Package 'HiddenTrunc'

November 12, 2026

Type Package

Title Hidden Truncation Probability Distributions and Statistical Inference

Version 0.2.1

Author Indranil Ghosh [aut, cre]

Maintainer Indranil Ghosh <ghoshi@uncw.edu>

Description Probability functions, random generation, likelihood-based inference, diagnostics, simulation, and Bayesian computation for probability models generated through hidden truncation. The initial release implements the upper-hidden-truncated Pareto type IV model and the two-sided hidden-truncated Arnold-Strauss bivariate exponential model, together with a generic hidden-truncation engine.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, graphics, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

URL https://github.com/ginni88/HiddenTrunc

BugReports https://github.com/ginni88/HiddenTrunc/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2026-11-12 19:00:08 UTC

Contents

HiddenTrunc-package 2
dhtbexp 2
dhtp4 3
fit_htbexp 4

fit_htp4 5
floyd_river 6
ht_density 6

Index **8**

`HiddenTrunc-package` *Hidden Truncation Probability Distributions and Inference*

Description

Probability functions, simulation, likelihood inference and Bayesian computation for hidden-truncation models.

Details

The initial implementation contains upper hidden-truncated Pareto IV and two-sided hidden-truncated Arnold-Strauss exponential models, plus generic hidden-truncation utilities.

`dhtbexp` *Hidden-truncated Arnold-Strauss exponential distribution*

Description

Functions for the two-sided hidden-truncated Arnold-Strauss bivariate exponential construction.

Usage

```
dhtbexp(x, a, b, c, d1, d2, log = FALSE)
phtbexp(q, a, b, c, d1, d2,
        lower.tail = TRUE, log.p = FALSE)
qhtbexp(p, a, b, c, d1, d2)
rhtbexp(n, a, b, c, d1, d2)
```

Arguments

`x`	Numeric evaluation points for density.
`q`	Numeric quantiles for the CDF.
`p`	Numeric probabilities for the quantile function.
`n`	Number of generated observations.
`a`	Positive exponential rate parameter.
`b`	Positive exponential rate parameter.
`c`	Positive interaction parameter.
`d1`	Lower hidden truncation bound, non-negative.

d2	Upper hidden truncation bound, strictly larger than d1.
log	Logical; if TRUE, return log-density.
lower.tail	Logical; use the lower tail when TRUE.
log.p	Logical; return log-probabilities when TRUE.

Value

Numeric vector of density, CDF, quantile, or random values.

---

dhtp4 *Hidden-truncated Pareto IV distribution*

Description

Density, cumulative distribution, quantile, random generation, and hazard functions for the hidden-truncated Pareto IV model.

Usage

```
dhtp4(x, alpha, sigma, theta, delta1, delta2, log = FALSE)
phtp4(q, alpha, sigma, theta, delta1, delta2,
      lower.tail = TRUE, log.p = FALSE)
qhtp4(p, alpha, sigma, theta, delta1, delta2,
      lower.tail = TRUE, log.p = FALSE)
rhtp4(n, alpha, sigma, theta, delta1, delta2)
hhtp4(x, alpha, sigma, theta, delta1, delta2)
```

Arguments

x	Numeric evaluation points (for the density or hazard).
q	Numeric quantiles for the CDF.
p	Numeric probabilities for the quantile function.
n	Non-negative number of observations to generate.
alpha	Positive Pareto IV tail-shape parameter.
sigma	Positive scale parameter.
theta	Positive selection parameter.
delta1	Positive power parameter.
delta2	Positive parameter controlling the hidden-selection transform.
log	Logical; return logarithm of the density.
lower.tail	Logical; return lower-tail probabilities when TRUE.
log.p	Logical; probabilities are on the logarithmic scale when TRUE.

Value

Numeric vector of evaluated values, quantiles, simulated observations, or hazards.

`fit_htbexp` *Inference for hidden-truncated exponential distribution*

Description

Maximum likelihood estimation, likelihood evaluation, and Bayesian sampling for the two-sided hidden-truncated exponential model.

Usage

```
fit_htbexp(x, start = NULL, control = list(maxit = 5000))
loglik_htbexp(par, x, transformed = FALSE)
bayes_htbexp(x, niter = 10000, burn = 2000, thin = 5,
             init = c(a = 1, b = 1, c = 0.2, d1 = 0.1, d2 = 1),
             gamma_shape = c(1, 1, 1),
             gamma_rate = c(1, 1, 1),
             delta = max(10, init[5] * 2),
             proposal_sd = rep(0.12, 5), seed = NULL)
```

Arguments

`x`	Positive numeric observations.
`start`	Optional five-parameter starting vector for MLE.
`control`	Optimization control list.
`par`	Parameter vector for the log-likelihood.
`transformed`	Logical; use transformed optimization coordinates when TRUE.
`niter`	Total number of MCMC iterations.
`burn`	Number of initial draws to discard.
`thin`	Thinning interval.
`init`	Initial positive five-parameter MCMC state with ordered truncation bounds.
`gamma_shape`	Shape hyperparameters for three Gamma priors.
`gamma_rate`	Rate hyperparameters for three Gamma priors.
`delta`	Upper MCMC truncation bound, exceeding the initial d2.
`proposal_sd`	Standard deviations for five Gaussian random-walk proposals.
`seed`	Optional random seed.

Value

A fitted-model object, numeric log-likelihood, or a list containing posterior draws and diagnostics.

fit_htp4 *Inference and simulation for hidden-truncated Pareto IV*

Description

Maximum likelihood fitting, log-likelihood, likelihood-ratio comparison, and Monte Carlo simulation under the Pareto IV hidden-truncation model.

Usage

```
fit_htp4(x, start = NULL, method = "L-BFGS-B",
          control = list(maxit = 3000), hessian = TRUE,
          delta2_fixed = 3, n_starts = 5)
loglik_htp4(par, x, transformed = FALSE)
lrt_htp4(x, fit = NULL, B = 0, seed = NULL)
sim_htp4(n = c(50, 100, 200), R = 100,
         par = c(alpha = 3, sigma = 2, theta = 1,
                 delta1 = 2, delta2 = 3), seed = NULL)
```

Arguments

x	Positive numeric sample.
start	Optional positive five-parameter starting vector for optimization.
method	Optimization algorithm passed to the fitter.
control	List of optimization control parameters.
hessian	Logical; whether to compute a numerical Hessian.
delta2_fixed	Positive value fixed for the non-identifiable delta2 coordinate.
n_starts	Number of optimization starting points.
par	Five-parameter model vector; for loglik_htp4, optionally on log scale.
transformed	Logical; whether par contains log-transformed parameters.
fit	Previously fitted htfit model; if NULL a model is fitted.
B	Number of bootstrap replications for the likelihood ratio test.
seed	Optional random seed.
n	Vector of simulated sample sizes.
R	Number of simulation replications.

Value

fit_htp4 returns an htfit object; loglik_htp4 returns a log-likelihood; lrt_htp4 returns a test object; sim_htp4 returns a simulation summary.

floyd_river — *Floyd River data and Pareto IV model assessments*

Description

Access Floyd River annual discharge data; compute a Kolmogorov-Smirnov discrepancy with optional parametric bootstrap; or obtain percentile bootstrap confidence intervals.

Usage

```
floyd_river()
gof_htp4(x, fit = NULL, B = 0, seed = NULL)
boot_ci_htp4(x, fit = NULL, B = 499,
             level = 0.95, seed = NULL)
```

Arguments

x	Positive numeric sample used to fit or assess the model.
fit	Optional htfit object. If NULL, fit_htp4 is called.
B	Number of parametric bootstrap replicates. Zero disables the bootstrap in gof_htp4.
seed	Optional seed for reproducibility.
level	Confidence level strictly between zero and one.

Value

floyd_river returns a numeric vector; gof_htp4 returns an htgof object; boot_ci_htp4 returns a data frame of confidence bounds.

ht_density — *Generic hidden-truncation utilities*

Description

Generic construction of hidden-truncated densities and inference using an explicit baseline density and selection probability.

Usage

```
ht_density(x, fx, selection_prob, normalizer, log = FALSE, ...)
dhidden(x, fx, selection_prob, normalizer, ...)
phidden(q, fx, selection_prob, normalizer, lower = 0, ...)
rhidden(n, fx, selection_prob, normalizer,
        lower = 0, upper_start = 1, ...)
ht_loglik(par, x, density, ...)
ht_mle(par, x, loglik, transform = exp, ...)
```

Arguments

x	Numeric evaluation points or observed data, depending on the function.
q	Quantiles at which to evaluate the CDF.
n	Number of random observations.
fx	Callable function for the baseline density.
selection_prob	Callable function for conditional selection probabilities.
normalizer	Positive normalization probability for the selection model.
log	Logical; return the log-density when TRUE.
lower	Lower integration/inversion bound.
upper_start	Positive starting upper bound for inverse CDF search.
par	Parameter vector for optimization or likelihood evaluation.
density	Density function accepting observations and a parameter vector.
loglik	Log-likelihood function for maximum likelihood estimation.
transform	Transformation mapping optimization coordinates to parameters.
...	Additional arguments passed to the density, selection, or likelihood functions.

Value

Density/CDF values, random observations, log-likelihood, or an optimization result, as appropriate.

Index

bayes_htbexp *(fit_htbexp)*, 4
boot_ci_htp4 *(floyd_river)*, 6

dhidden *(ht_density)*, 6
dhtbexp, 2
dhtp4, 3

fit_htbexp, 4
fit_htp4, 5
floyd_river, 6

gof_htp4 *(floyd_river)*, 6

hhtp4 *(dhtp4)*, 3
HiddenTrunc *(HiddenTrunc-package)*, 2
HiddenTrunc-package, 2
ht_density, 6
ht_loglik *(ht_density)*, 6
ht_mle *(ht_density)*, 6

loglik_htbexp *(fit_htbexp)*, 4
loglik_htp4 *(fit_htp4)*, 5
lrt_htp4 *(fit_htp4)*, 5

phidden *(ht_density)*, 6
phtbexp *(dhtbexp)*, 2
phtp4 *(dhtp4)*, 3

qhtbexp *(dhtbexp)*, 2
qhtp4 *(dhtp4)*, 3

rhidden *(ht_density)*, 6
rhtbexp *(dhtbexp)*, 2
rhtp4 *(dhtp4)*, 3

sim_htp4 *(fit_htp4)*, 5