Package ‘wARMASVp’

April 22, 2026

Type Package

Title Winsorized ARMA Estimation for Higher-Order Stochastic Volatility Models

Version 0.1.0

Description Estimation, simulation, hypothesis testing, and forecasting for univariate higher-order stochastic volatility SV(p) models. Supports Gaussian, Student-t, and Generalized Error Distribution (GED) innovations, with optional leverage effects. Estimation uses closed-form Winsorized ARMA-SV (W-ARMA-SV) moment-based methods that avoid numerical optimization. Hypothesis testing includes Local Monte Carlo (LMC) and Maximized Monte Carlo (MMC) procedures for leverage effects, heavy tails, and autoregressive order selection. Forecasting is based on Kalman filtering and smoothing. See Ahsan and Dufour (2021) <doi:10.1016/j.jeconom.2020.01.018>, Ahsan, Dufour, and Rodriguez Rondon (2025) for details.

License GPL (>= 3)

URL `https://github.com/roga11/wARMASVp`

BugReports `https://github.com/roga11/wARMASVp/issues`

Encoding UTF-8

Imports Rcpp (>= 1.0.0), gsignal, stats

Suggests pso, GenSA, testthat (>= 3.0.0), knitr, rmarkdown

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation yes

Author Gabriel Rodriguez Rondon [aut, cre] (ORCID: <`https://orcid.org/0009-0005-3769-9921`>), Nazmul Ahsan [aut], Jean-Marie Dufour [aut]

Maintainer Gabriel Rodriguez Rondon <`gabriel.rodriguezrondon@mail.mcgill.ca`>

Repository CRAN

Date/Publication 2026-04-22 08:20:08 UTC

Contents

`filter_svp` *Filter Latent Volatility from an Estimated SV(p) Model*

Description

Applies Kalman filtering (corrected or Gaussian mixture) and RTS smoothing to extract the latent log-volatility process from an estimated SV(p) model.

Usage

```
filter_svp(
  object,
  method = c("corrected", "mixture", "particle"),
  K = 7,
  M = 1000,
  seed = 42,
  del = 1e-10
)
```

Arguments

`object`	An `"svp"`, `"svp_t"`, or `"svp_ged"` object from `svp`.
`method`	Character. Filter method: `"corrected"` (default) for standard Kalman with distribution-specific $\sigma^2_\varepsilon(\nu)$, `"mixture"` for the Gaussian Mixture Kalman Filter (GMKF), or `"particle"` for the Bootstrap Particle Filter (BPF).
`K`	Integer. Number of mixture components for GMKF. Default 7.
`M`	Integer. Number of particles for BPF. Default 1000.
`seed`	Integer. Random seed for BPF. Default 42.
`del`	Numeric. Small constant for log transformation. Default `1e-10`.

Value

An object of class "svp_filter", a list containing:

w_filtered Filtered log-volatility (T-vector).

w_smoothed Smoothed log-volatility (T-vector).

zt Filtered standardized residuals.

zt_smoothed Smoothed standardized residuals.

P_filtered Filtered MSE of first state component.

P_predicted Predicted MSE of first state component.

xi_filtered Full filtered state vectors (p x T matrix).

xi_smoothed Full smoothed state vectors (p x T matrix).

loglik Approximate log-likelihood.

method Filter method used.

model The input model object.

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2)$y
fit <- svp(y, p = 1)
filt <- filter_svp(fit)
plot(filt$w_smoothed, type = "l")
```

---

forecast_svp — *Multi-Step Ahead Volatility Forecast*

Description

Applies Kalman filtering/smoothing to an estimated SV(p) model and produces multi-step ahead volatility forecasts with uncertainty quantification.

Usage

```
forecast_svp(
  object,
  H = 1,
  output = c("log-variance", "variance", "volatility"),
  filter_method = "corrected",
  K = 7,
  M = 1000,
  seed = 42,
  del = 1e-10
)
```

Arguments

`object`	An `"svp"`, `"svp_t"`, or `"svp_ged"` object from `svp`.
`H`	Integer. Maximum forecast horizon. Default 1.
`output`	Character. Primary output scale: `"log-variance"` (default, native log-volatility w_h), `"variance"` (conditional variance $\sigma^2_{T+h\|T}$), or `"volatility"` (conditional std dev $\sigma_{T+h\|T}$). All three are always computed and stored; this controls which is used by `print` and `plot` methods.
`filter_method`	Character. Filter method: `"corrected"` (default), `"mixture"` (GMKF), or `"particle"` (BPF).
`K`	Integer. Number of mixture components for GMKF. Default 7.
`M`	Integer. Number of particles for BPF. Default 1000.
`seed`	Integer. Random seed for BPF. Default 42.
`del`	Numeric. Small constant for log transformation. Default `1e-10`.

Value

An object of class `"svp_forecast"`, a list containing:

w_forecasted Primary forecast (scale determined by `output`).

log_var_forecast Log-volatility forecasts $w_{T+h|T}$.

var_forecast Conditional variance forecasts $\sigma^2_{T+h|T}$.

vol_forecast Conditional volatility forecasts $\sigma_{T+h|T}$.

P_forecast Forecast MSE $P_{T+h|T}$ for each horizon.

w_estimated Filtered log-volatility.

w_smoothed Smoothed log-volatility.

zt Filtered standardized residuals.

zt_smoothed Smoothed standardized residuals.

ys Demeaned log-squared returns.

mdl The estimated model object.

H The forecast horizon.

output The chosen output scale.

filter_output The `"svp_filter"` object from filtering.

Examples

```
sim <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2,
               leverage = TRUE, rho = -0.3)
fit <- svp(sim$y, p = 1, leverage = TRUE)
fc <- forecast_svp(fit, H = 10)
plot(fc)
```

lmc_ar *LMC Test for AR Order in SV(p) Models*

Description

Performs a Local Monte Carlo (LMC) test of the null hypothesis $H_0 : \phi_{p_0+1} = \cdots = \phi_p = 0$ (i.e., that an SV(p_0) model is sufficient against an SV(p) alternative).

Usage

```
lmc_ar(
  y,
  p_null,
  p_alt,
  J = 10,
  N = 99,
  burnin = 500,
  del = 1e-10,
  wDecay = FALSE,
  Bartlett = FALSE,
  Amat = NULL,
  sigvMethod = "factored"
)
```

Arguments

`y`	Numeric vector. Observed returns.
`p_null`	Integer. Order under the null hypothesis.
`p_alt`	Integer. Order under the alternative (`p_alt` > `p_null`).
`J`	Integer. Winsorizing parameter. Default 10.
`N`	Integer. Number of Monte Carlo replications. Default 99.
`burnin`	Integer. Burn-in for simulation. Default 500.
`del`	Numeric. Small constant for log transformation. Default `1e-10`.
`wDecay`	Logical. Use decaying weights. Default `FALSE`.
`Bartlett`	Logical. If `TRUE`, use Bartlett kernel HAC weighting matrix for a GMM-LRT-type test statistic. If `FALSE` (default), use the sum of squared extra AR coefficients.
`Amat`	Weighting matrix specification. `NULL` (default) for identity weighting, or `"Weighted"` for data-driven HAC. Takes precedence over `Bartlett`. User-supplied matrices are not supported for AR order tests.
`sigvMethod`	Character. Method for σ_v estimation: `"factored"` (default), `"hybrid"`, or `"direct"`.

Details

When `Bartlett = FALSE` (default), the test statistic is $T \sum_{j=p_0+1}^{p} \hat{\phi}_j^2$, i.e., the sum of squared extra AR coefficients scaled by sample size.

When `Bartlett = TRUE`, the test statistic is based on the GMM-LRT approach with a Bartlett kernel HAC weighting matrix: $S = T \times (M_{H_0} - M_{H_1})$, where M denotes the GMM criterion evaluated at the null and alternative estimates. Simulations yielding negative test statistics are discarded and re-drawn.

Value

An object of class `"svp_test"`, a list containing:

s0 Test statistic from observed data.

sN Simulated null distribution (vector of length N).

pval Monte Carlo p-value.

test_type Character string identifying the test.

null_param Name of the parameter(s) tested.

null_value Value(s) under the null hypothesis.

call The matched call.

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2)$y
test <- lmc_ar(y, p_null = 1, p_alt = 2, J = 10, N = 49)
print(test)
```

`lmc_ged` *LMC Test for GED Shape Parameter*

Description

Performs a Local Monte Carlo (LMC) test of the null hypothesis $H_0 : \nu = \nu_0$ for the shape parameter in an SV(p) model with GED errors. Testing $\nu_0 = 2$ corresponds to testing normality.

Usage

```
lmc_ged(
  y,
  p = 1,
  J = 10,
  N = 99,
  nu_null,
  burnin = 500,
```

```
  del = 1e-10,
  wDecay = FALSE,
  Bartlett = FALSE,
  Amat = NULL,
  direction = c("two-sided", "less", "greater"),
  sigvMethod = "factored",
  winsorize_eps = 0
)
```

Arguments

y	Numeric vector. Observed returns.
p	Integer. AR order of the volatility process. Default 1.
J	Integer. Winsorizing parameter. Default 10.
N	Integer. Number of Monte Carlo replications. Default 99.
nu_null	Numeric. Value of ν under the null hypothesis.
burnin	Integer. Burn-in for simulation. Default 500.
del	Numeric. Small constant for log transformation. Default 1e-10.
wDecay	Logical. Use decaying weights. Default FALSE.
Bartlett	Logical. Use Bartlett kernel HAC for weighting matrix. Default FALSE.
Amat	Weighting matrix specification. NULL (default) for identity weighting, "Weighted" for data-driven HAC, or a (p+3)x(p+3) matrix. Takes precedence over Bartlett.
direction	Character. Test direction: "two-sided" (default), "less" (H1: nu < nu_null), or "greater" (H1: nu > nu_null). Uses signed root of the LR statistic for one-sided tests.
sigvMethod	Character. Method for σ_v estimation: "factored" (default), "hybrid", or "direct".
winsorize_eps	Numeric. Winsorization threshold for moment conditions. Default 0 (no winsorization).

Value

An object of class "svp_test".

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2, errorType = "GED", nu = 1.5)$y
test <- lmc_ged(y, p = 1, J = 10, N = 49, nu_null = 2)
print(test)
```

lmc_lev *LMC Test for Leverage in SV(p) Models*

Description

Performs a Local Monte Carlo (LMC) test of the null hypothesis $H_0 : \rho = \rho_0$ (typically $\rho_0 = 0$, i.e., no leverage) using a GMM likelihood-ratio type statistic.

Usage

```
lmc_lev(
  y,
  p = 1,
  J = 10,
  N = 99,
  rho_null = 0,
  burnin = 500,
  rho_type = "pearson",
  del = 1e-10,
  trunc_lev = TRUE,
  wDecay = FALSE,
  Bartlett = FALSE,
  Amat = NULL,
  errorType = "Gaussian",
  logNu = FALSE,
  sigvMethod = "factored",
  winsorize_eps = 0
)
```

Arguments

`y`	Numeric vector. Observed returns.
`p`	Integer. Order of the volatility process. Default 1.
`J`	Integer. Winsorizing parameter. Default 10.
`N`	Integer. Number of Monte Carlo replications. Default 99.
`rho_null`	Numeric. Value of ρ under the null. Default 0.
`burnin`	Integer. Burn-in for simulation. Default 500.
`rho_type`	Character. Correlation type. Default `"pearson"`.
`del`	Numeric. Small constant for log transformation. Default `1e-10`.
`trunc_lev`	Logical. Truncate leverage correlation estimate to `[-0.999, 0.999]`. Default `TRUE`.
`wDecay`	Logical. Use decaying weights. Default `FALSE`.
`Bartlett`	Logical. If `TRUE`, use Bartlett kernel HAC weighting matrix. If `FALSE`, use identity matrix. Default `FALSE`.

Amat	Weighting matrix specification. NULL (default) for identity weighting, "Weighted" for data-driven HAC, or a numeric matrix of dimension (p+3)x(p+3) (Gaussian) or (p+4)x(p+4) (heavy-tail). Takes precedence over Bartlett.
errorType	Character. Error distribution: "Gaussian" (default), "Student-t", or "GED".
logNu	Logical. Use log-space for nu estimation (Student-t only). Default FALSE.
sigvMethod	Method for sigma_v estimation: "factored" (default), "direct", or "hybrid".
winsorize_eps	Number of extreme autocovariance lags to winsorize (0 = none). Default 0.

Value

An object of class "svp_test", a list containing:

s0 Test statistic from observed data.

sN Simulated null distribution (vector of length N).

pval Monte Carlo p-value.

test_type Character string identifying the test.

null_param Name of the parameter tested.

null_value Value under the null hypothesis.

call The matched call.

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2, leverage = TRUE, rho = -0.3)$y
test <- lmc_lev(y, p = 1, J = 10, N = 99)
print(test)
```

---

lmc_t *LMC Test for Student-t Tail Parameter*

Description

Performs a Local Monte Carlo (LMC) test of the null hypothesis $H_0 : \nu = \nu_0$ for the degrees of freedom parameter in an SV(p) model with Student-t errors. Testing $\nu_0 = \infty$ (or a large value) corresponds to testing for normality.

Usage

```
lmc_t(
  y,
  p = 1,
  J = 10,
  N = 99,
  nu_null,
```

```
  burnin = 500,
  del = 1e-10,
  wDecay = FALSE,
  Bartlett = FALSE,
  Amat = NULL,
  logNu = TRUE,
  direction = c("two-sided", "less", "greater"),
  sigvMethod = "factored",
  winsorize_eps = 0
)
```

Arguments

`y`	Numeric vector. Observed returns.
`p`	Integer. AR order of the volatility process. Default 1.
`J`	Integer. Winsorizing parameter. Default 10.
`N`	Integer. Number of Monte Carlo replications. Default 99.
`nu_null`	Numeric. Value of ν under the null hypothesis.
`burnin`	Integer. Burn-in for simulation. Default 500.
`del`	Numeric. Small constant for log transformation. Default `1e-10`.
`wDecay`	Logical. Use decaying weights. Default `FALSE`.
`Bartlett`	Logical. Use Bartlett kernel HAC for weighting matrix. Default `FALSE`.
`Amat`	Weighting matrix specification. `NULL` (default) for identity weighting, `"Weighted"` for data-driven HAC, or a `(p+3)x(p+3)` matrix. Takes precedence over `Bartlett`.
`logNu`	Logical. Use log-space for nu estimation. Default `TRUE`.
`direction`	Character. Test direction: `"two-sided"` (default), `"less"` (H1: nu < nu_null), or `"greater"` (H1: nu > nu_null). Uses signed root of the LR statistic for one-sided tests.
`sigvMethod`	Character. Method for σ_v estimation: `"factored"` (default), `"hybrid"`, or `"direct"`.
`winsorize_eps`	Numeric. Winsorization threshold for moment conditions. Default 0 (no winsorization).

Value

An object of class `"svp_test"`.

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2, errorType = "Student-t", nu = 5)$y
test <- lmc_t(y, p = 1, J = 10, N = 49, nu_null = 5)
print(test)
```

mmc_ar *MMC Test for AR Order in SV(p) Models*

Description

Performs a Maximized Monte Carlo (MMC) test of $H_0 : \phi_{p_0+1} = \cdots = \phi_p = 0$ by maximizing the MC p-value over nuisance parameters $(\phi_1, \ldots, \phi_{p_0}, \sigma_y, \sigma_v)$.

Usage

```
mmc_ar(
  y,
  p_null,
  p_alt,
  J = 10,
  N = 99,
  burnin = 500,
  eps = NULL,
  threshold = 1,
  method = "pso",
  maxit = NULL,
  del = 1e-10,
  wDecay = FALSE,
  Bartlett = FALSE,
  Amat = NULL,
  sigvMethod = "factored"
)
```

Arguments

y	Numeric vector. Observed returns.
p_null	Integer. Order under the null hypothesis.
p_alt	Integer. Order under the alternative (p_alt > p_null).
J	Integer. Winsorizing parameter. Default 10.
N	Integer. Number of Monte Carlo replications. Default 99.
burnin	Integer. Burn-in for simulation. Default 500.
eps	Numeric vector. Half-width of search region around estimated nuisance parameters. Default rep(0.3, p_null+2).
threshold	Numeric. Target p-value. Default 1.
method	Character. Optimization method: "pso" or "GenSA". Default "pso".
maxit	Integer. Maximum iterations/evaluations. Default depends on method.
del	Numeric. Small constant for log transformation. Default 1e-10.
wDecay	Logical. Use decaying weights. Default FALSE.

Bartlett	Logical. If TRUE, use Bartlett kernel HAC weighting matrix for a GMM-LRT-type test statistic. If FALSE (default), use the sum of squared extra AR coefficients.
Amat	Weighting matrix specification. NULL (default) for identity weighting, or "Weighted" for data-driven HAC. Takes precedence over Bartlett. User-supplied matrices are not supported for AR order tests.
sigvMethod	Character. Method for σ_v estimation: "factored" (default), "hybrid", or "direct".

Value

A list with optimization output including value (maximized p-value) and par (nuisance parameters at the maximum).

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2)$y
mmc <- mmc_ar(y, p_null = 1, p_alt = 2, J = 10, N = 19,
              method = "pso", maxit = 10)
mmc$value
```

---

mmc_ged *MMC Test for GED Shape Parameter*

Description

Performs a Maximized Monte Carlo (MMC) test of $H_0 : \nu = \nu_0$ for the GED shape parameter.

Usage

```
mmc_ged(
  y,
  p = 1,
  J = 10,
  N = 99,
  nu_null,
  burnin = 500,
  eps = NULL,
  threshold = 1,
  method = "pso",
  maxit = NULL,
  del = 1e-10,
  wDecay = FALSE,
  Bartlett = FALSE,
  Amat = NULL,
```

```
  direction = c("two-sided", "less", "greater"),
  sigvMethod = "factored",
  winsorize_eps = 0
)
```

Arguments

- `y` Numeric vector. Observed returns.
- `p` Integer. AR order of the volatility process. Default 1.
- `J` Integer. Winsorizing parameter. Default 10.
- `N` Integer. Number of Monte Carlo replications. Default 99.
- `nu_null` Numeric. Value of ν under the null hypothesis.
- `burnin` Integer. Burn-in for simulation. Default 500.
- `eps` Numeric vector. Half-width of search region around estimated nuisance parameters. Must have length `p+2` (one entry per nuisance parameter: $\phi_1, \ldots, \phi_p, \sigma_y, \sigma_v$). Default `rep(0.3, p+2)`.
- `threshold` Numeric. Target p-value. Default 1.
- `method` Character. Optimization method: `"pso"` or `"GenSA"`. Default `"pso"`.
- `maxit` Integer. Maximum iterations/evaluations. Default depends on method.
- `del` Numeric. Small constant for log transformation. Default `1e-10`.
- `wDecay` Logical. Use decaying weights. Default `FALSE`.
- `Bartlett` Logical. Use Bartlett kernel HAC for weighting matrix. Default `FALSE`.
- `Amat` Weighting matrix specification. `NULL` (default) for identity weighting, `"Weighted"` for data-driven HAC, or a `(p+3)x(p+3)` matrix. Takes precedence over `Bartlett`.
- `direction` Character. Test direction: `"two-sided"` (default), `"less"` (H1: nu < nu_null), or `"greater"` (H1: nu > nu_null). Uses signed root of the LR statistic for one-sided tests.
- `sigvMethod` Character. Method for σ_v estimation: `"factored"` (default), `"hybrid"`, or `"direct"`.
- `winsorize_eps` Numeric. Winsorization threshold for moment conditions. Default 0 (no winsorization).

Value

A list with optimization output including `value` (maximized p-value) and `par` (nuisance parameters at the maximum).

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2, errorType = "GED", nu = 1.5)$y
mmc <- mmc_ged(y, p = 1, J = 10, N = 19, nu_null = 2, method = "pso", maxit = 10)
mmc$value
```

mmc_lev	*MMC Test for Leverage in SV(p) Models*

Description

Performs a Maximized Monte Carlo (MMC) test of the null hypothesis $H_0 : \rho = \rho_0$ by maximizing the MC p-value over nuisance parameters (phi, sigma_y, sigma_v).

Usage

```
mmc_lev(
  y,
  p = 1,
  J = 10,
  N = 99,
  rho_null = 0,
  burnin = 500,
  eps = NULL,
  threshold = 1,
  method = "pso",
  maxit = NULL,
  rho_type = "pearson",
  del = 1e-10,
  trunc_lev = TRUE,
  wDecay = FALSE,
  Bartlett = FALSE,
  Amat = NULL,
  errorType = "Gaussian",
  logNu = FALSE,
  sigvMethod = "factored",
  winsorize_eps = 0
)
```

Arguments

`y`	Numeric vector. Observed returns.
`p`	Integer. Order of the volatility process. Default 1.
`J`	Integer. Winsorizing parameter. Default 10.
`N`	Integer. Number of Monte Carlo replications. Default 99.
`rho_null`	Numeric. Value of ρ under the null. Default 0.
`burnin`	Integer. Burn-in for simulation. Default 500.
`eps`	Numeric vector. Half-width of the search region around the estimated nuisance parameters. For Gaussian: length `p+2` (phi, sigma_y, sigma_v). For Student-t/GED: length `p+2` (phi, sigma_y, sigma_v; nu bounds set proportionally at +/-30 length `p+3` (phi, sigma_y, sigma_v, nu). Default `NULL` which uses `rep(0.3, p+2)` with proportional nu bounds.

threshold	Numeric. Target p-value (optimization stops if reached). Default 1.
method	Character. Optimization method: "pso" (particle swarm), "GenSA" (generalized simulated annealing). Default "pso".
maxit	Integer or list. Maximum iterations/evaluations for the optimizer. Default depends on method.
rho_type	Character. Correlation type. Default "pearson".
del	Numeric. Small constant for log transformation. Default 1e-10.
trunc_lev	Logical. Truncate leverage correlation estimate to [-0.999, 0.999]. Default TRUE.
wDecay	Logical. Use decaying weights. Default FALSE.
Bartlett	Logical. If TRUE, use Bartlett kernel HAC weighting matrix. If FALSE, use identity matrix. Default FALSE.
Amat	Weighting matrix specification. NULL (default) for identity weighting, "Weighted" for data-driven HAC, or a numeric matrix of dimension (p+3)x(p+3) (Gaussian) or (p+4)x(p+4) (heavy-tail). Takes precedence over Bartlett.
errorType	Character. Error distribution: "Gaussian" (default), "Student-t", or "GED".
logNu	Logical. Use log-space for nu estimation (Student-t only). Default FALSE.
sigvMethod	Method for sigma_v estimation: "factored" (default), "direct", or "hybrid".
winsorize_eps	Number of extreme autocovariance lags to winsorize (0 = none). Default 0.

Value

A list with the optimization output including:

value Maximized p-value.

par Nuisance parameter values at the maximum.

Additional fields depend on the optimization method used.

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2, leverage = TRUE, rho = -0.3)$y
mmc <- mmc_lev(y, p = 1, J = 10, N = 19, method = "pso", maxit = 10)
mmc$value
```

mmc_t	*MMC Test for Student-t Tail Parameter*

Description

Performs a Maximized Monte Carlo (MMC) test of $H_0 : \nu = \nu_0$ by maximizing the MC p-value over nuisance parameters (phi, sigma_y, sigma_v).

Usage

```
mmc_t(
  y,
  p = 1,
  J = 10,
  N = 99,
  nu_null,
  burnin = 500,
  eps = NULL,
  threshold = 1,
  method = "pso",
  maxit = NULL,
  del = 1e-10,
  wDecay = FALSE,
  Bartlett = FALSE,
  Amat = NULL,
  logNu = TRUE,
  direction = c("two-sided", "less", "greater"),
  sigvMethod = "factored",
  winsorize_eps = 0
)
```

Arguments

`y`	Numeric vector. Observed returns.
`p`	Integer. AR order of the volatility process. Default 1.
`J`	Integer. Winsorizing parameter. Default 10.
`N`	Integer. Number of Monte Carlo replications. Default 99.
`nu_null`	Numeric. Value of ν under the null hypothesis.
`burnin`	Integer. Burn-in for simulation. Default 500.
`eps`	Numeric vector. Half-width of search region around estimated nuisance parameters. Must have length p+2 (one entry per nuisance parameter: $\phi_1, \ldots, \phi_p, \sigma_y, \sigma_v$). Default `rep(0.3, p+2)`.
`threshold`	Numeric. Target p-value. Default 1.
`method`	Character. Optimization method: `"pso"` or `"GenSA"`. Default `"pso"`.

maxit	Integer. Maximum iterations/evaluations. Default depends on method.
del	Numeric. Small constant for log transformation. Default 1e-10.
wDecay	Logical. Use decaying weights. Default FALSE.
Bartlett	Logical. Use Bartlett kernel HAC for weighting matrix. Default FALSE.
Amat	Weighting matrix specification. NULL (default) for identity weighting, "Weighted" for data-driven HAC, or a (p+3)x(p+3) matrix. Takes precedence over Bartlett.
logNu	Logical. Use log-space for nu estimation. Default TRUE.
direction	Character. Test direction: "two-sided" (default), "less" (H1: nu < nu_null), or "greater" (H1: nu > nu_null). Uses signed root of the LR statistic for one-sided tests.
sigvMethod	Character. Method for σ_v estimation: "factored" (default), "hybrid", or "direct".
winsorize_eps	Numeric. Winsorization threshold for moment conditions. Default 0 (no winsorization).

Value

A list with optimization output including value (maximized p-value) and par (nuisance parameters at the maximum).

Examples

```
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2, errorType = "Student-t", nu = 5)$y
mmc <- mmc_t(y, p = 1, J = 10, N = 19, nu_null = 5, method = "pso", maxit = 10)
mmc$value
```

---

sim_svp — *Simulate from a Stochastic Volatility Model*

Description

Master simulation function for SV(p) models. Supports Gaussian, Student-t, and GED error distributions, with optional leverage effects. This mirrors the interface of svp for estimation.

Usage

```
sim_svp(
  n,
  phi,
  sigy,
  sigv,
  errorType = "Gaussian",
  leverage = FALSE,
```

```
  rho = 0,
  nu = NULL,
  burnin = 500
)
```

Arguments

`n`	Integer. Length of the simulated series.
`phi`	Numeric vector. AR coefficients for log-volatility (length p).
`sigy`	Numeric. Unconditional standard deviation of returns.
`sigv`	Numeric. Standard deviation of volatility innovations.
`errorType`	Character. Error distribution: `"Gaussian"` (default), `"Student-t"`, or `"GED"`.
`leverage`	Logical. If `TRUE`, simulate with leverage effects (correlated return and volatility shocks). Default is `FALSE`.
`rho`	Numeric. Leverage parameter (correlation between return and volatility shocks). Must be in `[-1, 1]`. Only used when `leverage = TRUE`. Default is 0.
`nu`	Numeric. Shape parameter for heavy-tailed distributions. Degrees of freedom for Student-t (must be > 2) or GED shape (must be > 0). Required when `errorType` is `"Student-t"` or `"GED"`.
`burnin`	Integer. Number of initial observations to discard. Default 500.

Details

The model is:

$$y_t = \sigma_y \exp(w_t/2) z_t$$

$$w_t = \phi_1 w_{t-1} + \cdots + \phi_p w_{t-p} + \sigma_v v_t$$

where z_t follows a distribution specified by `errorType` (Gaussian, Student-t, or GED), and v_t is i.i.d. standard normal. When `leverage = TRUE`, the correlation between z_t and v_{t+1} is ρ.

For Student-t errors with leverage, the scale-mixture representation $z_t = \zeta_t \lambda_t^{-1/2}$ is used, where leverage operates through the Gaussian component ζ_t. For GED errors with leverage, a Gaussian copula construction $z_t = F_{\mathrm{GED}}^{-1}(\Phi(\zeta_t))$ is used. In both cases the returned `z` is the *effective* return innovation (not the latent ζ_t), with marginal distribution matching the `errorType`.

Value

A named list of four length-`n` numeric vectors:

- `y` Observed returns y_t.
- `h` Log-volatility process w_t (equivalently h_t).
- `z` Return innovation such that $y_t = \sigma_y \exp(h_t/2)\, z_t$. Marginal distribution matches `errorType`: N(0,1) for Gaussian, t(ν) for Student-t, unit-variance GED(ν) for GED.
- `v` Volatility innovation such that $h_t - \sum_{j=1}^{p} \phi_j h_{t-j} = \sigma_v\, v_t$. Always N(0,1); under leverage, $v_t = \rho\, \zeta_{t-1} + \sqrt{1-\rho^2}\, \epsilon_t$.

See Also

svp for estimation.

Examples

```
# Gaussian SV(1), no leverage
sim <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2)
plot(sim$y, type = "l")

# Gaussian SV(1) with leverage
sim_lev <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2,
                   leverage = TRUE, rho = -0.5)
plot(sim_lev$y, type = "l")

# Student-t SV(1)
sim_t <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2,
                 errorType = "Student-t", nu = 5)
plot(sim_t$y, type = "l")

# GED SV(1)
sim_ged <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2,
                   errorType = "GED", nu = 1.5)
plot(sim_ged$y, type = "l")
```

---

svp *Estimate a Stochastic Volatility Model*

Description

Master estimation function for SV(p) models using the Winsorized ARMA-SV (W-ARMA-SV) method. Supports Gaussian, Student-t, and GED error distributions, with optional leverage effects.

Usage

```
svp(
  y,
  p = 1,
  J = 10,
  leverage = FALSE,
  errorType = "Gaussian",
  rho_type = "pearson",
  del = 1e-10,
  trunc_lev = TRUE,
  wDecay = FALSE,
  logNu = FALSE,
  sigvMethod = "factored",
  winsorize_eps = 0
)
```

Arguments

- `y` — Numeric vector. Observed returns (e.g., de-meaned log returns).
- `p` — Integer. Order of the volatility process. Default is 1.
- `J` — Integer. Winsorizing parameter controlling the number of autocovariance blocks used. Default is 10.
- `leverage` — Logical. If `TRUE`, estimate leverage parameter ρ. Default is `FALSE`.
- `errorType` — Character. Error distribution: `"Gaussian"` (default), `"Student-t"`, or `"GED"`.
- `rho_type` — Character. Correlation type for leverage estimation. One of `"pearson"` (default), `"kendall"`, or `"both"`.
- `del` — Numeric. Small constant for log transformation: $\log(y_t^2 + \delta)$. Default is `1e-10`.
- `trunc_lev` — Logical. If `TRUE`, truncate the estimated leverage parameter to `[-0.999, 0.999]`. Default is `TRUE`. Setting to `FALSE` can reduce bias in some cases but may yield estimates outside the parameter space.
- `wDecay` — Logical. Use linearly decaying weights in the WLS estimation. Default is `FALSE`.
- `logNu` — Logical. Solve for ν in log-space for numerical stability (Student-t only). Default is `FALSE`.
- `sigvMethod` — Character. Method for estimating σ_v. One of: `"factored"` (default) — factored-variance estimator (recommended; dominates RMSE in most settings, see ADRR 2025); `"direct"` — direct variance decomposition; `"hybrid"` — AD2021 closed-form for $p = 1$, falls back to `"direct"` for $p \geq 2$ (Student-t and GED only).
- `winsorize_eps` — Integer. Number of extreme autocovariance lags to winsorize (`0` = none). Used in Student-t and GED σ_ε^2 estimation. Default `0`.

Details

The model is:

$$y_t = \sigma_y \exp(w_t/2) z_t$$

$$w_t = \phi_1 w_{t-1} + \cdots + \phi_p w_{t-p} + \sigma_v v_t$$

where z_t follows a distribution specified by `errorType` (Gaussian, Student-t, or GED), and v_t is i.i.d. standard normal. When `leverage = TRUE`, the correlation between z_t and v_t is estimated as ρ.

For Student-t errors with leverage, the correction factor $C_t(\nu)$ from the scale-mixture representation is applied. For GED errors with leverage, the exact implicit equation is solved via 1D root-finding with Gauss-Hermite quadrature.

Value

Depending on `errorType`:

- `"Gaussian"`: An object of class `"svp"` (see below).
- `"Student-t"`: An object of class `"svp_t"`.
- `"GED"`: An object of class `"svp_ged"`.

The `"svp"` class contains:

mu Mean of $\log(y_t^2 + \delta)$.

phi Numeric vector of AR coefficients.

sigv Standard deviation of volatility innovations.

sigy Unconditional standard deviation.

rho Leverage parameter (if estimated, otherwise NA).

y The original data.

p, J Model order and winsorizing parameter.

errorType The error distribution used.

call The matched call.

References

Ahsan, N. and Dufour, J.-M. (2021). Simple estimators and inference for higher-order stochastic volatility models. *Journal of Econometrics*, 224(1), 181-197.

Ahsan, N., Dufour, J.-M., and Rodriguez Rondon, G. (2025). Estimation and inference for stochastic volatility models with heavy-tailed distributions.

See Also

svpSE for standard errors.

Examples

```
# Gaussian SV(1) without leverage (default)
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2)$y
fit <- svp(y)
summary(fit)

# With leverage
y2 <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2, leverage = TRUE, rho = -0.3)$y
fit2 <- svp(y2, leverage = TRUE)
coef(fit2)

# Student-t errors
y3 <- sim_svp(1000, phi = 0.9, sigy = 1, sigv = 0.2, errorType = "Student-t", nu = 5)$y
fit3 <- svp(y3, errorType = "Student-t")
summary(fit3)
```

svpSE *Simulation-Based Standard Errors for SV(p) Models*

Description

Computes standard errors and confidence intervals for estimated parameters by simulating from the fitted model and re-estimating. Supports all model types returned by `svp`: Gaussian (with or without leverage), Student-t, and GED.

Usage

```
svpSE(object, n_sim = 199, alpha = 0.05, burnin = 500, logNu = FALSE)
```

Arguments

`object`	A fitted model object from `svp`. Can be of class `"svp"`, `"svp_t"`, or `"svp_ged"`.
`n_sim`	Integer. Number of Monte Carlo replications. Default 199.
`alpha`	Numeric. Significance level for confidence intervals. Default 0.05.
`burnin`	Integer. Burn-in period for simulation. Default 500.
`logNu`	Logical. Solve for ν in log-space for numerical stability (Student-t only). Default is `FALSE`.

Value

A list with:

CI 2 x k matrix of confidence intervals (lower, upper).

SEsim0 Standard errors relative to true parameter values.

SEsim Standard errors relative to sample mean.

ISEconservative Conservative interval-based standard errors.

ISEliberal Liberal interval-based standard errors.

thetamat Matrix of parameter estimates from simulations.

Examples

```
# Gaussian SV(1)
y <- sim_svp(1000, phi = 0.95, sigy = 1, sigv = 0.2)$y
fit <- svp(y)
se <- svpSE(fit, n_sim = 49)
se$CI
```

Index