Package ‘RSDC’

August 4, 2026

Title Regime-Switching Dynamic Correlation Models

Version 1.7-0

Description Estimation, forecasting, simulation, and portfolio construction for regime-switching models with exogenous variables as in Pelletier (2006) <doi:10.1016/j.jeconom.2005.01.013>.

License GPL-3

Encoding UTF-8

Language en-US

URL https://github.com/ArdiaD/RSDC

BugReports https://github.com/ArdiaD/RSDC/issues

Imports Rdpack (>= 2.0), DEoptim, mvtnorm, numDeriv, generics, graphics, parallel, stats, utils, Rcpp

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown, testthat (>= 3.1.7), quadprog, Rsolnp, ggplot2

Config/testthat/edition 3

RdMacros Rdpack

Depends R (>= 4.0)

LazyData true

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author David Ardia [aut, cre, cph, fnd] (ORCID: <https://orcid.org/0000-0003-2823-782X>), Benjamin Seguin [aut], Roosevelt Ymele Nguemo [aut] (ORCID: <https://orcid.org/0009-0003-9399-6256>)

Maintainer David Ardia <david.ardia.ch@gmail.com>

Repository CRAN

Date/Publication 2026-08-04 09:30:08 UTC

Contents

`augment.rsdc_fit` *Augment data with fitted RSDC regime information*

Description

broom-style augmentation: returns the estimation residuals together with the smoothed regime probabilities and the most likely (Viterbi) regime path.

Usage

```
## S3 method for class 'rsdc_fit'
augment(x, ...)
```

Arguments

`x`	An `"rsdc_fit"` object (must carry its estimation residuals).
`...`	Unused; for generic compatibility.

Value

A data frame with one row per observation: the residual columns (`.resid1`, ...), smoothed regime probabilities (`.smoothed_p1`, ...) and the Viterbi state `.state`.

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
head(generics::augment(fit))
```

---

autoplot.rsdc_fit *Plot a fitted RSDC model with* **ggplot2**

Description

Plots the smoothed regime probabilities over the sample as stacked areas. Requires the suggested **ggplot2** package.

Usage

```
## S3 method for class 'rsdc_fit'
autoplot(object, ...)
```

Arguments

object	An "rsdc_fit" object.
...	Unused; for generic compatibility.

Value

A ggplot object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  # Two persistent regimes: low (0.1) vs high (0.8) correlation
  sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                       beta = matrix(qlogis(0.9), 2, 1),
                       mu = matrix(0, 2, 2),
                       sigma = array(c(1, 0.1, 0.1, 1,
                                       1, 0.8, 0.8, 1), c(2, 2, 2)),
                       N = 2, seed = 2)
  fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
  ggplot2::autoplot(fit)
}
```

ff5ind *Five Fama-French industry portfolios with MCCC and VIX (daily)*

Description

Daily value-weighted returns of five Fama-French industry portfolios (Manufacturing, Energy, High Tech, Health, Utilities) together with the Media Climate Change Concerns (MCCC) index and the CBOE Volatility Index (VIX), aligned on U.S. trading days from 2005-01-03 to 2025-06-30 (the sample end is set by the 2025 MCCC release). This is the empirical data set of the K = 5 illustration: filter each return series with a univariate GARCH-type model, standardize the residuals, and use `mccc` and/or `log(vix)` (standardized over the estimation window) as TVTP covariates.

Usage

```
data(ff5ind)
```

Format

A data frame with 5155 rows and 8 columns:

DATE Trading day, from 2005-01-03 to 2025-06-30.

Manuf, Enrgy, HiTec, Hlth, Utils Daily value-weighted returns of the corresponding Fama-French 10-industry portfolios, in percent.

mccc Media Climate Change Concerns (Aggregate) index. The index is published monthly; the value of the month containing `DATE` is forward-filled across its trading days (the same convention as `mccc`). Raw (unstandardized).

vix CBOE Volatility Index, daily close. Raw level; the illustration uses `scale(log(vix))`.

Source

Industry portfolios: Kenneth R. French Data Library, *10 Industry Portfolios (Daily)*, `https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`; industry definitions from Fama and French (1997); data (c) Eugene F. Fama and Kenneth R. French. MCCC: monthly *Aggregate* series of the Media Climate Change Concerns workbook (2025 release), `https://sentometrics-research.com/download/mccc/`, Ardia et al. (2023). VIX: Chicago Board Options Exchange, via FRED (series `VIXCLS`), `https://fred.stlouisfed.org/series/VIXCLS`.

References

Fama EF, French KR (1997). "Industry costs of equity." *Journal of Financial Economics*, **43**(2), 153–193. doi:10.1016/S0304405X(96)008963.

Ardia D, Bluteau K, Boudt K, Inghelbrecht K (2023). "Climate Change Concerns and the Performance of Green vs. Brown Stocks." *Management Science*, **69**(12), 7607–7632. doi:10.1287/mnsc.2022.4636.

Examples

```
data("ff5ind")
str(ff5ind)

# TVTP covariate matrices for the illustration window
X_mccc <- cbind(intercept = 1, MCCC   = as.numeric(scale(ff5ind$mccc)))
X_vix  <- cbind(intercept = 1, logVIX = as.numeric(scale(log(ff5ind$vix))))
```

---

glance.rsdc_fit *Glance at a fitted RSDC model*

Description

broom-style one-row model summary.

Usage

```
## S3 method for class 'rsdc_fit'
glance(x, ...)
```

Arguments

x	An "rsdc_fit" object.
...	Unused; for generic compatibility.

Value

A one-row data frame with method, n_regimes, logLik, AIC, BIC, df and nobs.

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
generics::glance(fit)
```

greenbrown — *Green vs Brown portfolio dataset*

Description

Daily returns for a green and a brown portfolios constructed following the equal-weighted 10-90 percentile approach.

Usage

```
data(greenbrown)
```

Format

A data frame with 2266 rows and three columns:

DATE Dates ranging from 2014-01-02 to 2022-12-30.

return_green Numeric returns for the green portfolio.

return_brown Numeric returns for the brown portfolio.

Source

Built from the source workbook in `data-raw/green-brown-ptf.xlsx` (not shipped on CRAN); see `data-raw/greenbrown.R`.

Examples

```
data("greenbrown")
str(greenbrown)
head(greenbrown)
```

mccc — *Media Climate Change Concerns (MCCC) index*

Description

Daily Media Climate Change Concerns (MCCC) index used as the exogenous covariate driving the time-varying transition probabilities (TVTP) in the empirical illustration. Only the headline *Aggregate* series is retained. The index is published at the monthly frequency; here it is forward-filled across the calendar days of each month and aligned to the `greenbrown` trading calendar, so that `mccc` is row-for-row mergeable with `greenbrown` on `DATE`.

Usage

```
data(mccc)
```

Format

A data frame with 2266 rows and two columns:

DATE Dates ranging from 2014-01-02 to 2022-12-30, matching `greenbrown`.

mccc Numeric Media Climate Change Concerns (Aggregate) index value for the month containing `DATE`. Provided in raw (unstandardized) form; standardize over your estimation window before use as a covariate.

Source

Monthly *Aggregate* series in `inst/extdata/mccc-monthly.csv`, extracted from the Media Climate Change Concerns workbook of (Ardia et al. 2023).

References

Ardia D, Bluteau K, Boudt K, Inghelbrecht K (2023). "Climate Change Concerns and the Performance of Green vs. Brown Stocks." *Management Science*, **69**(12), 7607–7632. doi:10.1287/mnsc.2022.4636.

Examples

```
data("mccc")
str(mccc)
head(mccc)

# Row-aligned with greenbrown; build a TVTP covariate matrix (intercept + MCCC)
data("greenbrown")
stopifnot(identical(mccc$DATE, greenbrown$DATE))
X <- cbind(intercept = 1, MCCC = as.numeric(scale(mccc$mccc)))
```

`RSDC` *RSDC: Regime-Switching Dynamic Correlation Models*

Description

The **RSDC** package provides a comprehensive framework for modeling, estimating, and forecasting correlation structures in multivariate time series under regime-switching dynamics. It supports both fixed transition probabilities and *time-varying transition probabilities* (TVTP) driven by exogenous variables.

The methodology is particularly suited to empirical asset pricing and portfolio management applications, enabling users to incorporate macroeconomic, financial, or climate-related predictors into the regime dynamics. The package integrates the full workflow — from model estimation to covariance matrix reconstruction and portfolio optimization — in a single, reproducible pipeline.

Main Features

- **Model estimation and filtering:** `rsdc_hamilton` (Hamilton filter), `rsdc_likelihood` (likelihood computation), `rsdc_estimate` (parameter estimation).
- **Correlation and covariance forecasting:** `rsdc_forecast` (in-sample / 70-30 path), `rsdc_forecast_ahead` (multi-step-ahead), `rsdc_corr_bands` (parameter-uncertainty bands), `rsdc_viterbi` (most likely regime path).
- **Portfolio construction:** `rsdc_minvar` (minimum-variance portfolios), `rsdc_maxdiv` (maximum-diversification portfolios).
- **Simulation:** `rsdc_simulate` (simulate TVTP regime-switching series).
- **Fitted-model methods:** `rsdc_estimate()` returns an object of class `"rsdc_fit"` with `print`, `summary`, `coef`, `logLik` (so `AIC`/`BIC` work), `nobs`, `vcov`, `confint`, `predict`, and `simulate` methods.

Authors

David Ardia, Benjamin Seguin and Roosevelt Ymele Nguemo. David Ardia is the maintainer (also copyright holder and funder).

References

Engle RF (2002). "Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models." *Journal of Business & Economic Statistics*, **20**(3), 339–350. doi:10.1198/073500102288618487.

Hamilton JD (1989). "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle." *Econometrica*, **57**(2), 357–384. doi:10.2307/1912559.

Pelletier D (2006). "Regime switching for dynamic correlations." *Journal of Econometrics*, **131**(1-2), 445–473. doi:10.1016/j.jeconom.2005.01.013.

Examples

```
# Quickstart: simulate two correlation regimes, fit, and inspect the fit
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
summary(fit)
c(AIC = AIC(fit), BIC = BIC(fit))
# Most likely regime path
table(rsdc_viterbi(fit))
# Getting-started vignette:  browseVignettes("RSDC")
# The companion article (methodology + a five-industry out-of-sample study)
# is reproducible from the packaged data; see citation("RSDC").
```

rsdc_bootstrap — *Parametric bootstrap standard errors for a fitted RSDC model*

Description

Simulates B data sets from the fitted data-generating process, re-estimates each one — *warm-started* at the original MLE so the global search is skipped — and returns the bootstrap covariance, standard errors and percentile intervals of the parameters. This complements the analytic (Hessian/OPG/sandwich) covariances from vcov.rsdc_fit and, being simulation-based, respects the parameter bounds ($\rho \in (-1, 1)$, transition probabilities in $[0, 1]$).

Usage

```
rsdc_bootstrap(object, B = 199, X = NULL, seed = NULL, level = 0.95, cores = 1)
```

Arguments

object	An object of class "rsdc_fit".
B	Integer number of bootstrap replications (default 199).
X	Covariate matrix for "tvtp" (defaults to the stored estimation covariates).
seed	Optional integer seed for reproducibility.
level	Confidence level for the percentile intervals (default 0.95).
cores	Integer (default 1). Number of cores used to re-estimate the replicates in parallel (via the **parallel** package). All B data sets are simulated first, from a single RNG stream, so the result is identical for any number of cores.

Details

For method = "tvtp" the data are simulated with rsdc_simulate from the fitted beta and covariate matrix X; for "noX" from the fitted (fixed) transition matrix; for "const" as i.i.d. draws from the single regime. Each replicate is re-estimated with the same method/N and control = list(start = coef(object), compute_se = FALSE), which warm-starts the local optimizer at the original estimates (fast, deterministic given a seed). Replicates whose re-estimation fails are dropped.

Value

A list with replicates ($B' \times$ npar matrix of successful re-estimates), vcov, se, ci (percentile intervals) and the effective number of replicates B.

See Also

vcov.rsdc_fit, rsdc_simulate, rsdc_estimate

Examples

```
set.seed(1)
X <- cbind(1, as.numeric(scale(seq_len(300))))
y <- scale(matrix(rnorm(300 * 2), 300, 2))
fit <- rsdc_estimate("tvtp", residuals = y, N = 2, X = X)
bs <- rsdc_bootstrap(fit, B = 50, seed = 1)
bs$se
bs$ci
```

---

rsdc_corr_bands *Uncertainty bands for the predicted correlation path*

Description

Propagates parameter uncertainty into the (smoothed-probability-weighted) predicted correlation path of a fitted "rsdc_fit" model. Parameter vectors are drawn from the asymptotic sampling distribution $\mathcal{N}(\hat{\theta}, \widehat{\mathrm{Var}}(\hat{\theta}))$ (using the covariance returned by vcov.rsdc_fit); for each draw the Hamilton filter/smoother is rerun and the predicted correlation path recomputed. The pointwise quantiles of these paths form the band.

Usage

```
rsdc_corr_bands(
  object,
  residuals = NULL,
  X = NULL,
  B = 500L,
  level = 0.95,
  seed = NULL,
  cores = 1
)
```

Arguments

object	An "rsdc_fit" object (from rsdc_estimate) estimated with control = list(compute_se = TRUE) so a covariance is available.
residuals	Optional $T \times K$ residual matrix; defaults to the matrix stored on the fit.
X	Optional $T \times p$ covariate matrix ("tvtp" only); defaults to the matrix stored on the fit.
B	Integer. Number of parameter draws (default 500).
level	Confidence level for the pointwise band (default 0.95).
seed	Optional integer seed for reproducibility.
cores	Integer (default 1). Number of cores used to rerun the filter over the draws in parallel (via the **parallel** package). All draws are generated up front, so the result is identical for any number of cores.

Details

This reflects estimation (parameter) uncertainty in the correlation parameters and transition dynamics; it does not add the irreducible regime-classification uncertainty already summarized by the smoothed probabilities.

Draws are taken from the *unconstrained* Gaussian approximation; draws that yield an invalid model (e.g. $|\rho| \geq 1$, a non-positive-definite correlation matrix, or an invalid transition matrix) are dropped with a warning and the band is computed from the remaining draws. When the estimate sits close to a parameter bound this truncates the sampling distribution, so the reported band can be somewhat too narrow there; the attribute `B_used` reports how many draws survived.

Value

A list with one element per correlation pair ($C = K(K-1)/2$ of them), each a $T \times 3$ matrix with columns `fit`, `lower`, `upper`; plus attributes `level` and `B_used` (draws that yielded a valid path).

See Also

`rsdc_forecast`, `rsdc_bootstrap`

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
bands <- rsdc_corr_bands(fit, B = 100, seed = 1)
head(bands[[1]])
```

`rsdc_estimate` *Estimate Regime-Switching or Constant Correlation Model (Wrapper)*

Description

Unified front-end that dispatches to one of three estimators:

- `f_optim()` — TVTP specification (`method = "tvtp"`).
- `f_optim_noX()` — fixed transition matrix (`method = "noX"`).
- `f_optim_const()` — constant correlation, single regime (`method = "const"`).

Usage

```
rsdc_estimate(
  method = c("tvtp", "noX", "const"),
  residuals,
  N = 2,
  X = NULL,
  out_of_sample = FALSE,
  control = list()
)
```

Arguments

`method`	Character. One of `"tvtp"`, `"noX"`, `"const"`.
`residuals`	Numeric matrix $T \times K$. Typically standardized residuals/returns.
`N`	Integer. Number of regimes. Ignored when `method = "const"`.
`X`	Numeric matrix $T \times p$ of exogenous covariates (required for `"tvtp"`).
`out_of_sample`	Logical. If `TRUE`, a fixed 70/30 split is applied prior to estimation.
`control`	Optional named list of estimation settings, forwarded to the backends and optimizers. Any subset of the following elements may be supplied (defaults in parentheses):

`start` (`NULL`) Warm start; skips the **DEoptim** global search. Two forms are accepted. A *numeric vector* in the objective's packing order (see `rsdc_likelihood`) refines a single point locally — e.g. re-examine an earlier optimum with `control = list(start = coef(fit0))`. An `rsdc_starts` *object* runs a data-driven multi-start instead: each start is refined locally, the highest-likelihood fit is kept, and the fit carries `start_logliks` and `start_pars`. Since 1.7-0 the default global search is feasible at any K (partial-correlation parameterization); `rsdc_starts` remains a fast deterministic complement — when both routes agree, the optimum is cross-certified: `st <- rsdc_starts(y, N = 2, method = "noX")` then `control = list(start = st, cores = 4)`. A warm start disables `n_starts`.

`n_starts` (1) Number of independent global+local searches from seeds `seed`, `seed + 1, ...`, keeping the highest-likelihood fit. Because **DEoptim** is a *stochastic* global search, a single run can land in a suboptimal basin of a multimodal likelihood; independent searches make that failure both unlikely and *visible*: with `n_starts >= 2` the fit carries `start_logliks`, and a warning is issued when the best value is not replicated by at least two searches. `control = list(n_starts = 4, cores = 4)` is a good default for multimodal surfaces (e.g. `"noX"` with $N \geq 3$) at the wall-clock cost of a single search.

`cores` (1) Parallel workers, via the base **parallel** package (forked on Unix, PSOCK cluster on Windows). With either multi-start form, the starts are fitted in parallel — `control = list(n_starts = 4, cores = 4)` — and the result is identical for any value, because each fit is deterministic given its start or seed; only the wall time changes.

seed (123) RNG seed of the stochastic **DEoptim** search: control = list(seed = 42). Unused when a warm start skips the global search.

compute_se (TRUE) Compute observed-information standard errors at the optimum. Disable for speed inside loops (bootstraps, searches): control = list(compute_se = FALSE).

itermax (500), NP (10 x **npar**), steptol (50) Budget of the **DEoptim** global search: maximum generations, population size, early stop after steptol stagnant generations. E.g. a quick exploratory fit: control = list(itermax = 100, NP = 200).

maxit (1000) Maximum iterations of the local L-BFGS-B refinement (optim).

do_trace (FALSE) Print optimizer progress.

Details

- **Method selection:** match.arg() validates method.
- **Inputs:** "tvtp" requires non-NULL X; N is ignored for "const".
- **Split:** If out_of_sample = TRUE, the first 70% is used for fitting.

Value

An object of class "rsdc_fit": a list with components

transition_matrix Estimated transition matrix (1×1 for "const"; for "tvtp", evaluated at the in-sample covariate means).

correlations Regime lower-triangular correlations (rows ordered by ascending mean).

covariances Array $K \times K \times N$ of regime correlation matrices.

log_likelihood In-sample log-likelihood; full-sample when out_of_sample = FALSE.

log_likelihood_oos OOS log-likelihood on held-out 30 percent, or NULL.

beta TVTP coefficients (only for "tvtp"; NULL otherwise).

par, coefficients Named vector of estimated parameters in objective order.

vcov Observed-information variance-covariance at the MLE, or NULL if the Hessian is singular/unavailable (e.g. at a bound).

se Named standard errors (sqrt(diag(vcov))), or NULL.

convergence optim convergence code (0 = converged).

npar, nobs Number of free parameters and observations used in estimation.

method, N, K, p, call Fit metadata.

The returned object supports the standard fitted-model generics — print, summary, coef, logLik (so AIC/BIC work), nobs, vcov, confint, predict, simulate, and plot — documented together in Methods for fitted RSDC models.

References

Mullen K, Ardia D, Gil D, Windover D, Ulrich J (2011). "DEoptim: An R Package for Global Optimization by Differential Evolution." *Journal of Statistical Software*, **40**(6), 1–26. doi:10.18637/jss.v040.i06.

Hamilton JD (1989). "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle." *Econometrica*, **57**(2), 357–384. doi:10.2307/1912559.

Pelletier D (2006). "Regime switching for dynamic correlations." *Journal of Econometrics*, **131**(1-2), 445–473. doi:10.1016/j.jeconom.2005.01.013.

See Also

Methods for fitted RSDC models, `rsdc_hamilton` and `rsdc_likelihood`.

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
print(fit)
summary(fit)
c(AIC = AIC(fit), BIC = BIC(fit))
X <- cbind(1, scale(seq_len(nrow(sim$observations))))
fit_tvtp <- rsdc_estimate("tvtp", residuals = sim$observations, N = 2, X = X)
coef(fit_tvtp)
```

`rsdc_fit-methods` *Methods for fitted RSDC models*

Description

A model fitted with `rsdc_estimate` is returned as an object of class `"rsdc_fit"`, which supports the standard fitted-model generics — so it can be used like any other fitted model in R (`lm`, `glm`, ...): print and summarize it, extract coefficients, standard errors and confidence intervals, compute `AIC`/`BIC`, forecast, simulate new data from the fitted process, and plot the regime probabilities. Each method is described in the *Functions* section below.

Usage

```
## S3 method for class 'rsdc_fit'
print(x, digits = 4, ...)

## S3 method for class 'rsdc_fit'
coef(object, ...)

## S3 method for class 'rsdc_fit'
nobs(object, ...)

## S3 method for class 'rsdc_fit'
logLik(object, ...)

## S3 method for class 'rsdc_fit'
vcov(object, type = c("hessian", "opg", "sandwich", "bootstrap"), ...)

## S3 method for class 'rsdc_fit'
confint(
  object,
  parm,
  level = 0.95,
  type = c("hessian", "opg", "sandwich", "bootstrap"),
  ...
)

## S3 method for class 'rsdc_fit'
summary(object, ...)

## S3 method for class 'rsdc_fit'
predict(
  object,
  residuals,
  sigma_matrix,
  value_cols,
  X = NULL,
  out_of_sample = FALSE,
  ...
)

## S3 method for class 'rsdc_fit'
simulate(object, nsim = 1, seed = NULL, X = NULL, n = NULL, ...)

## S3 method for class 'rsdc_fit'
plot(x, which = c("smoothed", "filtered"), ...)
```

Arguments

x, object	An object of class "rsdc_fit", as returned by rsdc_estimate.

`digits`	Number of significant digits for printing.
`...`	Further arguments passed to methods (`vcov` forwards `B`, `X`, `seed`, and `cores` to `rsdc_bootstrap` when `type = "bootstrap"`; otherwise unused).
`type`	Covariance estimator: one of `"hessian"`, `"opg"`, `"sandwich"` (numerical) or `"bootstrap"` (simulation-based).
`parm`	Vector of parameter names/indices (default: all).
`level`	Confidence level.
`residuals`	Numeric matrix $T \times K$ of standardized residuals on which the forecast filter is run.
`sigma_matrix`	Numeric matrix of conditional standard deviations.
`value_cols`	Columns of `sigma_matrix` giving the asset order.
`X`	Covariate matrix $T \times p$ (required for a `"tvtp"` fit; `simulate` uses its row count as the simulation length).
`out_of_sample`	Logical; passed to `rsdc_forecast` (70/30 split).
`nsim`	Unused (kept for generic compatibility; one path is returned).
`seed`	Optional RNG seed.
`n`	Series length for `"noX"`/`"const"` (defaults to the fitted T).
`which`	Either `"smoothed"` (default) or `"filtered"`.

Value

`print` and `plot` return `x` invisibly. `coef`, `nobs`, `logLik`, `vcov` and `confint` return the usual objects of those generics. `summary` returns a `"summary.rsdc_fit"` list with a coefficient table (estimate, SE, z, p) and regime diagnostics. `predict` returns the result of `rsdc_forecast`. `simulate` returns a list with `states` and `observations`.

Functions

- `print(rsdc_fit)`: Compact printer for a fitted model.
- `coef(rsdc_fit)`: Coefficient vector (named, in objective order).
- `nobs(rsdc_fit)`: Number of observations used in estimation.
- `logLik(rsdc_fit)`: Log-likelihood (carries `df` and `nobs` so `stats::AIC/BIC` work out of the box).
- `vcov(rsdc_fit)`: Variance-covariance matrix of the estimates. The *numerical* estimators are `type = "hessian"` (default, inverse observed information), `"opg"` (outer product of gradients) and `"sandwich"` (QML/robust $H^{-1}(\sum_t s_t s_t')H^{-1}$); the *simulation-based* estimator is `"bootstrap"`, which calls `rsdc_bootstrap` (pass `B`, `X`, `seed`, and `cores` through `...`). The bootstrap is recomputed on each call.
- `confint(rsdc_fit)`: Wald confidence intervals from the chosen covariance (numerical or `"bootstrap"`). For bootstrap *percentile* intervals instead of Wald, use `rsdc_bootstrap` directly.
- `summary(rsdc_fit)`: Summary with a coefficient table (estimate, SE, z, p).

- `predict(rsdc_fit)`: Forecast from a fitted model (wraps `rsdc_forecast`); supply the residuals, conditional volatilities, and (for `"tvtp"`) the covariate matrix `X`.
- `simulate(rsdc_fit)`: Simulate from a fitted model. For `"tvtp"` supply a covariate matrix `X` (its row count sets the length); for `"noX"` the fixed transition matrix is used; for `"const"` a single regime is drawn.
- `plot(rsdc_fit)`: Plot the smoothed regime probabilities (one panel per regime) for a fitted `"noX"`/`"tvtp"` model. `which = "filtered"` plots filtered probabilities instead.

See Also

`rsdc_estimate` for fitting; `rsdc_forecast`, `rsdc_bootstrap`.

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
summary(fit)
confint(fit)
c(AIC = AIC(fit), BIC = BIC(fit))
plot(fit)
```

`rsdc_forecast` *Forecast Covariance/Correlation Paths from an RSDC Model*

Description

Generates per-period correlation and covariance matrices from a fitted model: `"const"` (constant correlation), `"noX"` (fixed transition matrix), or `"tvtp"` (time-varying transition probabilities).

Usage

```
rsdc_forecast(
  method = c("tvtp", "noX", "const"),
  N,
  residuals,
  X = NULL,
  final_params,
  sigma_matrix,
  value_cols,
  out_of_sample = FALSE,
  control = list()
)
```

Arguments

method	Character. One of "tvtp", "noX", "const".
N	Integer. Number of regimes (ignored for "const").
residuals	Numeric matrix $T \times K$ used to compute correlations or run the filter.
X	Optional numeric matrix $T \times p$ (required for "tvtp").
final_params	List with fitted parameters (e.g., from rsdc_estimate): must include correlations, and either transition_matrix ("noX") or beta ("tvtp"); include log_likelihood for BIC computation.
sigma_matrix	Numeric matrix $T \times K$ of forecasted standard deviations.
value_cols	Character/integer vector of columns in sigma_matrix that define asset order.
out_of_sample	Logical. If TRUE, use a fixed 70/30 split; otherwise use the whole sample.
control	Optional list; supports threshold (in (0,1), default 0.7).

Details

- **Forecast horizon:** If out_of_sample = TRUE, filter on the first threshold fraction and forecast on the remainder.
- **Correlation paths:**
  - "const" — constant correlation from final_params$correlations[1,], repeated across time.
  - "noX"/"tvtp" — regime-probability weighted average of regime correlations. Out-of-sample (out_of_sample = TRUE) the weights are the *filtered* probabilities $\Pr(S_t \mid \Omega_t)$, which use no *future* information beyond time t (the OOS window is initialized from the in-sample terminal filtered state). Note these condition on the *contemporaneous* U_t, so they are a real-time *nowcast*, not a one-step-ahead forecast formed before the covariate is seen; for that use rsdc_forecast_ahead. The full-sample case uses *smoothed* probabilities $\Pr(S_t \mid \Omega_T)$. The returned smoothed_probs field always contains the smoothed probabilities.
- **Covariance build:** Reconstruct R_t from the pairwise vector (columns ordered by combn(K, 2)), set $D_t = \mathrm{diag}(\sigma_{t,1}, \ldots, \sigma_{t,K})$, and $\Sigma_t = D_t R_t D_t$.
- **BIC:** Parameter count k is N * ncol(X) + N * K * (K - 1) / 2 for "tvtp" with $N = 2$, N * (N-1) * ncol(X) + N * K * (K - 1) / 2 for "tvtp" with $N \geq 3$, N * (N - 1) + N * K * (K - 1) / 2 for "noX", and K * (K - 1) / 2 for "const".

Value

smoothed_probs $N \times T^*$ smoothed probabilities ("noX"/"tvtp" only).

sigma_matrix $T^* \times K$ slice aligned to the forecast horizon.

cov_matrices List of $K \times K$ covariance matrices $\Sigma_t = D_t R_t D_t$.

predicted_correlations $T^* \times \binom{K}{2}$ pairwise correlations in combn(K, 2) order.

BIC Information criterion $\log(n)\, k - 2\,\ell$. When out_of_sample = TRUE and final_params$log_likelihood_oos is non-NULL, uses the OOS log-likelihood and OOS sample size; otherwise uses the IS values. **Caveat:** with out_of_sample = TRUE this is an out-of-sample predictive score (held-out log-likelihood penalized by the IS parameter count), *not* a conventional in-sample BIC; use it only to compare models fit on the same split, and do not interpret it as a textbook BIC. The backends (rsdc_estimate) hardcode a 70/30 split, so the OOS log-likelihood is only fully consistent with `rsdc_forecast` when the default `threshold = 0.7` is used.

y $T^* \times K$ residual matrix aligned to the forecast horizon.

See Also

rsdc_hamilton, rsdc_estimate, rsdc_minvar, rsdc_maxdiv

Examples

```
set.seed(123)
T <- 60; K <- 3; N <- 2
y <- scale(matrix(rnorm(T*K), T, K))
vols <- matrix(0.2 + 0.05*abs(sin(seq_len(T)/7)), T, K)
rho <- rbind(c(0.10, 0.05, 0.00), c(0.60, 0.40, 0.30))
Pfix <- matrix(c(0.9, 0.1, 0.2, 0.8), 2, 2, byrow = TRUE)
rsdc_forecast("noX", N, y, NULL,
              list(correlations = rho, transition_matrix = Pfix, log_likelihood = -200),
              vols, 1:K)
```

`rsdc_forecast_ahead` *Multi-step-ahead regime and correlation forecasts*

Description

Produces genuine h-step-ahead forecasts of the regime distribution and the implied correlation matrix from the end of the estimation sample. Starting from the terminal filtered regime probabilities ξ_T, the regime distribution is propagated through the Markov chain, $\pi_{T+k} = \pi_{T+k-1} P_{T+k}$, and the forecast correlations are the regime-probability-weighted averages of the regime correlation matrices, $\hat{\rho}_{T+k} = \sum_s \pi_{T+k}(s)\, \rho^{(s)}$.

Usage

```
rsdc_forecast_ahead(
  object,
  horizon = 10L,
  residuals = NULL,
  X = NULL,
  X_future = NULL
)
```

Arguments

`object`	An `"rsdc_fit"` object from rsdc_estimate.
`horizon`	Integer forecast horizon $h \geq 1$ (default `10`).
`residuals`	Optional $T \times K$ residuals used to obtain the terminal filtered state; defaults to the matrix stored on the fit.

X Optional in-sample covariate matrix ("tvtp"); defaults to the fit's stored matrix.

X_future Optional $h \times p$ matrix of future covariates ("tvtp" only). If NULL, the last observed covariate row is reused at every horizon.

Details

For "noX"/"const" the transition matrix is constant, so $\pi_{T+k} = \xi_T P^k$. For "tvtp" each step uses a covariate row: supply future covariates in X_future (an $h \times p$ matrix); if it is NULL the last in-sample covariate row is held constant and a message is emitted.

Value

A list with:

horizon Integer vector 1:h.

regime_probs $h \times N$ matrix of forecast regime probabilities.

predicted_correlations $h \times C$ matrix ($C = K(K-1)/2$) of forecast correlations, one column per asset pair.

See Also

rsdc_forecast for the in-sample / 70-30 path.

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
rsdc_forecast_ahead(fit, horizon = 5)
```

---

rsdc_hamilton *Hamilton Filter (Fixed P or TVTP)*

Description

Runs the Hamilton (1989) filter for a multivariate regime-switching *correlation* model. Supports either a fixed (time-invariant) transition matrix P or time-varying transition probabilities (TVTP) built from exogenous covariates X: a logistic link for $N = 2$ or a softmax link for $N \geq 3$. Returns filtered/smoothed regime probabilities and the log-likelihood.

Usage

```
rsdc_hamilton(
  y,
  X = NULL,
  beta = NULL,
  rho_matrix,
  K,
  N,
  P = NULL,
  xi_init = NULL,
  engine = c("cpp", "r")
)
```

Arguments

`y`	Numeric matrix $T \times K$ of observations (e.g., standardized residuals/returns). Columns are treated as mean-zero with unit variance; only the correlation structure is modeled.
`X`	Optional numeric matrix $T \times p$ of covariates for TVTP. Required if `beta` is supplied.
`beta`	Optional numeric matrix of TVTP coefficients. For $N = 2$: an $N \times p$ matrix; row i gives the logistic coefficient vector so that $p_{ii,t} = \mathrm{logit}^{-1}(X_t^\top \beta_i)$. For $N \geq 3$: an $N \times (N-1)p$ matrix packed row-wise; row i contains $N-1$ consecutive length-p softmax vectors; the N-th logit is fixed at 0 (reference category).
`rho_matrix`	Numeric matrix $N \times C$ of regime correlation parameters, where $C = K(K-1)/2$. Each row is the lower-triangular part (by `lower.tri`) of a regime's correlation matrix.
`K`	Integer. Number of observed series (columns of `y`).
`N`	Integer. Number of regimes.
`P`	Optional $N \times N$ fixed transition matrix. Used only when `X` or `beta` is `NULL`.
`xi_init`	Optional numeric vector of length `N`. If supplied, the forward filter is initialized from this vector (normalized internally to sum to 1) instead of the default uniform $1/N$. Intended for two-pass out-of-sample forecasting: run the filter on the in-sample period first, extract the terminal column of `filtered_probs`, and pass it here to initialize the out-of-sample filter run. Must be non-negative and finite.
`engine`	Character; `"cpp"` (default) runs the filter/smoother in C++ (**RcppArmadillo**), `"r"` the pure-R reference. Both give identical results (used for equivalence testing).

Details

- **Correlation rebuild:** For regime m, a correlation matrix R_m is reconstructed from `rho_matrix[m, ]` (lower-triangular fill + symmetrization). Non-PD proposals are penalized.
- **Transition dynamics:**

  - *Fixed P:* If `X` or `beta` is missing, a constant P is used (user-provided via `P`; otherwise uniform $1/N$ rows).
  - *TVTP:* With `X` and `beta`: for $N = 2$, $p_{ii,t} = \text{logit}^{-1}(X_t^\top \beta_i)$ and $p_{ij,t} = 1 - p_{ii,t}$ $(j \neq i)$; for $N \geq 3$, a softmax over $N - 1$ free logit vectors per row with the N-th logit fixed at 0 (reference category); log-sum-exp stabilized.
- **Numerical safeguards:** A small ridge is added before inversion; if filtering degenerates at a time step, `log_likelihood = -Inf` is returned.

Value

A list with:

filtered_probs $N \times T$ matrix of filtered probabilities $\Pr(S_t = j \mid \Omega_t)$.

smoothed_probs $N \times T$ matrix of smoothed probabilities $\Pr(S_t = j \mid \Omega_T)$.

log_likelihood Scalar log-likelihood of the model given `y`.

Note

For $N = 2$, TVTP uses a logistic link on the diagonal; off-diagonals follow by complement. For $N \geq 3$, a full softmax is used with $(N - 1)$ free logit vectors per row; all entries are independently determined.

X-timing: uses `X[t, ]` (contemporaneous) to form P_t, consistent with `rsdc_simulate`.

References

Hamilton JD (1989). "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle." *Econometrica*, **57**(2), 357–384. doi:10.2307/1912559.

See Also

`rsdc_likelihood` and `rsdc_estimate`.

Examples

```
set.seed(1)
T <- 50; K <- 3; N <- 2
y <- scale(matrix(rnorm(T * K), T, K), center = TRUE, scale = TRUE)

# Example rho: two regimes with different average correlations
rho <- rbind(c(0.10, 0.05, 0.00),
             c(0.60, 0.40, 0.30))  # lower-tri order for K=3

# Fixed-P filtering
Pfix <- matrix(c(0.9, 0.1,
                 0.2, 0.8), nrow = 2, byrow = TRUE)
out_fix <- rsdc_hamilton(y = y, X = NULL, beta = NULL,
                         rho_matrix = rho, K = K, N = N, P = Pfix)
str(out_fix$filtered_probs)

# TVTP filtering (include an intercept yourself)
```

```
X <- cbind(1, scale(seq_len(T)))
beta <- rbind(c(1.2, 0.0),
              c(0.8, -0.1))
out_tvtp <- rsdc_hamilton(y = y, X = X, beta = beta,
                          rho_matrix = rho, K = K, N = N)
out_tvtp$log_likelihood
```

`rsdc_likelihood` *Negative Log-Likelihood for Regime-Switching Correlation Models*

Description

Computes the negative log-likelihood for a multivariate *correlation-only* regime-switching model, with either a fixed (time-invariant) transition matrix or time-varying transition probabilities (TVTP) driven by exogenous covariates. Likelihood evaluation uses the Hamilton (1989) filter.

Usage

```
rsdc_likelihood(params, y, exog = NULL, K, N)
```

Arguments

`params` Numeric vector of model parameters packed as:

- **No exogenous covariates** (`exog = NULL`): first $N(N-1)$ transition parameters (for the fixed transition matrix), followed by $N \times K(K-1)/2$ correlation parameters, stacked *row-wise by regime* in `lower.tri` order.
- **With exogenous covariates** (`exog` provided): first $N \times p$ TVTP coefficients (`beta`) for $N = 2$, or $N \times (N-1)p$ coefficients for $N \geq 3$ (row i holds $N-1$ length-p softmax vectors; last logit = 0 reference), followed by $N \times K(K-1)/2$ correlation parameters, stacked *row-wise by regime* in `lower.tri` order.

`y` Numeric matrix $T \times K$ of observations (e.g., standardized residuals). Columns are treated as mean-zero, unit-variance; only correlation is modeled.

`exog` Optional numeric matrix $T \times p$ of exogenous covariates. If supplied, a TVTP specification is used.

`K` Integer. Number of observed series (columns of `y`).

`N` Integer. Number of regimes.

Details

- **Transition dynamics:**
  - *Fixed P (no `exog`):* `params` begins with transition parameters. For $N = 2$, the implementation maps them to $P = \begin{pmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{pmatrix}$. For $N \geq 3$, row i of P holds $N-1$ free probabilities (packed row-wise in `params`) filling columns $1, \ldots, N-1$; the last column is the row complement $1 - \sum_{j<N} P_{ij}$. Proposals with a negative entry (row sum of free probabilities above 1) are penalized.

  - *TVTP:* with exog, for $N = 2$, $p_{ii,t} = \text{logit}^{-1}(X_t^\top \beta_i)$ and $p_{ij,t} = 1 - p_{ii,t}$ ($j \neq i$); for $N \geq 3$, a softmax over $N - 1$ free logit vectors per row with the N-th logit fixed at 0 (reference category).

- **Correlation build:** per regime, the lower-triangular vector is filled into a symmetric correlation matrix. Non-positive-definite proposals or $|\rho| \geq 1$ are penalized via a large objective value.
- **Evaluation:** delegates to `rsdc_hamilton`; if the filter returns `log_likelihood = -Inf`, a large penalty is returned.

Value

Numeric scalar: the *negative* log-likelihood to be minimized by an optimizer.

Note

The function is written for use inside optimizers; it performs inexpensive validation and returns large penalties for invalid parameterizations instead of stopping with errors.

See Also

`rsdc_hamilton` (filter), `optim`, and `DEoptim`

Examples

```
# Small toy example (N = 2, K = 3), fixed P (no exog)
set.seed(1)
T <- 40; K <- 3; N <- 2
y <- scale(matrix(rnorm(T * K), T, K), center = TRUE, scale = TRUE)

# Pack parameters: trans (p11, p22), then rho by regime (lower-tri order)
p11 <- 0.9; p22 <- 0.8
rho1 <- c(0.10, 0.05, 0.00)  # (2,1), (3,1), (3,2)
rho2 <- c(0.60, 0.40, 0.30)
params <- c(p11, p22, rho1, rho2)

nll <- rsdc_likelihood(params, y = y, exog = NULL, K = K, N = N)
nll

# TVTP example: add X and beta (pack beta row-wise, then rho)
X <- cbind(1, scale(seq_len(T)))
beta <- rbind(c(1.2, 0.0),
              c(0.8, -0.1))
params_tvtp <- c(as.vector(t(beta)), rho1, rho2)
nll_tvtp <- rsdc_likelihood(params_tvtp, y = y, exog = X, K = K, N = N)
nll_tvtp
```

`rsdc_maxdiv` *Maximum-Diversification Portfolio (Rolling Weights)*

Description

Computes rolling maximum-diversification (MaxDiv) portfolio weights from a sequence of per-period covariance matrices implied by forecasted volatilities and correlations. Falls back to equal weights if the nonlinear solver fails.

Usage

```
rsdc_maxdiv(
  sigma_matrix,
  value_cols,
  predicted_corr,
  y,
  long_only = TRUE,
  lag = FALSE
)
```

Arguments

`sigma_matrix`	Numeric matrix $T \times K$ of forecasted standard deviations.
`value_cols`	Character/integer vector naming columns in `sigma_matrix` (asset order).
`predicted_corr`	Numeric matrix $T \times \binom{K}{2}$ of pairwise correlations in `combn(K, 2)` column order.
`y`	Numeric matrix $T \times K$ of asset returns (for realized stats).
`long_only`	Logical. If `TRUE`, impose $w \geq 0$ and $\sum_i w_i = 1$; otherwise bounds are $-1 \leq w_i \leq 1$ with $\sum_i w_i = 1$.
`lag`	Logical. If `FALSE` (default), the period-t weights (built from Σ_t) are applied to the same period's returns `y[t, ]` (in-sample evaluation). If `TRUE`, weights chosen at $t-1$ earn the period-t return (`sum(y[t, ] * weights[t-1, ])`) and the first period's return is `NA`; use `lag = TRUE` for a look-ahead-free out-of-sample backtest.

Details

- **Covariance build:** For each t, reconstruct R_t from the pairwise vector; set $D_t = \mathrm{diag}(\sigma_{t,1}, \ldots, \sigma_{t,K})$ and $\Sigma_t = D_t R_t D_t$.
- **Objective (MaxDiv):** maximize $\mathrm{DR}(w) = \frac{\sum_i w_i \sigma_{t,i}}{\sqrt{w^\top \Sigma_t w}}$ subject to $\sum_i w_i = 1$ and bounds on w. Implemented by minimizing the negative ratio.
- **Solver:** `Rsolnp::solnp` with equality $\sum_i w_i = 1$ and bounds by `long_only`; on error, weights default to $1/K$.

Value

`weights` $T \times K$ matrix of weights.

`returns` Vector of realized portfolio returns `sum(y[t,] * weights[t,])`.

`diversification_ratios` Vector of realized diversification ratios.

`mean_diversification` Average diversification ratio.

`K` Number of assets.

`assets` Asset names.

`volatility` Standard deviation of realized portfolio returns.

See Also

`rsdc_minvar`, `solnp`

Examples

```
# Toy example with K = 3
if (requireNamespace("Rsolnp", quietly = TRUE)) {
  T <- 50; K <- 3
  set.seed(42)
  vols <- matrix(0.2 + 0.05*abs(sin(seq_len(T)/7)), T, K)
  colnames(vols) <- paste0("A", 1:K)
  # simple, stationary correlations (order: (2,1), (3,1), (3,2))
  pred_corr <- cbind(rep(0.20, T), rep(0.10, T), rep(0.05, T))
  rets <- matrix(rnorm(T*K, sd = 0.01), T, K); colnames(rets) <- colnames(vols)

  mx <- rsdc_maxdiv(sigma_matrix   = vols,
                    value_cols     = colnames(vols),
                    predicted_corr = pred_corr,
                    y              = rets,
                    long_only      = TRUE)
  head(mx$weights)
  mx$mean_diversification
}
```

`rsdc_minvar` *Minimum-Variance Portfolio (Rolling Weights)*

Description

Computes rolling minimum-variance (MV) portfolio weights from a sequence of per-period covariance matrices implied by forecasted volatilities and pairwise correlations. Supports long-only or unconstrained MV. If the QP solver fails at a time step, the routine falls back to equal weights.

Usage

```
rsdc_minvar(
  sigma_matrix,
  value_cols,
  predicted_corr,
  y,
  long_only = TRUE,
  lag = FALSE
)
```

Arguments

`sigma_matrix`	Numeric matrix $T \times K$ of forecasted *volatilities* (standard deviations), one column per asset.
`value_cols`	Character or integer vector giving the columns in `sigma_matrix` to use as assets (order defines the asset order).
`predicted_corr`	Numeric matrix $T \times P$ of pairwise correlations, where $P = \binom{K}{2}$ and the columns correspond to `combn(K, 2)` order.
`y`	Numeric matrix $T \times K$ of asset returns aligned with `sigma_matrix`. Used only to compute the realized portfolio volatility.
`long_only`	Logical. If `TRUE` (default), imposes long-only MV with the full-investment constraint $\sum_i w_i = 1$ and $w_i \geq 0$. If `FALSE`, solves unconstrained MV with only $\sum_i w_i = 1$.
`lag`	Logical. If `FALSE` (default), the period-t weights (built from Σ_t) are applied to the same period's returns `y[t, ]` (in-sample evaluation). If `TRUE`, weights chosen at $t-1$ earn the period-t return (`sum(y[t, ] * weights[t-1, ])`) and the first period's return is `NA`; use `lag = TRUE` for a look-ahead-free out-of-sample backtest.

Details

- **Covariance build:** For each t, a correlation matrix R_t is reconstructed from `predicted\_corr[t, ]` (columns in `combn(K, 2)` order) by placing each pairwise correlation in the corresponding off-diagonal entries of a $K \times K$ identity matrix. Let $D_t = \operatorname{diag}(\sigma_{t,1}, \ldots, \sigma_{t,K})$ and $\Sigma_t = D_t R_t D_t$.
- **Optimization:** Minimize $\frac{1}{2} w^\top \Sigma_t w$ subject to $\mathbf{1}^\top w = 1$ and, if `long_only`, $w \geq 0$ (solved with `quadprog::solve.QP`).
- **Failure handling:** If the QP fails at time \(t\), weights default to equal allocation \(w_i = 1/K\).

Value

An object of class `"minvar_portfolio"`:

weights $T \times K$ matrix of MV weights (one row per time).

cov_matrices List of length T with the per-period $K \times K$ covariance matrices.

volatility Realized standard deviation of portfolio returns (same units as y).

y The input y matrix (coerced to $T \times K$).

K Number of assets.

See Also

rsdc_maxdiv (maximum diversification), solve.QP

Examples

```
# Toy example with K = 3 (requires the suggested 'quadprog' package)
if (requireNamespace("quadprog", quietly = TRUE)) {
  T <- 50; K <- 3
  set.seed(42)
  vols <- matrix(0.2 + 0.05*abs(sin(seq_len(T)/7)), T, K)
  colnames(vols) <- paste0("A", 1:K)
  # simple, stationary correlations
 pred_corr <- cbind(rep(0.20, T), rep(0.10, T), rep(0.05, T))  # order: (2,1), (3,1), (3,2)
  rets <- matrix(rnorm(T*K, sd = 0.01), T, K); colnames(rets) <- colnames(vols)

  mv <- rsdc_minvar(sigma_matrix  = vols,
                    value_cols    = colnames(vols),
                    predicted_corr= pred_corr,
                    y             = rets,
                    long_only     = TRUE)
  head(mv$weights)
  mv$volatility
}
```

---

rsdc_simulate *Simulate Multivariate Regime-Switching Data (TVTP)*

Description

Simulates a multivariate time series from a regime-switching model with *time-varying transition probabilities* (TVTP) driven by covariates X. Transition probabilities are generated via a multinomial logistic (softmax) link; observations are drawn from regime-specific Gaussian distributions.

Usage

```
rsdc_simulate(n, X, beta, mu, sigma, N, seed = NULL)
```

Arguments

n	Integer. Number of time steps to simulate; must be at least 2.
X	Numeric matrix $n \times p$ of covariates used to form the transition probabilities. Row X[1,] forms P_1, used only to draw the initial state from $t(P_1)\,\pi_0$; rows X[t,] for $t \geq 2$ form P_t.

beta	Numeric matrix $N \times (N-1) \cdot p$. Transition coefficients packed row-wise by regime, matching the parameterization of rsdc_hamilton and rsdc_estimate. For $N = 2$, each row is a length-p logistic vector: $p_{ii,t} = \mathrm{logit}^{-1}(X_t^\top \beta_i)$. For $N \geq 3$, each row packs $N-1$ softmax vectors of length p; the N-th logit is the reference (zero).
mu	Numeric matrix $N \times K$. Regime-specific mean vectors.
sigma	Numeric array $K \times K \times N$. Regime-specific covariance (here, correlation/variance) matrices; each $K \times K$ slice must be symmetric positive definite.
N	Integer. Number of regimes.
seed	Optional integer. If supplied, sets the RNG seed for reproducibility.

Details

- **Initial state and first draw:** The initial regime S_1 is drawn from $t(P_1)\,\pi_0$ with $\pi_0 = (1/N, \ldots, 1/N)$ and P_1 built from X_1 via the same logistic/softmax link, matching the Hamilton-filter t=1 prior in rsdc_hamilton; the first observation y_1 is drawn from $\mathcal{N}(\mu_{S_1}, \Sigma_{S_1})$.
- **TVTP transition:** For $t \geq 2$ and $N = 2$, $p_{ii,t} = \mathrm{logit}^{-1}(X_t^\top \beta_i)$; for $N \geq 3$, a softmax is used with $N-1$ free logit vectors per row and the N-th logit fixed at zero (reference category). Both use log-sum-exp stabilization.
- **Sampling:** Given S_{t-1}, draw S_t from the categorical distribution with probabilities $P_t(S_{t-1}, \cdot)$ and $y_t \sim \mathcal{N}(\mu_{S_t}, \Sigma_{S_t})$.
- **First-slice NA:** `transition_matrices[,,1]` is always `NA`. At $t = 1$ there is no predecessor state, so no transition matrix is computed. The assignment loop begins at $t = 2$. Users iterating over the returned array should skip the first slice or use `transition_matrices[,,2:n]`.

Value

A list with:

states Integer vector of length `n`; the simulated regime index at each time.

observations Numeric matrix $n \times K$; the simulated observations.

transition_matrices Array $N \times N \times n$; the transition matrix P_t used at each time step (with P_1 undefined by construction; see Details).

Note

Requires **mvtnorm** for multivariate normal sampling (called as `mvtnorm::rmvnorm`).

X-timing: uses `X[t, ]` (contemporaneous) to form P_t, consistent with rsdc_hamilton.

See Also

rsdc_hamilton (filter/evaluation), rsdc_estimate (estimators), rsdc_forecast (forecasting)

Examples

```
set.seed(123)
n <- 200; K <- 3; N <- 2; p <- 2
X <- cbind(1, scale(seq_len(n)))

beta <- matrix(0, nrow = N, ncol = (N - 1) * p)
beta[1, ] <- c(1.2,  0.0)
beta[2, ] <- c(1.0, -0.1)

mu <- rbind(c(0, 0, 0),
            c(0, 0, 0))
rho <- rbind(c(0.10, 0.05, 0.00),
             c(0.60, 0.40, 0.30))
Sig <- array(0, dim = c(K, K, N))
for (m in 1:N) {
  R <- diag(K); R[lower.tri(R)] <- rho[m, ]; R[upper.tri(R)] <- t(R)[upper.tri(R)]
  Sig[, , m] <- R
}
sim <- rsdc_simulate(n = n, X = X, beta = beta, mu = mu, sigma = Sig, N = N, seed = 99)
```

`rsdc_starts` *Data-driven warm starts for high-dimensional estimation*

Description

Builds a set of feasible starting vectors for `rsdc_estimate` from the empirical correlations of low/.../high average-correlation sub-periods. Pass the result via `control = list(start = )` to run a warm-started multi-start search (see *Details*).

Usage

```
rsdc_starts(
  residuals,
  N = 2,
  method = c("tvtp", "noX", "const"),
  X = NULL,
  window = 126,
  n_starts = 5,
  shrink = NULL,
  stay = 0.95
)

## S3 method for class 'rsdc_starts'
print(x, ...)
```

Arguments

residuals	Numeric matrix $T \times K$ of standardized residuals, as passed to rsdc_estimate.
N	Integer. Number of regimes (ignored for method = "const").
method	Character. One of "tvtp", "noX", "const".
X	Covariate matrix $T \times p$ (required for "tvtp").
window	Integer. Rolling-correlation window used to classify observations into regimes (default 126, about six months of daily data).
n_starts	Integer. Number of starting vectors (default 5). The shrinkage factors are spaced evenly from 1 to 0.45.
shrink	Optional numeric vector in $(0, 1]$ of shrinkage factors, overriding n_starts (expert setting; one start per factor).
stay	Initial stay probability of each regime (default 0.95).
x	An "rsdc_starts" object.
...	Unused; for generic compatibility.

Details

In moderate dimensions ($K \leq 4$) the default global search (**DEoptim**) works well. In higher dimensions it breaks down: a vector of $K(K-1)/2$ pairwise correlations drawn uniformly in $(-1, 1)$ is almost never a valid (positive-definite) correlation matrix, so the random initial population contains no feasible point. rsdc_starts sidesteps this by initializing each regime at an *empirical* correlation matrix, which is positive semi-definite by construction and positive definite whenever the group has enough non-collinear observations (starts that are not usable are filtered out by the feasibility check below):

1. compute the rolling mean pairwise correlation over window observations;
2. split the sample into N groups by quantiles of that series (calm, ..., turbulent), so regimes are ordered by ascending correlation, matching the package's identification convention;
3. use each group's empirical correlation matrix as that regime's starting values, with persistent initial transitions (stay probability stay) and, for "tvtp", zero covariate slopes (the data reveal the covariate effect during estimation);
4. replicate the start under n_starts shrinkage factors of the correlations towards zero (which preserves positive definiteness), spanning "regimes as contrasted as the split suggests" to "regimes much closer together" — protection against local optima.

Value

An object of class "rsdc_starts": a list with

starts n_starts $\times$ npar matrix; each row is a feasible starting vector in the objective's packing order, with columns named as in coef().

loglik0 Log-likelihood evaluated at each start (finite = feasible).

method, N, K, p The model the starts are built for (validated by rsdc_estimate when the object is supplied).

window, shrink, stay How the starts were constructed.

regime_split Integer vector of length T: the quantile group (1 = lowest correlation) each observation was assigned to.

Methods (by generic)

- `print(rsdc_starts)`: Compact printer: model, dimensions, and feasibility of each start.

See Also

rsdc_estimate (accepts the object via `control$start`), rsdc_likelihood (parameter packing).

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
st <- rsdc_starts(sim$observations, N = 2, method = "noX", n_starts = 3)
print(st)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2,
                     control = list(start = st))
fit$start_logliks   # spread across starts (stability diagnostic)
```

---

rsdc_viterbi — *Most likely regime path (Viterbi decoding)*

Description

Returns the single most likely sequence of regimes $(S_1, \dots, S_T)$ under a fitted `"rsdc_fit"` model, via the Viterbi algorithm. This is the maximum a posteriori *joint* state path and complements the marginal filtered/smoothed regime probabilities from rsdc_forecast: the smoothed probabilities are decoded pointwise, whereas Viterbi enforces a globally coherent transition path.

Usage

```
rsdc_viterbi(object, residuals = NULL, X = NULL)
```

Arguments

`object`	An `"rsdc_fit"` object from rsdc_estimate.
`residuals`	Optional $T \times K$ residual matrix; defaults to the matrix stored on the fit.
`X`	Optional $T \times p$ covariate matrix (`"tvtp"` only); defaults to the matrix stored on the fit.

Value

An integer vector of length T giving the most likely regime at each time, in the model's ascending-correlation regime labeling (1 = lowest-correlation state).

See Also

rsdc_forecast for marginal regime probabilities.

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
# Decoded path recovers the simulated regimes
table(viterbi = rsdc_viterbi(fit), truth = sim$states)
```

---

tidy.rsdc_fit — *Tidy a fitted RSDC model*

Description

broom-style tidier returning a one-row-per-parameter coefficient table.

Usage

```
## S3 method for class 'rsdc_fit'
tidy(x, ...)
```

Arguments

x	An "rsdc_fit" object.
...	Unused; for generic compatibility.

Value

A data frame with columns `term`, `estimate`, `std.error`, `statistic` and `p.value`.

Examples

```
# Two persistent regimes: low (0.1) vs high (0.8) correlation
sim <- rsdc_simulate(n = 500, X = matrix(1, 500, 1),
                     beta = matrix(qlogis(0.9), 2, 1),
                     mu = matrix(0, 2, 2),
                     sigma = array(c(1, 0.1, 0.1, 1,
                                     1, 0.8, 0.8, 1), c(2, 2, 2)),
                     N = 2, seed = 2)
fit <- rsdc_estimate("noX", residuals = sim$observations, N = 2)
generics::tidy(fit)
```

Index