Package ‘simStateSpace’

November 12, 2023

Title Simulate Data from State Space Models

Version 1.0.0

Description Offers an efficient and user-friendly framework
for simulating data in state space models
where n subjects/units is greater than one
which is common in social and behavioral sciences.

URL https://github.com/jeksterslab/simStateSpace,
https://jeksterslab.github.io/simStateSpace/

BugReports https://github.com/jeksterslab/simStateSpace/issues

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 3.0.0)

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp

Suggests knitr, rmarkdown, testthat, Matrix

RoxygenNote 7.2.3

NeedsCompilation yes

Author Ivan Jacob Agaloos Pesigan [aut, cre, cph]
(<https://orcid.org/0000-0003-4818-8420>)

Maintainer Ivan Jacob Agaloos Pesigan <r.jeksterslab@gmail.com>

R topics documented:

OU2SSM — *Convert Parameters from the Ornstein–Uhlenbeck Model to State Space Model Parameterization*

Description

This function converts parameters from the Ornstein–Uhlenbeck model to state space model parameterization. See details for more information.

Usage

```
OU2SSM(mu, phi, sigma_sqrt, delta_t)
```

Arguments

`mu`	Numeric vector. The long-term mean or equilibrium level ($\boldsymbol{\mu}$).
`phi`	Numeric matrix. The rate of mean reversion, determining how quickly the variable returns to its mean ($\boldsymbol{\Phi}$).
`sigma_sqrt`	Numeric matrix. Cholesky decomposition of the matrix of volatility or randomness in the process ($\boldsymbol{\Sigma}$).
`delta_t`	Numeric. Time interval (δ_t).

Details

The state space parameters as a function of the Ornstein–Uhlenbeck model parameters are given by

$$\boldsymbol{\beta} = \exp\left(-\boldsymbol{\Phi}\Delta_t\right)$$

$$\boldsymbol{\alpha} = -\boldsymbol{\Phi}^{-1}\left(\boldsymbol{\beta} - \mathbf{I}_p\right)$$

$$\mathrm{vec}\left(\boldsymbol{\Psi}\right) = \left\{\left[\left(-\boldsymbol{\Phi} \otimes \mathbf{I}_p\right) + \left(\mathbf{I}_p \otimes -\boldsymbol{\Phi}\right)\right]\left[\exp\left(\left[\left(-\boldsymbol{\Phi} \otimes \mathbf{I}_p\right) + \left(\mathbf{I}_p \otimes -\boldsymbol{\Phi}\right)\right]\Delta_t\right) - \mathbf{I}_{p \times p}\right]\mathrm{vec}\left(\boldsymbol{\Sigma}\right)\right\}$$

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Simulation of State Space Models Data Functions: `Sim2Matrix()`, `SimSSM0Fixed()`, `SimSSM0Vary()`, `SimSSM0()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVAR()`

Examples

```
p <- k <- 2
mu <- c(5.76, 5.18)
phi <- matrix(data = c(0.10, -0.05, -0.05, 0.10), nrow = p)
sigma_sqrt <- chol(
  matrix(data = c(2.79, 0.06, 0.06, 3.27), nrow = p)
)
delta_t <- 0.10

OU2SSM(
  mu = mu,
  phi = phi,
  sigma_sqrt = sigma_sqrt,
  delta_t = delta_t
)
```

`Sim2Matrix` *Simulation Output to Matrix*

Description

This function converts the output of `SimSSM0()`, `SimSSMOU()`, `SimSSMVAR()`, `SimSSM0Fixed()`, `SimSSMOUFixed()`, or `SimSSMVARFixed()` to a matrix.

Usage

```
Sim2Matrix(x, eta = FALSE)
```

Arguments

`x`	R object. Output of `SimSSM0()`, `SimSSMOU()`, `SimSSMVAR()`, `SimSSM0Fixed()`, `SimSSMOUFixed()`, or `SimSSMVARFixed()`.
`eta`	Logical. If `eta = TRUE`, include `eta`. If `eta = FALSE`, exclude `eta`.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), SimSSM0Fixed(), SimSSM0Vary(), SimSSM0(), SimSSMOUFixed(), SimSSMOUVary(), SimSSMOU(), SimSSMVARFixed(), SimSSMVARVary(), SimSSMVAR()

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
I <- diag(k)
I_sqrt <- chol(I)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0_sqrt <- I_sqrt
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi_sqrt <- I_sqrt
nu <- null_vec
lambda <- I
theta_sqrt <- chol(diag(x = 0.50, nrow = k))
time <- 50
burn_in <- 0

# generate data
ssm <- SimSSM0(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  time = time,
  burn_in = burn_in
)

# list to matrix
mat <- Sim2Matrix(ssm)
str(mat)
head(mat)

# generate data
ssm <- SimSSM0Fixed(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
```

```
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  time = time,
  burn_in = burn_in
)

# list to matrix
mat <- Sim2Matrix(ssm)
str(mat)
head(mat)
```

SimSSM0 *Simulate Data from a State Space Model (n = 1)*

Description

This function simulates data from a state space model. See details for more information.

Usage

```
SimSSM0(
  mu0,
  sigma0_sqrt,
  alpha,
  beta,
  psi_sqrt,
  nu,
  lambda,
  theta_sqrt,
  time,
  burn_in
)
```

Arguments

mu0	Numeric vector. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\|0}$).
sigma0_sqrt	Numeric matrix. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\|0}$).
alpha	Numeric vector. Vector of intercepts for the dynamic model ($\boldsymbol{\alpha}$).
beta	Numeric matrix. Transition matrix relating the values of the latent variables at time `t - 1` to those at time `t` ($\boldsymbol{\beta}$).
psi_sqrt	Numeric matrix. Cholesky decomposition of the process noise covariance matrix ($\boldsymbol{\Psi}$).
nu	Numeric vector. Vector of intercepts for the measurement model ($\boldsymbol{\nu}$).

`lambda`	Numeric matrix. Factor loading matrix linking the latent variables to the observed variables ($\boldsymbol{\Lambda}$).
`theta_sqrt`	Numeric matrix. Cholesky decomposition of the measurement error covariance matrix ($\boldsymbol{\Theta}$).
`time`	Positive integer. Number of time points to simulate.
`burn_in`	Positive integer. Number of burn-in points to exclude before returning the results.

Details

The measurement model is given by

$$\mathbf{y}_t = \boldsymbol{\nu} + \boldsymbol{\Lambda} \boldsymbol{\eta}_t + \boldsymbol{\varepsilon}_t \quad \text{with} \quad \boldsymbol{\varepsilon}_t \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Theta}\right)$$

where $\mathbf{y}_t$, $\boldsymbol{\eta}_t$, and $\boldsymbol{\varepsilon}_t$ are random variables and $\boldsymbol{\nu}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\Theta}$ are model parameters. $\mathbf{y}_t$ is a vector of observed random variables at time t, $\boldsymbol{\eta}_t$ is a vector of latent random variables at time t, and $\boldsymbol{\varepsilon}_t$ is a vector of random measurement errors at time t, while $\boldsymbol{\nu}$ is a vector of intercept, $\boldsymbol{\Lambda}$ is a matrix of factor loadings, and $\boldsymbol{\Theta}$ is the covariance matrix of $\boldsymbol{\varepsilon}$.

The dynamic structure is given by

$$\boldsymbol{\eta}_t = \boldsymbol{\alpha} + \boldsymbol{\beta} \boldsymbol{\eta}_{t-1} + \boldsymbol{\zeta}_t \quad \text{with} \quad \boldsymbol{\zeta}_t \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Psi}\right)$$

where $\boldsymbol{\eta}_t$, $\boldsymbol{\eta}_{t-1}$, and $\boldsymbol{\zeta}_t$ are random variables and $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\Psi}$ are model parameters. $\boldsymbol{\eta}_t$ is a vector of latent variables at time t, $\boldsymbol{\eta}_{t-1}$ is a vector of latent variables at time $t-1$, and $\boldsymbol{\zeta}_t$ is a vector of dynamic noise at time t while $\boldsymbol{\alpha}$ is a vector of intercepts, $\boldsymbol{\beta}$ is a matrix of autoregression and cross regression coefficients, and $\boldsymbol{\Psi}$ is the covariance matrix of $\boldsymbol{\zeta}_t$.

Value

Returns a list with the following elements:

- `y`: A `t` by `k` matrix of values for the manifest variables.
- `eta`: A `t` by `p` matrix of values for the latent variables.
- `time`: A vector of discrete time points from 1 to `t`.
- `n`: Number of individuals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: With R examples*. Springer International Publishing. doi:10.1007/9783319524528

See Also

Other Simulation of State Space Models Data Functions: `OU2SSM()`, `Sim2Matrix()`, `SimSSM0Fixed()`, `SimSSM0Vary()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVAR()`

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
I <- diag(k)
I_sqrt <- chol(I)
null_vec <- rep(x = 0, times = k)
mu0 <- null_vec
sigma0_sqrt <- I_sqrt
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi_sqrt <- I_sqrt
nu <- null_vec
lambda <- I
theta_sqrt <- chol(diag(x = 0.50, nrow = k))
time <- 50
burn_in <- 0

# generate data
ssm <- SimSSM0(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

`SimSSM0Fixed` *Simulate Data using a State Space Model Parameterization for n > 1 Individuals (Fixed Parameters)*

Description

This function simulates data using a state space model parameterization for `n > 1` individuals. In this model, the parameters are invariant across individuals.

Usage

```
SimSSM0Fixed(
  n,
  mu0,
```

```
  sigma0_sqrt,
  alpha,
  beta,
  psi_sqrt,
  nu,
  lambda,
  theta_sqrt,
  time,
  burn_in
)
```

Arguments

n	Positive integer. Number of individuals.
mu0	Numeric vector. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\mid 0}$).
sigma0_sqrt	Numeric matrix. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\mid 0}$).
alpha	Numeric vector. Vector of intercepts for the dynamic model ($\boldsymbol{\alpha}$).
beta	Numeric matrix. Transition matrix relating the values of the latent variables at time `t - 1` to those at time `t` ($\boldsymbol{\beta}$).
psi_sqrt	Numeric matrix. Cholesky decomposition of the process noise covariance matrix ($\boldsymbol{\Psi}$).
nu	Numeric vector. Vector of intercepts for the measurement model ($\boldsymbol{\nu}$).
lambda	Numeric matrix. Factor loading matrix linking the latent variables to the observed variables ($\boldsymbol{\Lambda}$).
theta_sqrt	Numeric matrix. Cholesky decomposition of the measurement error covariance matrix ($\boldsymbol{\Theta}$).
time	Positive integer. Number of time points to simulate.
burn_in	Positive integer. Number of burn-in points to exclude before returning the results.

Details

The measurement model is given by

$$\mathbf{y}_{i,t} = \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\eta}_{i,t} + \boldsymbol{\varepsilon}_{i,t} \quad \text{with} \quad \boldsymbol{\varepsilon}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Theta}\right)$$

where $\mathbf{y}_{i,t}$, $\boldsymbol{\eta}_{i,t}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and $\boldsymbol{\nu}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\Theta}$ are model parameters. $\mathbf{y}_{i,t}$ is a vector of observed random variables at time t and individual i, $\boldsymbol{\eta}_{i,t}$ is a vector of latent random variables at time t and individual i, and $\boldsymbol{\varepsilon}_{i,t}$ is a vector of random measurement errors at time t and individual i, while $\boldsymbol{\nu}$ is a vector of intercept, $\boldsymbol{\Lambda}$ is a matrix of factor loadings, and $\boldsymbol{\Theta}$ is the covariance matrix of $\boldsymbol{\varepsilon}$.

The dynamic structure is given by

$$\boldsymbol{\eta}_{i,t} = \boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{\eta}_{i,t-1} + \boldsymbol{\zeta}_{i,t} \quad \text{with} \quad \boldsymbol{\zeta}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Psi}\right)$$

where $\boldsymbol{\eta}_{i,t}$, $\boldsymbol{\eta}_{i,t-1}$, and $\boldsymbol{\zeta}_{i,t}$ are random variables and $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\Psi}$ are model parameters. $\boldsymbol{\eta}_{i,t}$ is a vector of latent variables at time t and individual i, $\boldsymbol{\eta}_{i,t-1}$ is a vector of latent variables at time $t-1$ and individual i, and $\boldsymbol{\zeta}_{i,t}$ is a vector of dynamic noise at time t and individual i while $\boldsymbol{\alpha}$ is a vector of intercepts, $\boldsymbol{\beta}$ is a matrix of autoregression and cross regression coefficients, and $\boldsymbol{\Psi}$ is the covariance matrix of $\boldsymbol{\zeta}_{i,t}$.

Value

Returns a list of length `n`. Each element is a list with the following elements:

- `y`: A `t` by `k` matrix of values for the manifest variables.
- `eta`: A `t` by `p` matrix of values for the latent variables.
- `time`: A vector of discrete time points from 1 to `t`.
- `id`: A vector of ID numbers of length `t`.
- `n`: Number of individuals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: With R examples*. Springer International Publishing. doi:10.1007/9783319524528

See Also

Other Simulation of State Space Models Data Functions: `OU2SSM()`, `Sim2Matrix()`, `SimSSM0Vary()`, `SimSSM0()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVAR()`

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
I <- diag(k)
I_sqrt <- chol(I)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0_sqrt <- I_sqrt
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi_sqrt <- I_sqrt
nu <- null_vec
lambda <- I
theta_sqrt <- chol(diag(x = 0.50, nrow = k))
time <- 50
burn_in <- 0
```

```
# generate data
ssm <- SimSSM0Fixed(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

SimSSM0Vary *Simulate Data using a State Space Model Parameterization for n > 1 Individuals (Varying Parameters)*

Description

This function simulates data using a state space model parameterization for n > 1 individuals. In this model, the parameters can vary across individuals.

Usage

```
SimSSM0Vary(
  n,
  mu0,
  sigma0_sqrt,
  alpha,
  beta,
  psi_sqrt,
  nu,
  lambda,
  theta_sqrt,
  time,
  burn_in
)
```

Arguments

n	Positive integer. Number of individuals.
mu0	List of numeric vectors. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\|0}$).

sigma0_sqrt	List of numeric matrices. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta} \mid 0}$).
alpha	List of numeric vectors. Vector of intercepts for the dynamic model ($\boldsymbol{\alpha}$).
beta	List of numeric matrices. Transition matrix relating the values of the latent variables at time `t - 1` to those at time `t` ($\boldsymbol{\beta}$).
psi_sqrt	List of numeric matrices. Cholesky decomposition of the process noise covariance matrix ($\boldsymbol{\Psi}$).
nu	List of numeric vectors. Vector of intercepts for the measurement model ($\boldsymbol{\nu}$).
lambda	List of numeric matrices. Factor loading matrix linking the latent variables to the observed variables ($\boldsymbol{\Lambda}$).
theta_sqrt	List of numeric matrices. Cholesky decomposition of the measurement error covariance matrix ($\boldsymbol{\Theta}$).
time	Positive integer. Number of time points to simulate.
burn_in	Positive integer. Number of burn-in points to exclude before returning the results.

Details

Parameters can vary across individuals by providing a list of parameter values. If the length of any of the parameters (`mu0`, `sigma0_sqrt`, `alpha`, `beta`, `psi_sqrt`, `nu`, `lambda`, and `theta_sqrt`) is less the `n`, the function will cycle through the available values.

Value

Returns a list of length `n`. Each element is a list with the following elements:

- `y`: A `t` by `k` matrix of values for the manifest variables.
- `eta`: A `t` by `p` matrix of values for the latent variables.
- `time`: A vector of discrete time points from 1 to `t`.
- `id`: A vector of ID numbers of length `t`.
- `n`: Number of individuals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: With R examples*. Springer International Publishing. doi:10.1007/9783319524528

See Also

Other Simulation of State Space Models Data Functions: `OU2SSM()`, `Sim2Matrix()`, `SimSSM0Fixed()`, `SimSSM0()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVAR()`

Examples

```
# prepare parameters
# In this example, beta varies across individuals
set.seed(42)
k <- p <- 3
iden <- diag(k)
iden_sqrt <- chol(iden)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- list(null_vec)
sigma0_sqrt <- list(iden_sqrt)
alpha <- list(null_vec)
beta <- list(
  diag(x = 0.1, nrow = k),
  diag(x = 0.2, nrow = k),
  diag(x = 0.3, nrow = k),
  diag(x = 0.4, nrow = k),
  diag(x = 0.5, nrow = k)
)
psi_sqrt <- list(iden_sqrt)
nu <- list(null_vec)
lambda <- list(iden)
theta_sqrt <- list(chol(diag(x = 0.50, nrow = k)))
time <- 50
burn_in <- 0

ssm <- SimSSM0Vary(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

`SimSSMOU` *Simulate Data from the Ornstein–Uhlenbeck Model using a State Space Model Parameterization (n = 1)*

Description

This function simulates data from the Ornstein–Uhlenbeck model using a state space model parameterization. See details for more information.

Usage

```
SimSSMOU(
  mu0,
  sigma0_sqrt,
  mu,
  phi,
  sigma_sqrt,
  nu,
  lambda,
  theta_sqrt,
  delta_t,
  time,
  burn_in
)
```

Arguments

`mu0`	Numeric vector. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\mid 0}$).
`sigma0_sqrt`	Numeric matrix. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\mid 0}$).
`mu`	Numeric vector. The long-term mean or equilibrium level ($\boldsymbol{\mu}$).
`phi`	Numeric matrix. The rate of mean reversion, determining how quickly the variable returns to its mean ($\boldsymbol{\Phi}$).
`sigma_sqrt`	Numeric matrix. Cholesky decomposition of the matrix of volatility or randomness in the process ($\boldsymbol{\Sigma}$).
`nu`	Numeric vector. Vector of intercepts for the measurement model ($\boldsymbol{\nu}$).
`lambda`	Numeric matrix. Factor loading matrix linking the latent variables to the observed variables ($\boldsymbol{\Lambda}$).
`theta_sqrt`	Numeric matrix. Cholesky decomposition of the measurement error covariance matrix ($\boldsymbol{\Theta}$).
`delta_t`	Numeric. Time interval (δ_t).
`time`	Positive integer. Number of time points to simulate.
`burn_in`	Positive integer. Number of burn-in points to exclude before returning the results.

Details

The measurement model is given by

$$\mathbf{y}_t = \boldsymbol{\nu} + \boldsymbol{\Lambda} \boldsymbol{\eta}_t + \boldsymbol{\varepsilon}_t \quad \mathrm{with} \quad \boldsymbol{\varepsilon}_t \sim \mathcal{N} \left(\mathbf{0}, \boldsymbol{\Theta} \right)$$

where $\mathbf{y}_t$, $\boldsymbol{\eta}_t$, and $\boldsymbol{\varepsilon}_t$ are random variables and $\boldsymbol{\nu}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\Theta}$ are model parameters. $\mathbf{y}_t$ is a vector of observed random variables at time t, $\boldsymbol{\eta}_t$ is a vector of latent random variables at time t, and $\boldsymbol{\varepsilon}_t$ is a vector of random measurement errors at time t, while $\boldsymbol{\nu}$ is a vector of intercept, $\boldsymbol{\Lambda}$ is a matrix of factor loadings, and $\boldsymbol{\Theta}$ is the covariance matrix of $\boldsymbol{\varepsilon}$.

The dynamic structure is given by

$$\mathrm{d}\boldsymbol{\eta}_t = \boldsymbol{\Phi}\left(\boldsymbol{\mu} - \boldsymbol{\eta}_t\right)\mathrm{d}t + \boldsymbol{\Sigma}^{\frac{1}{2}}\mathrm{d}\mathbf{W}_t$$

where $\boldsymbol{\mu}$ is the long-term mean or equilibrium level, $\boldsymbol{\Phi}$ is the rate of mean reversion, determining how quickly the variable returns to its mean, $\boldsymbol{\Sigma}$ is the matrix of volatility or randomness in the process, and $\mathrm{d}\boldsymbol{W}$ is a Wiener process or Brownian motion, which represents random fluctuations.

Value

Returns a list with the following elements:

- `y`: A `t` by `k` matrix of values for the manifest variables.
- `eta`: A `t` by `p` matrix of values for the latent variables.
- `time`: A vector of continuous time points of length `t` starting from 0 with `delta_t` increments.
- `n`: Number of individuals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the brownian motion. *Physical Review*, *36*(5), 823–841. doi:10.1103/physrev.36.823

See Also

Other Simulation of State Space Models Data Functions: `OU2SSM()`, `Sim2Matrix()`, `SimSSM0Fixed()`, `SimSSM0Vary()`, `SimSSM0()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVAR()`

Examples

```
# prepare parameters
set.seed(42)
p <- k <- 2
I <- diag(p)
I_sqrt <- chol(I)
mu0 <- c(-3.0, 1.5)
sigma0_sqrt <- I_sqrt
mu <- c(5.76, 5.18)
phi <- matrix(data = c(0.10, -0.05, -0.05, 0.10), nrow = p)
sigma_sqrt <- chol(
  matrix(data = c(2.79, 0.06, 0.06, 3.27), nrow = p)
)
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta_sqrt <- chol(diag(x = 0.50, nrow = k))
delta_t <- 0.10
time <- 50
```

```
burn_in <- 0

# generate data
ssm <- SimSSMOU(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  mu = mu,
  phi = phi,
  sigma_sqrt = sigma_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

`SimSSMOUFixed` *Simulate Data from an Ornstein–Uhlenbeck Model using a State Space Model Parameterization for n > 1 Individuals (Fixed Parameters)*

Description

This function simulates data from an Ornstein–Uhlenbeck model using a state space model parameterization for `n > 1` individuals. In this model, the parameters are invariant across individuals. See details for more information.

Usage

```
SimSSMOUFixed(
  n,
  mu0,
  sigma0_sqrt,
  mu,
  phi,
  sigma_sqrt,
  nu,
  lambda,
  theta_sqrt,
  delta_t,
  time,
  burn_in
)
```

Arguments

n	Positive integer. Number of individuals.
mu0	Numeric vector. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\mid 0}$).
sigma0_sqrt	Numeric matrix. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\mid 0}$).
mu	Numeric vector. The long-term mean or equilibrium level ($\boldsymbol{\mu}$).
phi	Numeric matrix. The rate of mean reversion, determining how quickly the variable returns to its mean ($\boldsymbol{\Phi}$).
sigma_sqrt	Numeric matrix. Cholesky decomposition of the matrix of volatility or randomness in the process ($\boldsymbol{\Sigma}$).
nu	Numeric vector. Vector of intercepts for the measurement model ($\boldsymbol{\nu}$).
lambda	Numeric matrix. Factor loading matrix linking the latent variables to the observed variables ($\boldsymbol{\Lambda}$).
theta_sqrt	Numeric matrix. Cholesky decomposition of the measurement error covariance matrix ($\boldsymbol{\Theta}$).
delta_t	Numeric. Time interval (δ_t).
time	Positive integer. Number of time points to simulate.
burn_in	Positive integer. Number of burn-in points to exclude before returning the results.

Details

The measurement model is given by

$$\mathbf{y}_{i,t} = \boldsymbol{\nu} + \boldsymbol{\Lambda} \boldsymbol{\eta}_{i,t} + \boldsymbol{\varepsilon}_{i,t} \quad \text{with} \quad \boldsymbol{\varepsilon}_{i,t} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Theta}\right)$$

where $\mathbf{y}_{i,t}$, $\boldsymbol{\eta}_{i,t}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and $\boldsymbol{\nu}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\Theta}$ are model parameters. $\mathbf{y}_{i,t}$ is a vector of observed random variables at time t and individual i, $\boldsymbol{\eta}_{i,t}$ is a vector of latent random variables at time t and individual i, and $\boldsymbol{\varepsilon}_{i,t}$ is a vector of random measurement errors at time t and individual i, while $\boldsymbol{\nu}$ is a vector of intercept, $\boldsymbol{\Lambda}$ is a matrix of factor loadings, and $\boldsymbol{\Theta}$ is the covariance matrix of $\boldsymbol{\varepsilon}$.

The dynamic structure is given by

$$\mathrm{d} \boldsymbol{\eta}_{i,t} = \boldsymbol{\Phi} \left(\boldsymbol{\mu} - \boldsymbol{\eta}_{i,t} \right) \mathrm{d}t + \boldsymbol{\Sigma}^{\frac{1}{2}} \mathrm{d} \mathbf{W}_{i,t}$$

where $\boldsymbol{\mu}$ is the long-term mean or equilibrium level, $\boldsymbol{\Phi}$ is the rate of mean reversion, determining how quickly the variable returns to its mean, $\boldsymbol{\Sigma}$ is the matrix of volatility or randomness in the process, and $\mathrm{d}\boldsymbol{W}$ is a Wiener process or Brownian motion, which represents random fluctuations.

Value

Returns a list of length `n`. Each element is a list with the following elements:

- `y`: A `t` by `k` matrix of values for the manifest variables.
- `eta`: A `t` by `p` matrix of values for the latent variables.
- `time`: A vector of continuous time points of length `t` starting from 0 with `delta_t` increments.
- `id`: A vector of ID numbers of length `t`.
- `n`: Number of individuals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the brownian motion. *Physical Review*, *36*(5), 823–841. doi:10.1103/physrev.36.823

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), Sim2Matrix(), SimSSM0Fixed(), SimSSM0Vary(), SimSSM0(), SimSSMOUVary(), SimSSMOU(), SimSSMVARFixed(), SimSSMVARVary(), SimSSMVAR()

Examples

```
# prepare parameters
set.seed(42)
p <- k <- 2
I <- diag(p)
I_sqrt <- chol(I)
n <- 5
mu0 <- c(-3.0, 1.5)
sigma0_sqrt <- I_sqrt
mu <- c(5.76, 5.18)
phi <- matrix(data = c(0.10, -0.05, -0.05, 0.10), nrow = p)
sigma_sqrt <- chol(
  matrix(data = c(2.79, 0.06, 0.06, 3.27), nrow = p)
)
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta_sqrt <- chol(diag(x = 0.50, nrow = k))
delta_t <- 0.10
time <- 50
burn_in <- 0

# generate data
ssm <- SimSSMOUFixed(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  mu = mu,
  phi = phi,
  sigma_sqrt = sigma_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)
```

```
str(ssm)
```

SimSSMOUVary *Simulate Data from an Ornstein–Uhlenbeck Model using a State Space Model Parameterization for n > 1 Individuals (Varying Parameters)*

Description

This function simulates data from an Ornstein–Uhlenbeck model using a state space model parameterization for n > 1 individuals. In this model, the parameters can vary across individuals.

Usage

```
SimSSMOUVary(
  n,
  mu0,
  sigma0_sqrt,
  mu,
  phi,
  sigma_sqrt,
  nu,
  lambda,
  theta_sqrt,
  delta_t,
  time,
  burn_in
)
```

Arguments

n	Positive integer. Number of individuals.
mu0	Numeric vector. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\mid 0}$).
sigma0_sqrt	Numeric matrix. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\mid 0}$).
mu	List of numeric vectors. The long-term mean or equilibrium level ($\boldsymbol{\mu}$).
phi	List of numeric matrices. The rate of mean reversion, determining how quickly the variable returns to its mean ($\boldsymbol{\Phi}$).
sigma_sqrt	List of numeric matrices. Cholesky decomposition of the matrix of volatility or randomness in the process ($\boldsymbol{\Sigma}$).
nu	Numeric vector. Vector of intercepts for the measurement model ($\boldsymbol{\nu}$).
lambda	Numeric matrix. Factor loading matrix linking the latent variables to the observed variables ($\boldsymbol{\Lambda}$).
theta_sqrt	Numeric matrix. Cholesky decomposition of the measurement error covariance matrix ($\boldsymbol{\Theta}$).

delta_t	Numeric. Time interval (δ_t).
time	Positive integer. Number of time points to simulate.
burn_in	Positive integer. Number of burn-in points to exclude before returning the results.

Details

Parameters can vary across individuals by providing a list of parameter values. If the length of any of the parameters (mu0, sigma0_sqrt, mu, phi, sigma_sqrt, nu, lambda, theta_sqrt) is less the n, the function will cycle through the available values.

Value

Returns a list of length n. Each element is a list with the following elements:

- y: A t by k matrix of values for the manifest variables.
- eta: A t by p matrix of values for the latent variables.
- time: A vector of discrete time points from 1 to t.
- id: A vector of ID numbers of length t.
- n: Number of individuals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the brownian motion. *Physical Review*, *36*(5), 823–841. doi:10.1103/physrev.36.823

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), Sim2Matrix(), SimSSM0Fixed(), SimSSM0Vary(), SimSSM0(), SimSSMOUFixed(), SimSSMOU(), SimSSMVARFixed(), SimSSMVARVary(), SimSSMVAR()

Examples

```
# prepare parameters
# In this example, phi varies across individuals
set.seed(42)
p <- k <- 2
iden <- diag(p)
iden_sqrt <- chol(iden)
n <- 5
mu0 <- list(c(-3.0, 1.5))
sigma0_sqrt <- list(iden_sqrt)
mu <- list(c(5.76, 5.18))
phi <- list(
```

```
  as.matrix(Matrix::expm(diag(x = -0.1, nrow = k))),
  as.matrix(Matrix::expm(diag(x = -0.2, nrow = k))),
  as.matrix(Matrix::expm(diag(x = -0.3, nrow = k))),
  as.matrix(Matrix::expm(diag(x = -0.4, nrow = k))),
  as.matrix(Matrix::expm(diag(x = -0.5, nrow = k)))
)
sigma_sqrt <- list(
  chol(
    matrix(data = c(2.79, 0.06, 0.06, 3.27), nrow = p)
  )
)
nu <- list(rep(x = 0, times = k))
lambda <- list(diag(k))
theta_sqrt <- list(chol(diag(x = 0.50, nrow = k)))
delta_t <- 0.10
time <- 50
burn_in <- 0

ssm <- SimSSMOUVary(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  mu = mu,
  phi = phi,
  sigma_sqrt = sigma_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

`SimSSMVAR` *Simulate Data from the Vector Autoregressive Model using a State Space Model Parameterization (n = 1)*

Description

This function simulates data from the vector autoregressive model using a state space model parameterization. See details for more information.

Usage

```
SimSSMVAR(mu0, sigma0_sqrt, alpha, beta, psi_sqrt, time, burn_in)
```

Arguments

mu0	Numeric vector. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\|0}$).
sigma0_sqrt	Numeric matrix. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\|0}$).
alpha	Numeric vector. Vector of intercepts for the dynamic model ($\boldsymbol{\alpha}$).
beta	Numeric matrix. Transition matrix relating the values of the latent variables at time `t - 1` to those at time `t` ($\boldsymbol{\beta}$).
psi_sqrt	Numeric matrix. Cholesky decomposition of the process noise covariance matrix ($\boldsymbol{\Psi}$).
time	Positive integer. Number of time points to simulate.
burn_in	Positive integer. Number of burn-in points to exclude before returning the results.

Details

The measurement model is given by

$$\mathbf{y}_t = \boldsymbol{\eta}_t.$$

The dynamic structure is given by

$$\boldsymbol{\eta}_t = \boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{\eta}_{t-1} + \boldsymbol{\zeta}_t \quad \text{with} \quad \boldsymbol{\zeta}_t \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Psi}\right)$$

where $\boldsymbol{\eta}_t$, $\boldsymbol{\eta}_{t-1}$, and $\boldsymbol{\zeta}_t$ are random variables and $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\Psi}$ are model parameters. $\boldsymbol{\eta}_t$ is a vector of latent variables at time t, $\boldsymbol{\eta}_{t-1}$ is a vector of latent variables at $t-1$, and $\boldsymbol{\zeta}_t$ is a vector of dynamic noise at time t while $\boldsymbol{\alpha}$ is a vector of intercepts, $\boldsymbol{\beta}$ is a matrix of autoregression and cross regression coefficients, and $\boldsymbol{\Psi}$ is the covariance matrix of $\boldsymbol{\zeta}_t$.

Value

Returns a list with the following elements:

- `y`: A `t` by `k` matrix of values for the manifest variables.
- `eta`: A `t` by `p` matrix of values for the latent variables.
- `time`: A vector of discrete time points from 1 to `t`.
- `n`: Number of individuals.

References

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: With R examples*. Springer International Publishing. doi:10.1007/9783319524528

See Also

Other Simulation of State Space Models Data Functions: `OU2SSM()`, `Sim2Matrix()`, `SimSSM0Fixed()`, `SimSSM0Vary()`, `SimSSM0()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`

Examples

```
# prepare parameters
set.seed(42)
k <- 3
I <- diag(k)
I_sqrt <- chol(I)
null_vec <- rep(x = 0, times = k)
mu0 <- null_vec
sigma0_sqrt <- I_sqrt
alpha <- null_vec
beta <- diag(x = 0.5, nrow = k)
psi_sqrt <- I_sqrt
time <- 50
burn_in <- 0

# generate data
ssm <- SimSSMVAR(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

`SimSSMVARFixed` *Simulate Data from a Vector Autoregressive Model using a State Space Model Parameterization for n > 1 Individuals (Fixed Parameters)*

Description

This function simulates data from a vector autoregressive model using a state space model parameterization for `n > 1` individuals. In this model, the parameters are invariant across individuals.

Usage

```
SimSSMVARFixed(n, mu0, sigma0_sqrt, alpha, beta, psi_sqrt, time, burn_in)
```

Arguments

`n`	Positive integer. Number of individuals.
`mu0`	Numeric vector. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\|0}$).
`sigma0_sqrt`	Numeric matrix. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\|0}$).

alpha	Numeric vector. Vector of intercepts for the dynamic model (α).
beta	Numeric matrix. Transition matrix relating the values of the latent variables at time `t - 1` to those at time `t` (β).
psi_sqrt	Numeric matrix. Cholesky decomposition of the process noise covariance matrix (Ψ).
time	Positive integer. Number of time points to simulate.
burn_in	Positive integer. Number of burn-in points to exclude before returning the results.

Value

Returns a list of length `n`. Each element is a list with the following elements:

- `y`: A `t` by `k` matrix of values for the manifest variables.
- `eta`: A `t` by `p` matrix of values for the latent variables.
- `time`: A vector of discrete time points from 1 to `t`.
- `id`: A vector of ID numbers of length `t`.
- `n`: Number of individuals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: With R examples*. Springer International Publishing. doi:10.1007/9783319524528

See Also

Other Simulation of State Space Models Data Functions: `OU2SSM()`, `Sim2Matrix()`, `SimSSM0Fixed()`, `SimSSM0Vary()`, `SimSSM0()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARVary()`, `SimSSMVAR()`

Examples

```
# prepare parameters
set.seed(42)
k <- 3
iden <- diag(k)
iden_sqrt <- chol(iden)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0_sqrt <- iden_sqrt
alpha <- null_vec
beta <- diag(x = 0.5, nrow = k)
psi_sqrt <- iden_sqrt
time <- 50
```

```
burn_in <- 0

ssm <- SimSSMVARFixed(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

`SimSSMVARVary`	*Simulate Data from a Vector Autoregressive Model using a State Space Model Parameterization for n > 1 Individuals (Varying Parameters)*

Description

This function simulates data from a vector autoregressive model using a state space model parameterization for `n > 1` individuals. In this model, the parameters can vary across individuals.

Usage

```
SimSSMVARVary(n, mu0, sigma0_sqrt, alpha, beta, psi_sqrt, time, burn_in)
```

Arguments

`n`	Positive integer. Number of individuals.
`mu0`	List of numeric vectors. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta}\|0}$).
`sigma0_sqrt`	List of numeric matrices. Cholesky decomposition of the covariance matrix of initial latent variable values ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\|0}$).
`alpha`	List of numeric vectors. Vector of intercepts for the dynamic model ($\boldsymbol{\alpha}$).
`beta`	List of numeric matrices. Transition matrix relating the values of the latent variables at time `t - 1` to those at time `t` ($\boldsymbol{\beta}$).
`psi_sqrt`	List of numeric matrices. Cholesky decomposition of the process noise covariance matrix ($\boldsymbol{\Psi}$).
`time`	Positive integer. Number of time points to simulate.
`burn_in`	Positive integer. Number of burn-in points to exclude before returning the results.

Details

Parameters can vary across individuals by providing a list of parameter values. If the length of any of the parameters (mu0, sigma0_sqrt, alpha, beta, and psi_sqrt) is less the n, the function will cycle through the available values.

Value

Returns a list of length n. Each element is a list with the following elements:

- y: A t by k matrix of values for the manifest variables.
- eta: A t by p matrix of values for the latent variables.
- time: A vector of discrete time points from 1 to t.
- id: A vector of ID numbers of length t.
- n: Number of individuals.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: With R examples*. Springer International Publishing. doi:10.1007/9783319524528

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), Sim2Matrix(), SimSSM0Fixed(), SimSSM0Vary(), SimSSM0(), SimSSMOUFixed(), SimSSMOUVary(), SimSSMOU(), SimSSMVARFixed(), SimSSMVAR()

Examples

```
# prepare parameters
# In this example, beta varies across individuals
set.seed(42)
k <- 3
iden <- diag(k)
iden_sqrt <- chol(iden)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- list(null_vec)
sigma0_sqrt <- list(iden_sqrt)
alpha <- list(null_vec)
beta <- list(
  diag(x = 0.1, nrow = k),
  diag(x = 0.2, nrow = k),
  diag(x = 0.3, nrow = k),
  diag(x = 0.4, nrow = k),
  diag(x = 0.5, nrow = k)
)
```

```
psi_sqrt <- list(iden_sqrt)
time <- 50
burn_in <- 0

ssm <- SimSSMVARVary(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

Index