Package ‘HTDV’

May 7, 2026

Type Package

Title Hypothesis Testing for Dependent Variables with Unbalanced Data

Version 0.2.0

Description Implements hierarchical Bayesian inference, robust frequentist inference, and distribution-free inference for dependent and unbalanced data under strong-mixing conditions. Supports triangular-array, weighted-sum and mixingale convergence regimes with Whittle and composite likelihoods, heteroskedasticity-and-autocorrelation-consistent variance estimation, block bootstrap with automatic block length, fixed-bandwidth HAR inference, adaptive conformal prediction, Bayesian decision under Region of Practical Equivalence, bridge-sampling Bayes factors, and predictive comparison via the Widely Applicable Information Criterion and leave-future-out cross-validation. Methods follow Andrews (1991) <doi:10.2307/2938229>, Kiefer and Vogelsang (2005) <doi:10.1017/S0266466605050565>, Patton, Politis and White (2009) <doi:10.1080/07474930802459016>, Vehtari, Gelman and Gabry (2017) <doi:10.1007/s11222-016-9696-4>, Kruschke (2018) <doi:10.1177/2515245918771304>, and Gibbs and Candes (2021) <doi:10.48550/arXiv.2106.00170>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports rstan (>= 2.26.0), stats, parallel,

Suggests bayesplot, bridgesampling, knitr, loo, posterior, readxl, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.3

URL `https://github.com/IsadoreNabi/HTDV`

BugReports `https://github.com/IsadoreNabi/HTDV/issues`

NeedsCompilation no

Author Jose Mauricio Gomez Julian [aut, cre] (ORCID:
<https://orcid.org/0009-0000-2412-3150>)

Maintainer Jose Mauricio Gomez Julian <isadorenabi@pm.me>

Repository CRAN

Date/Publication 2026-05-02 09:50:14 UTC

Contents

`htdv_bf` *Bridge-Sampling Bayes Factor*

Description

Computes the Bayes factor between two fitted `htdv_fit` objects via bridge sampling. Requires the `bridgesampling` package.

Usage

```
htdv_bf(fit1, fit0, ...)
```

Arguments

`fit1`	Model in the numerator.
`fit0`	Model in the denominator.
`...`	Additional arguments passed to `bridgesampling::bridge_sampler`.

Value

A list with `bf10`, `log_bf10`, `logml1`, `logml0`, and `error_percentage`.

References

Meng, X.-L., & Wong, W.H. (1996). Simulating Ratios of Normalizing Constants. Statistica Sinica 6(4): 831-860. Gronau, Q.F. et al. (2017). A Tutorial on Bridge Sampling. JMP 81: 80-97.

Examples

```
if (requireNamespace("bridgesampling", quietly = TRUE)) {
  x <- rnorm(50)
  f1 <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
                refresh = 0, seed = 1)
  f0 <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
                refresh = 0, seed = 2)
  htdv_bf(f1, f0)$bf10
}
```

`htdv_boot` *Block Bootstrap with Automatic Block Length*

Description

Implements circular and stationary block bootstraps (Politis-Romano 1992, 1994) with Patton-Politis-White (2009) automatic block length selection.

Usage

```
htdv_boot(
  x,
  statistic,
  R = 1999L,
  type = c("circular", "stationary"),
  block_length = "auto",
  level = 0.95
)
```

Arguments

`x`	Numeric vector.
`statistic`	Function `function(x)` returning a scalar or numeric vector.
`R`	Number of bootstrap replicates.
`type`	`"circular"` or `"stationary"`.
`block_length`	Either `"auto"` (Patton-Politis-White) or a positive integer.
`level`	Confidence level for intervals.

Value

A list with `t0` (statistic on original data), `t_star` (bootstrap distribution), `ci_percentile`, `ci_basic`, `ci_studentized`, and `block_length`.

References

Politis, D.N., & Romano, J.P. (1994). The Stationary Bootstrap. JASA 89(428): 1303-1313. Patton, A., Politis, D.N., & White, H. (2009). Correction to Automatic Block-Length Selection for the Dependent Bootstrap. Econometric Reviews 28(4): 372-375.

Examples

```
set.seed(1)
x <- arima.sim(model = list(ar = 0.5), n = 200, rand.gen = rnorm)
out <- htdv_boot(as.numeric(x), mean, R = 500, type = "stationary")
out$ci_percentile
```

`htdv_composite` *Composite Log-Likelihood for Dependent Data*

Description

Pairwise or k-tuple composite log-likelihood (Varin, Reid and Firth 2011) for strictly stationary sequences.

Usage

```
htdv_composite(x, density_fun, theta, k = 1L, log = FALSE)
```

Arguments

`x`	Numeric vector.
`density_fun`	Function `function(block, theta)` returning the joint density of a `(k+1)`-block of consecutive observations.
`theta`	Numeric parameter vector.
`k`	Window width (pairwise if `k = 1`).
`log`	Logical; if `TRUE`, `density_fun` already returns log densities.

Value

A list with `loglik` (scalar), `n_blocks`, and `block_values`.

References

Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. Statistica Sinica, 21(1), 5-42.

Examples

```
x <- arima.sim(model = list(ar = 0.4), n = 200, rand.gen = rnorm)
gauss_ar1 <- function(block, theta) {
  phi <- theta[1]; sigma2 <- theta[2]
  a <- block[1]; b <- block[2]
  dnorm(a, 0, sqrt(sigma2 / (1 - phi^2)), log = TRUE) +
    dnorm(b, phi * a, sqrt(sigma2), log = TRUE)
}
htdv_composite(as.numeric(x), gauss_ar1, c(0.4, 1), k = 1, log = TRUE)$loglik
```

`htdv_conformal` *Adaptive Conformal Inference for Dependent Data*

Description

Online adaptive conformal prediction intervals with gradient update on the miscoverage rate (Gibbs-Candes 2021). Provides long-run marginal coverage without exchangeability.

Usage

```
htdv_conformal(
  x,
  predictor,
  residual_fun = function(e) abs(e),
  alpha_target = 0.1,
  lambda = 0.05,
  burn_in = 20L
)
```

Arguments

`x`	Numeric vector of sequential observations.
`predictor`	Function `function(history)` returning a scalar one-step point prediction.
`residual_fun`	Function of residuals to calibrate the score; default absolute residual.
`alpha_target`	Target miscoverage rate.
`lambda`	Step size for the gradient update.
`burn_in`	Integer; the first `burn_in` observations are used only to warm up the predictor.

Value

A list with `intervals` (matrix of lower/upper), `coverage` (empirical), `alpha_path`, and `point_predictions`.

References

Gibbs, I., & Candes, E. (2021). Adaptive Conformal Inference under Distribution Shift. NeurIPS 34: 1660-1672.

Examples

```
x <- arima.sim(model = list(ar = 0.6), n = 200, rand.gen = rnorm)
pred <- function(hist) if (length(hist) >= 1) hist[length(hist)] else 0
out <- htdv_conformal(as.numeric(x), pred, alpha_target = 0.1,
                      lambda = 0.05, burn_in = 20)
out$coverage
```

`htdv_diagnostics` *MCMC Diagnostics for htdv_fit Objects*

Description

Computes split-Rhat, bulk and tail ESS, E-BFMI, divergence count, and treedepth saturation; all standard HMC diagnostics (Vehtari et al. 2021, Betancourt 2016).

Usage

```
htdv_diagnostics(
  fit,
  rhat_threshold = 1.01,
  ess_threshold = 400,
  bfmi_threshold = 0.3
)
```

Arguments

`fit`	An `htdv_fit` object.
`rhat_threshold`	Convergence threshold for Rhat.
`ess_threshold`	Minimum effective sample size.
`bfmi_threshold`	Minimum acceptable energy-BFMI.

Value

A list with `rhat`, `ess_bulk`, `ess_tail`, `bfmi`, `divergences`, `max_treedepth`, `passed` (logical), and `failures` (character vector).

References

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Buerkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved Rhat for assessing convergence of MCMC. Bayesian Analysis 16(2): 667-718. Betancourt, M. (2016). Diagnosing Suboptimal Cotangent Disintegrations. <doi:10.48550/arXiv.1604.00695>.

Examples

```
x <- rnorm(50)
fit <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
                refresh = 0, seed = 1)
htdv_diagnostics(fit)$passed
```

`htdv_empirical_benchmarks` *HTDV Empirical Benchmarks Against Published References*

Description

Output of the three-dataset external validation protocol of Section 12-ter of the companion paper. Each dataset is run against a pre-existing reference result from the published literature on the same vintage, with the three HTDV inferential layers (HAR, bootstrap, hierarchical Bayesian) reporting their point estimate, 95% interval, and a Bayesian diagnostic-pass flag.

Usage

```
htdv_empirical_benchmarks
```

Format

A data frame with three rows (one per dataset) and the following columns:

dataset Name of the benchmark dataset.

reference Citation of the reference paper.

n Sample size after preprocessing.

sample Sample window (string).

reference_value Reference point value.

reference_se Reference standard error (`NA` when the reference does not provide one).

har_point, har_ci_lo, har_ci_hi HAR-Wald point and 95% CI.

bayes_point, bayes_ci_lo, bayes_ci_hi Bayesian point and 95% credible interval.

bayes_diag_pass, bayes_rhat_max, bayes_ess_min, bayes_divergences Bayesian HMC diagnostics.

boot_point, boot_ci_lo, boot_ci_hi Stationary block bootstrap point and 95% percentile CI.

agreement_har, agreement_bayes, agreement_boot Per-layer agreement verdict: `"agreement"`, `"htdv-strict (above ref)"`, or `"htdv-strict (below ref)"`.

Details

The three benchmarks (E1, E2, E3) and their references are:

- **E1.** Post-1984 US CPI inflation against Stock and Watson (2007). FRED-MD vintage 2026-03, variable `CPIAUCSL` transformed to annualized monthly inflation.
- **E2.** Long-run mean of log-CAPE against Campbell and Shiller (1998). Robert Shiller's online historical dataset.
- **E3.** US vs. Canada 10-year government bond yield differential against the iid Welch baseline. FRED series `GS10` and `IRLTLT01CAM156N`, common sample 1990-01 to latest.

Source

Generated by `run_benchmarks.R` of the companion paper repository. See `vignette("HTDV-validation")` for the methodological narrative and the persistence-ladder reading.

`htdv_envelope` *Berger-Robust Envelope Posterior*

Description

Builds an envelope posterior over a list of `htdv_fit` objects by entropy-maximizing mixture weights subject to predictive consistency (Berger 1994).

Usage

```
htdv_envelope(fits, target = "theta")
```

Arguments

`fits`	List of `htdv_fit` objects on the same data.
`target`	Name of the Stan parameter over which to build the envelope.

Value

A list with `draws` (envelope draws for the target parameter), `weights`, `intervals`, and `component_intervals`.

References

Berger, J.O. (1994). An Overview of Robust Bayesian Analysis. Test 3(1): 5-124.

Examples

```
x <- rnorm(50)
f1 <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
               refresh = 0, seed = 1)
f2 <- htdv_fit(x, model = "wsc", chains = 2, iter = 500,
               refresh = 0, seed = 2)
htdv_envelope(list(f1, f2), target = "theta")$intervals
```

`htdv_equivalence_constants`

Numerical Constants for TAC/WSC/MPC Metric Equivalence

Description

Computes the explicit finite-sample constants K_TAC(gamma, q), K_MPC(gamma) and the unified two-sided envelope (c_L, c_U) from Theorems 5-8 of the companion paper.

Usage

```
htdv_equivalence_constants(gamma, q = 6, n = 100, C_alpha = 1)
```

Arguments

`gamma`	Mixing decay rate; must satisfy `gamma > 1`.
`q`	Moment order used in Ibragimov-Rio covariance bound; must satisfy `q > 2` and `gamma * (1 - 2/q) > 1`.
`n`	Sample size used for the two-sided envelope.
`C_alpha`	Strong-mixing constant (default 1).

Value

A list with components `K_TAC`, `K_MPC`, `c_L`, `c_U`, `b_optimal`, `m_optimal`, and `regime` (a short label describing which method is favored).

Examples

```
htdv_equivalence_constants(gamma = 2, q = 6, n = 60)
```

`htdv_fit` *Hierarchical Bayesian Fit for Dependent Variables*

Description

Fits the hierarchical Bayesian model described in the companion paper via Hamiltonian Monte Carlo (NUTS) using **rstan**. Nuisance parameters `sigma_inf`, `gamma_mix`, and (for WSC) `b` carry priors and are integrated over.

Usage

```
htdv_fit(
  x,
  model = c("tac", "wsc", "mpc", "whittle", "composite"),
  chains = 4L,
  iter = 4000L,
  warmup = iter%/%2L,
  refresh = max(iter%/%10L, 1L),
  adapt_delta = 0.99,
  max_treedepth = 12L,
  seed = NULL,
  ...
)

htdv_tac(x, ...)

htdv_wsc(x, ...)

htdv_mpc(x, ...)
```

Arguments

`x`	Numeric vector of observations.
`model`	One of `"tac"`, `"wsc"`, `"mpc"`, `"whittle"`, `"composite"`.
`chains`	Number of HMC chains.
`iter`	Total iterations per chain (warmup plus sampling).
`warmup`	Number of warmup iterations.
`refresh`	Stan progress refresh rate (0 suppresses output).
`adapt_delta`	HMC adaptation target acceptance.
`max_treedepth`	Maximum NUTS treedepth.
`seed`	Optional integer seed forwarded to **rstan**.
`...`	Additional arguments forwarded to `rstan::sampling`.

Value

An object of class `htdv_fit`, a list with components `stanfit`, `model`, `data`, `n`, `call`.

References

Hoffman, M.D., & Gelman, A. (2014). The No-U-Turn Sampler. JMLR 15(1): 1593-1623. Betancourt, M. (2016). Diagnosing Suboptimal Cotangent Disintegrations in HMC. <doi:10.48550/arXiv.1604.00695>.

Examples

```
x <- rnorm(50)
fit <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
                refresh = 0, seed = 1)
class(fit)
```

`htdv_fixedb` *Fixed-Bandwidth HAR Inference (Kiefer-Vogelsang 2005)*

Description

Fixed-b HAR Wald test where the bandwidth is a fixed fraction of the sample size. Critical values are obtained by simulation from the asymptotic functional distribution.

Usage

```
htdv_fixedb(
  x,
  theta0 = 0,
  B = 0.1,
  kernel = c("bartlett", "parzen", "qs"),
  sims = 5000L,
  alpha = 0.05
)
```

Arguments

`x`	Numeric vector.
`theta0`	Null value for the mean (default 0).
`B`	Fixed bandwidth fraction in (0, 1].
`kernel`	Kernel choice (see `htdv_lrv`).
`sims`	Number of Monte Carlo draws for the critical-value simulation.
`alpha`	Significance level.

Value

A list with `statistic`, `critical_value`, `p_value`, `reject`, `bandwidth`, and `kernel`.

References

Kiefer, N.M., & Vogelsang, T.J. (2005). A New Asymptotic Theory for Heteroskedasticity-Autocorrelation Robust Tests. Econometric Theory 21(6): 1130-1164.

Examples

```
set.seed(1)
x <- arima.sim(model = list(ar = 0.5), n = 200, rand.gen = rnorm)
htdv_fixedb(as.numeric(x), theta0 = 0, B = 0.2, sims = 1000)$p_value
```

`htdv_lrv` *Long-Run Variance Estimation (HAC)*

Description

Heteroskedasticity-and-autocorrelation-consistent long-run variance with data-driven Andrews (1991) optimal bandwidth, Bartlett, Parzen, and Quadratic-Spectral kernels.

Usage

```
htdv_lrv(
  x,
  kernel = c("bartlett", "parzen", "qs"),
  bandwidth = "andrews",
  prewhiten = FALSE
)
```

Arguments

`x`	Numeric vector of observations.
`kernel`	Kernel choice: `"bartlett"`, `"parzen"`, or `"qs"` (Quadratic-Spectral).
`bandwidth`	Either `"andrews"` (data-driven) or a positive numeric scalar.
`prewhiten`	Logical; if `TRUE`, AR(1) prewhiten-recolor (Andrews-Monahan 1992).

Value

A list with components `lrv` (long-run variance estimate), `bandwidth`, `kernel`, and `ar_coef` (only if prewhitened).

References

Andrews, D.W.K. (1991). Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. Econometrica 59(3): 817-858. Newey, W.K. & West, K.D. (1987). A Simple, Positive Semi-Definite Heteroskedasticity and Autocorrelation Consistent Covariance Matrix. Econometrica 55(3): 703-708.

Examples

```
set_seed_user <- 42
x <- arima.sim(model = list(ar = 0.6), n = 200, rand.gen = rnorm)
out <- htdv_lrv(as.numeric(x), kernel = "qs", bandwidth = "andrews")
out$lrv
```

`htdv_ppc` *Posterior-Predictive Checks on Dependence Statistics*

Description

Replicates autocorrelation, Ljung-Box, spectral discrepancy, and a detrended-fluctuation Hurst exponent under the posterior and computes posterior-predictive p-values.

Usage

```
htdv_ppc(fit, x_obs, lag_max = 10L)
```

Arguments

`fit`	An `htdv_fit` object whose posterior-predictive replicates are stored as parameter `x_rep`.
`x_obs`	The observed data vector.
`lag_max`	Maximum lag for autocorrelation.

Value

A list with `p_acf`, `p_ljung_box`, `p_spectral`, `p_hurst`.

Examples

```
x <- rnorm(50)
fit <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
                refresh = 0, seed = 1)
htdv_ppc(fit, x)
```

`htdv_rope` *Region-of-Practical-Equivalence (ROPE) Decision*

Description

Posterior decision under Kruschke's (2018) Region of Practical Equivalence.

Usage

```
htdv_rope(draws, rope, level = 0.95)
```

Arguments

`draws`	Numeric vector of posterior draws of the parameter of interest.
`rope`	Length-2 numeric vector `c(lower, upper)` defining the practical-equivalence region.
`level`	Credibility level for the highest-density interval.

Value

A list with `decision` (`"accept"`, `"reject"`, `"undecided"`), `hdi`, `rope`, and `prob_in_rope`.

References

Kruschke, J.K. (2018). Rejecting or Accepting Parameter Values in Bayesian Estimation. Advances in Methods and Practices in Psychological Science 1(2): 270-280.

Examples

```
draws <- rnorm(4000, mean = 0.05, sd = 0.1)
htdv_rope(draws, rope = c(-0.1, 0.1), level = 0.95)$decision
```

`htdv_simstudy` *Factorial Monte Carlo Simulation Study for Dependent-Unbalanced Data*

Description

Runs the pre-registered factorial Monte Carlo study of Section 12-bis of the companion paper. The study crosses four factors (per-group sample size, AR(1) coefficient, innovation tail, imbalance ratio) and evaluates three inferential layers (HAR-Wald, stationary block bootstrap, hierarchical Bayesian HMC) on each replication.

Usage

```
htdv_simstudy(
  n_grid = c(40L, 80L, 200L, 500L),
  phi_grid = c(0, 0.3, 0.6, 0.85),
  tail_grid = c("normal", "t5", "t3", "t2_1"),
  imb_grid = c(1, 1.5, 3, 6),
  delta_grid = c(0, 0.1, 0.25, 0.5),
  R = 500L,
  seed = 20260422L,
  n_cores = 1L,
  layers = c("har", "boot", "bayes"),
  bayes_chains = 2L,
  bayes_iter = 600L,
  bayes_warmup = 300L,
  boot_R = 999L,
  rope = c(-0.1, 0.1),
  alpha = 0.05,
  out_dir = NULL,
  progress = TRUE
)
```

Arguments

`n_grid`	Integer vector of primary sample sizes n_1.
`phi_grid`	Numeric vector of AR(1) coefficients in $(-1, 1)$.
`tail_grid`	Character vector; any subset of `"normal"`, `"t5"`, `"t3"`, `"t2_1"`.
`imb_grid`	Numeric vector of imbalance ratios n_1/n_2.
`delta_grid`	Numeric vector of location shifts in units of σ_∞.
`R`	Integer number of Monte Carlo replications per cell.
`seed`	Integer master seed.
`n_cores`	Integer number of workers for cell-level parallelism. On non-Unix platforms parallelism falls back to serial.
`layers`	Character vector; any subset of `c("har","boot","bayes")`.
`bayes_chains`	Number of HMC chains per Bayesian fit.
`bayes_iter`	Total HMC iterations (warmup + sampling).
`bayes_warmup`	HMC warmup iterations.
`boot_R`	Bootstrap replicates per call.
`rope`	Length-2 numeric vector; ROPE for the Bayesian decision on the raw (non-standardized) delta.
`alpha`	Significance level (nominal size of tests; $1 - \alpha$ is the coverage target for intervals).
`out_dir`	Optional directory for per-cell incremental RDS results. Each cell is written to its own `cell_<id>.rds` so that concurrent workers cannot collide on a shared

	file. Final aggregation reads the directory back, so successful cells are preserved even if some workers later die. If `NULL`, no per-cell artifacts are written and the only result is the in-memory return value.
`progress`	Logical; print a one-line status per completed cell.

Value

A data frame with one row per (cell, replication, layer). Columns: `cell_id`, `n`, `phi`, `tail`, `imb`, `n1`, `n2`, `delta`, `sigma_inf`, `layer`,`replicate`, `reject`, `ci_lo`, `ci_hi`, `estimate`, `covered`, `ci_length`,`rhat_max`, `ess_min`, `divergences`, `diag_pass`, `runtime_sec`.

Data-generating process

Two independent AR(1) series $x_t^{(1)}$ of length $n_1 = n$ and $x_t^{(2)}$ of length $n_2 = \max(2, \mathrm{round}(n/\mathrm{imb}))$ are generated. The innovations are drawn from `rnorm`, scaled Student-$t_{(5)}$, scaled Student-$t_{(3)}$ or scaled Student-$t_{(2.1)}$ so that the nominal innovation variance is one when the law has a finite second moment. Group one has mean zero; group two has mean $\Delta \cdot \sigma_\infty$ where $\sigma_\infty = 1/\sqrt{1-\phi^2}$.

Inferential layers

- **HAR-Wald (`"har"`).** Welch-style two-sample statistic $z = (\bar{x}_1 - \bar{x}_2)/\sqrt{\hat{\sigma}_1^2/n_1 + \hat{\sigma}_2^2/n_2}$ with long-run variances estimated via `htdv_lrv` (QS kernel, Andrews bandwidth) on each group's demeaned series. Rejection at asymptotic χ_1^2 critical value.
- **Stationary block bootstrap (`"boot"`).** Each group resampled independently using `htdv_boot` with `block_length = "auto"` (Patton- Politis-White). Percentile interval on $\bar{x}_1^* - \bar{x}_2^*$.
- **Hierarchical Bayesian (`"bayes"`).** Two-sample AR(1) Stan model (`simstudy_two_sample.stan`); posterior interval on $\delta = \alpha_2 - \alpha_1$. A run is flagged `diag_pass = TRUE` only if $\hat{R} < 1.01$, bulk and tail ESS above 200, and zero divergences.

References

Kiefer, N.M., & Vogelsang, T.J. (2005). Econometric Theory 21(6): 1130-1164. Patton, A., Politis, D.N., & White, H. (2009). Econometric Reviews 28(4): 372-375. Kruschke, J.K. (2018). Advances in Methods and Practices in Psychological Science 1(2): 270-280.

Examples

```
res <- htdv_simstudy(n_grid = c(40, 80),
                     phi_grid = c(0, 0.6),
                     tail_grid = c("normal"),
                     imb_grid = c(1, 3),
                     delta_grid = c(0, 0.25),
                     R = 5L, n_cores = 1L,
                     layers = c("har", "boot"))
head(htdv_simstudy_summary(res))
```

`htdv_simstudy_summary` *Aggregate a Simulation-Study Output*

Description

Computes empirical size (or rejection rate), coverage, and mean interval length per (cell x layer) from the raw output of `htdv_simstudy`.

Usage

```
htdv_simstudy_summary(res, alpha = 0.05)
```

Arguments

`res`	Data frame returned by `htdv_simstudy`.
`alpha`	Nominal size used in the study (must match what `htdv_simstudy` was called with).

Value

A data frame with one row per (cell, layer) and columns `n`, `phi`, `tail`, `imb`, `delta`, `layer`, `n_reps`, `reject_rate`, `coverage`,`mean_ci_length`, `mean_runtime`, `diag_pass_rate`.

Examples

```
res <- htdv_simstudy(n_grid = c(40), phi_grid = c(0.3),
                     tail_grid = "normal", imb_grid = 1,
                     delta_grid = c(0, 0.3), R = 5,
                     layers = c("har"))
htdv_simstudy_summary(res)
```

`htdv_simstudy_warnings`
Flag Simulation-Study Cells in the Limit-of-Identification Zone

---

Description

Reads an aggregated `htdv_simstudy_summary` output and returns the subset of cells where the Bayesian diagnostic-pass rate falls below a user-set threshold. Such cells are typically located in the corner of the design where strong autocorrelation meets a small sample (high ϕ and low n); the AR(1) likelihood there approaches its limit of identification, MCMC diagnostics flag the resulting posterior, and practitioners should treat the cell with extra caution—either by increasing HMC iterations, tightening the priors on ϕ, or falling back to the conformal anchor whose validity does not require identifiability.

Usage

```
htdv_simstudy_warnings(summ, threshold = 0.7)
```

Arguments

`summ`	Data frame from `htdv_simstudy_summary`.
`threshold`	Minimum acceptable Bayesian `diag_pass_rate`. Default `0.7`.

Value

A data frame with the columns `n`, `phi`, `tail`, `imb`, `delta`, `layer`, `diag_pass_rate`, restricted to Bayesian rows whose pass rate is below `threshold`, plus an integer column `flag_index` (sequential identifier of flagged cells, useful for cross-referencing in reports and in vignettes).

Examples

```
res <- htdv_simstudy(n_grid = c(40, 200),
                     phi_grid = c(0, 0.85),
                     tail_grid = "normal",
                     imb_grid = c(1, 6),
                     delta_grid = c(0, 0.25),
                     R = 50L)
summ <- htdv_simstudy_summary(res)
htdv_simstudy_warnings(summ, threshold = 0.7)
```

`htdv_sim_summary` *Aggregated Output of the HTDV Factorial Simulation Study*

Description

Sign-corrected aggregated results of the factorial Monte Carlo study described in Section 12-bis of the companion paper. The study crosses five factors at four levels each (sample size, AR(1) coefficient, innovation tail, imbalance ratio, location shift) for 1024 cells, each replicated 500 times against three inferential layers (HAR, bootstrap, hierarchical Bayesian). One cell (n=500, phi=0.85, t3, imb=1, delta=0) was lost to an I/O collision and is excluded; the table contains 1023 cells x 3 layers = 3069 rows.

Usage

```
htdv_sim_summary
```

Format

A data frame with one row per (cell, layer) and the following columns:

n Per-group primary sample size (40, 80, 200, 500).

phi AR(1) coefficient (0, 0.3, 0.6, 0.85).

tail Innovation distribution: `"normal"`, `"t5"`, `"t3"`, `"t2_1"`.

imb Imbalance ratio n1/n2 (1, 1.5, 3, 6).

delta Location shift in units of sigma_inf (0, 0.1, 0.25, 0.5).

layer Inferential layer: `"har"`, `"boot"`, `"bayes"`.

n_reps Number of replications aggregated (500 for all cells).

reject_rate Empirical rejection rate at nominal alpha = 0.05.

coverage Fraction of replications whose 95% interval covered the true parameter (sign-corrected for HAR and bootstrap).

mean_ci_length Average length of the 95% interval.

mean_runtime Mean per-replication runtime in seconds.

diag_pass_rate Fraction of replications passing all HMC diagnostic gates (Bayesian layer only; `NA` otherwise).

Source

Generated by `run_simstudy.R` of the companion paper repository on master seed `20260422`, with sign-corrected coverage produced by `recover_coverage.R`. Total wall-clock 31.1 hours on a 16-core Linux workstation. See `vignette("HTDV-validation")` for the full report.

`htdv_stack` *Predictive Stacking of Bayesian Models*

Description

Computes stacking weights (Yao, Vehtari, Simpson, Gelman 2018) across a list of fitted models by LOO-log-score maximization.

Usage

```
htdv_stack(fits, method = c("log_score"))
```

Arguments

`fits`	Named list of `htdv_fit` objects.
`method`	Stacking objective. Only `"log_score"` supported in this release.

Value

A list with `weights`, `model_names`, and `elpd_per_model`.

Examples

```
if (requireNamespace("loo", quietly = TRUE)) {
  x <- rnorm(50)
  f1 <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
                 refresh = 0, seed = 1)
  f2 <- htdv_fit(x, model = "wsc", chains = 2, iter = 500,
                 refresh = 0, seed = 2)
  htdv_stack(list(tac = f1, wsc = f2))$weights
}
```

`htdv_summary` *Print-Friendly Summary for an HTDV Fit*

Description

Print-Friendly Summary for an HTDV Fit

Usage

```
htdv_summary(fit, rope = NULL, target = "theta")
```

Arguments

`fit`	An `htdv_fit` object.
`rope`	Optional two-element vector for ROPE decision on the target.
`target`	Parameter name to summarize.

Value

A data frame with posterior mean, sd, 2.5%, 50%, 97.5%, Rhat, bulk and tail ESS for `target`, plus the ROPE decision if requested.

Examples

```
x <- rnorm(50)
fit <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
                refresh = 0, seed = 1)
htdv_summary(fit, target = "theta")
```

`htdv_waic_lfo` *WAIC and Leave-Future-Out Cross-Validation*

Description

Computes the Widely Applicable Information Criterion (Watanabe 2010) and leave-future-out expected log predictive density (Buerkner-Gabry-Vehtari 2020) from a fitted `htdv_fit` object.

Usage

```
htdv_waic_lfo(fit, L = NULL, k_threshold = 0.7)
```

Arguments

`fit`	An `htdv_fit` object with pointwise log-likelihood stored as parameter `log_lik`.
`L`	Block-prefix length; leave-future-out starts at time `L+1`. Default `floor(n/5)`.
`k_threshold`	Pareto-k refresh threshold; when exceeded, importance sampling is refreshed.

Value

A list with `waic`, `elpd_lfo`, `elpd_se`, `pareto_k`, and `refresh_times`.

References

Watanabe, S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion. JMLR 11: 3571-3594. Buerkner, P.-C., Gabry, J., & Vehtari, A. (2020). Approximate Leave-Future-Out Cross-Validation for Bayesian Time Series Models. Journal of Statistical Computation and Simulation 90(14): 2499-2523.

Examples

```
if (requireNamespace("loo", quietly = TRUE)) {
  x <- rnorm(50)
  fit <- htdv_fit(x, model = "tac", chains = 2, iter = 500,
                  refresh = 0, seed = 1)
  htdv_waic_lfo(fit)$waic
}
```

`htdv_whittle` *Whittle Likelihood for Stationary Series*

Description

Computes the Whittle pseudo-log-likelihood at Fourier frequencies for a user-supplied spectral density.

Usage

```
htdv_whittle(x, spec_fun, theta, taper = FALSE)
```

Arguments

`x`	Numeric vector.
`spec_fun`	Function `function(lambda, theta)` returning spectral density values at angular frequencies `lambda`.
`theta`	Numeric parameter vector passed to `spec_fun`.
`taper`	Logical; if `TRUE`, a 10-percent cosine taper is applied.

Value

A list with `loglik` (scalar), `frequencies`, `periodogram`, and `fitted_spectrum`.

References

Whittle, P. (1953). Estimation and Information in Stationary Time Series. Arkiv foer Matematik, 2(5), 423-434. Dzhaparidze, K. (1986). Parameter Estimation and Hypothesis Testing in Spectral Analysis of Stationary Time Series. Springer.

Examples

```
x <- arima.sim(model = list(ar = 0.5), n = 200, rand.gen = rnorm)
ar1_spec <- function(lambda, theta) {
  phi <- theta[1]; sigma2 <- theta[2]
  sigma2 / (2 * pi * (1 - 2 * phi * cos(lambda) + phi^2))
}
htdv_whittle(as.numeric(x), ar1_spec, c(0.5, 1))$loglik
```

Index