Package 'RSDC'

September 3, 2025

Title Regime-Switching Dynamic Correlation Models

Version 1.1-2

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

RoxygenNote 7.3.2

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

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

Imports Rdpack (>= 2.0), DEoptim, mvtnorm, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0), quadprog, Rsolnp

Config/testthat/edition 3

RdMacros Rdpack

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author David Ardia [aut, cre] (ORCID: <https://orcid.org/0000-0003-2823-782X>), Benjamin Seguin [aut]

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

Repository CRAN

Date/Publication 2025-09-03 08:00:24 UTC

Contents

`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 2024-12-30.

return_green Numeric returns for the green portfolio.

return_brown Numeric returns for the brown portfolio.

Source

Originally data in `inst/extdata/green-brown-ptf.xlsx`.

Examples

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

RSDC *RSDC: Regime-Switching Correlation Models for Portfolio Analysis*

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`.
- **Portfolio construction:** `rsdc_minvar` (minimum-variance portfolios), `rsdc_maxdiv` (maximum-diversification portfolios).
- **Simulation:** `rsdc_simulate` (simulate TVTP regime-switching series).

Authors

David Ardia and Benjamin Seguin

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.2005.01.004.

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 list. Currently forwards `do_trace = FALSE` and `seed = 123` to the backends.

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\

Value

`transition_matrix` Estimated transition matrix (1×1 for `"const"`).

`correlations` Regime lower-triangular correlations.

`covariances` Array of full correlation matrices.

`log_likelihood` Maximized log-likelihood.

`beta` TVTP coefficients (only for `"tvtp"`).

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.2005.01.004.

See Also

`rsdc_hamilton` and `rsdc_likelihood`.

Examples

```
y <- scale(matrix(rnorm(100 * 3), 100, 3))
rsdc_estimate("const", residuals = y)
rsdc_estimate("noX", residuals = y, N = 2)
X <- cbind(1, scale(seq_len(nrow(y))))
rsdc_estimate("tvtp", residuals = y, N = 2, X = X)
```

`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" — empirical correlation of residuals, repeated across time.
  - "noX"/"tvtp" — smoothed-probability weighted average of regime correlations.
- **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", 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 Bayesian Information Criterion $\mathrm{BIC} = \log(n)\, k - 2\,\ell$.

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_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` via a logistic link. Returns filtered/smoothed regime probabilities and the log-likelihood.

Usage

```
rsdc_hamilton(y, X = NULL, beta = NULL, rho_matrix, K, N, P = NULL)
```

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 $N \times p$. TVTP coefficients; row i governs persistence of regime i via `plogis(X[t, ] %*% beta[i, ])`.
`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`.

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`, diagonal entries use `plogis(X[t, ] %*% beta[i, ])`. Off-diagonals are equal and sum to $1 - p_{ii,t}$. For $N = 1$, $P_t = [1]$.
- **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

TVTP uses a logistic link on the diagonal; off-diagonals are equal by construction.

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`, row i corresponds to regime i), 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 $P = \begin{pmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{pmatrix}$.
  - *TVTP:* with `exog`, diagonal persistence is $p_{ii,t} = \mathrm{logit}^{-1}(X_t^\top \beta_i)$; off-diagonals are equal and sum to $1 - p_{ii,t}$.

- **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)
```

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 $\sum_i w_i = 1$.

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)
```

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 $\sum_i w_i = 1$.

Details

- **Covariance build:** For each t, a correlation matrix R_t is reconstructed ... Let $D_t = \mathrm{diag}(\sigma_{t,1}, \ldots, \sigma_{t,K})$ 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 (see Note on units).

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

K Number of assets.

Note on units

The realized portfolio return at time \(t\) is computed as `sum(y[t, ] * weights[t, ]) / 100`. This assumes `y` is expressed in \ remove the `/ 100` in the implementation or convert inputs accordingly.

See Also

`rsdc_maxdiv` (maximum diversification), `solve.QP`

Examples

```
# Toy example with K = 3
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.
`X`	Numeric matrix $n \times p$ of covariates used to form the transition probabilities. Row `X[t, ]` corresponds to covariates available at time `t`. Only rows `1:(n-1)` are used to transition from `t-1` to `t`.
`beta`	Numeric array $N \times N \times p$. Softmax coefficients for the multinomial transition model; `beta[i, j, ]` parameterizes the transition from state i to state j.
`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 sampled uniformly; the first observation y_1 is drawn from $\mathcal{N}(\mu_{S_1}, \Sigma_{S_1})$.
- **TVTP via softmax:** For $t \geq 2$, the row i of P_t is
$$P_t(i,j) = \frac{\exp\big(X_{t-1}^\top \beta_{i,j}\big)}{\sum_{h=1}^{N} \exp\big(X_{t-1}^\top \beta_{i,h}\big)},$$
computed with 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 $y_t \sim \mathcal{N}(\mu_{S_t}, \Sigma_{S_t})$.

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`).

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 <- array(0, dim = c(N, N, p))
beta[1, 1, ] <- c(1.2,  0.0)
beta[2, 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)
```

Index