimation tested against reference population sizes constructed from field data. PLoS One, 9(6):e98840.

- **Cottontail rabbits** (`cottontail_rabbit.txt`):

  Edwards, W. R. and Eberhardt, L. (1967). Estimating cottontail abundance from livetrapping data. The Journal of Wildlife Management, 31(1):87–96.

- **Diabetes in Northern Italy** (`diabetes_italy.txt`):

  Bruno, G., Laporte, R. E., Merletti, F., Biggeri, A., McCarty, D., and Pagano, G. (1994). National diabetes programs: application of capture-recapture to count diabetes? Diabetes care, 17(6):548–556.

- **Domestic violence in the Netherlands** (`domestic_violence.txt`):

  van der Heijden, P., Cruyff, M., and Böhning, D. (2014). Capture recapture to estimate criminal populations. Encyclopedia of criminology and criminal justice. Berlin: Springer, pages 267–276.

- **Dutch illegal immigrants** (`dutch_illegal_immigrants.txt`):

  van der Heijden, P. G., Bustami, R., Cruyff, M. J., Engbersen, G., and Van Houwelingen, H. C. (2003). Point and interval estimation of the population size using the truncated poisson regression model. Statistical Modelling, 3(4):305–322.

- **Flare stars in Pleiades** (`flare_stars.txt`):

  Böhning, D. and Friedl, H. (2021). Population size estimation based upon zero-truncated, one-inflated and sparse count data: Estimating the number of dice snakes in graz and flare stars in the pleiades. Statistical Methods & Applications, 30(4):1197–1217.

  Ambartsumyan, V., Mirzoyan, L., Parsamyan, E. S., Chavushyan, O., and Erastova, L. (1970). Flare stars in the pleiades. Astrophysics, 6(1):1–10.

- **Forced labor worldwide** (`forced_labor.txt`):

  Böhning, D. (2016). Ratio Plot and Ratio Regression with Applications to Social and Medical Sciences. Statistical Science, 31(2):205 – 218.

- **Guiana dolphins** (`guiana_dolphins.txt`):

  Freitas-Nery, M. and Marino-Simao, S. (2012). Capture-recapture abundance estimate of guiana dolphins in southeastern brazil. Ciencias Marinas, 38(3):529–541.

- **Hepatitis A in Taiwan** (`hepatitis_taiwan.txt`):

  Chao, A., Tsay, P., Lin, S.-H., Shau, W.-Y., and Chao, D.-Y. (2001). The applications of capture-recapture models to epidemiological data. Statistics in medicine, 20(20):3123–3157

- **Heroin** and **methamphetamine users in Bangkok** (`heroin_usage.txt` and `meth_usage.txt`):

  Böhning, D., Suppawattanabodee, B., Kusolvisitkul, W., and Viwatwongkasem, C. (2004). Estimating the number of drug users in bangkok 2001: A capture–recapture approach using repeated entries in one list. European journal of epidemiology, 19:1075–1083

- **Rotterdom opiate users** (`opiate_users.txt`):

  Cruyff, M. J. and Van Der Heijden, P. G. (2008). Point and interval estimation of the population size using a zero-truncated negative binomial regression model. Biometrical Journal: Journal of Mathematical Methods in Biosciences, 50(6):1035–1050.

- **Scrapie holdings in France** (`scrapie_france.txt`):

  Vergne, T., Vilas, V. J. D. R., Cameron, A., Dufour, B., and Grosbois, V. (2015). Capture–recapture approaches and the surveillance of livestock diseases: A review. Preventive veterinary medicine, 120(3-4):253–264.

- **Taxicabs in Edinburgh** (`taxicab_edinburgh.txt`):

  Carothers, A. (1973). Capture-recapture methods applied to a population with known parameters. The Journal of Animal Ecology, pages 125–146.

4.2 Intellectual Property and Licensing

Intellectual property rights remain with the original authors or journals as specified in the primary sources. To the best of the author's knowledge, these data were originally published as open-access supplements or were made available for public academic use.

4.3 Data Dictionary

All `.txt` files in the `datasets/` folder follow a consistent format: comma-separated text with two columns and no header row. The first column represents the number of captures (frequency), while the second column indicates the count of individuals with that capture frequency.

The file `sampling_effort.csv` contains summary statistics and metadata required for the empirical analysis.

- `true_N` provides true population sizes if known; `T` specifies the number of sample draws in each study.
- `sum_n`, `mean_n`, `n1--n18` describe the sum of sample sizes, mean sample size over all sampling occasions and individual sample sizes per sample draw, respectively.
- `coef_var_n` provides the coefficient of variation of sample sizes for each dataset.