Package ‘simStateSpace’

November 28, 2023

Title Simulate Data from State Space Models

Version 1.0.1.9000

Description Provides a streamlined and user-friendly framework
for simulating data in state space models,
particularly when the number of subjects/units (n) exceeds one,
a scenario commonly encountered in social and behavioral sciences.
For an introduction to state space models in social and behavioral sciences,
refer to Chow, Ho, Hamaker, and Dolan (2010) <doi:10.1080/10705511003661553>.

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 2
Sim2Matrix 3
SimSSM 5
SimSSMFixed 9

SimSSMLinGrowth 14
SimSSMOU 17
SimSSMOUFixed 22
SimSSMOUVary 26
SimSSMVAR 30
SimSSMVARFixed 33
SimSSMVARVary 36
SimSSMVary 39

Index **43**

`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

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

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

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

Value

Returns a list of state space parameters:

- `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`: Numeric matrix. The process noise covariance matrix (Ψ).

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Simulation of State Space Models Data Functions: `Sim2Matrix()`, `SimSSMFixed()`, `SimSSMLinGrowth()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVAR()`, `SimSSMVary()`, `SimSSM()`

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 `SimSSM()`, `SimSSMOU()`, `SimSSMVAR()`, `SimSSMFixed()`, `SimSSMOUFixed()`, `SimSSMVARFixed()`, `SimSSMVary()`, `SimSSMOUVary()`, or `SimSSMVARVary()` to a matrix.

Usage

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

Arguments

x	R object. Output of SimSSM(), SimSSMOU(), SimSSMVAR(), SimSSMFixed(), SimSSMOUFixed(), SimSSMVARFixed(), SimSSMVary(), SimSSMOUVary(), or SimSSMVARVary().
eta	Logical. If eta = TRUE, include eta. If eta = FALSE, exclude eta.
long	Logical. If long = TRUE, use long format. If long = FALSE, use wide format.

Value

Returns a matrix of simulated data.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), SimSSMFixed(), SimSSMLinGrowth(), SimSSMOUFixed(), SimSSMOUVary(), SimSSMOU(), SimSSMVARFixed(), SimSSMVARVary(), SimSSMVAR(), SimSSMVary(), SimSSM()

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 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 = 0.50, nrow = k)
psi_sqrt <- iden_sqrt
nu <- null_vec
lambda <- iden
theta_sqrt <- chol(diag(x = 0.50, nrow = k))
time <- 50
burn_in <- 0

# generate data
ssm <- SimSSM(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
```

```
    type = 0,
    time = time,
    burn_in = burn_in
  )

  # list to matrix
  mat <- Sim2Matrix(ssm, long = TRUE)
  str(mat)
  head(mat)
  mat <- Sim2Matrix(ssm, long = FALSE)
  str(mat)
  head(mat)

  # generate data
  ssm <- SimSSMFixed(
    n = n,
    mu0 = mu0,
    sigma0_sqrt = sigma0_sqrt,
    alpha = alpha,
    beta = beta,
    psi_sqrt = psi_sqrt,
    nu = nu,
    lambda = lambda,
    theta_sqrt = theta_sqrt,
    type = 0,
    time = time,
    burn_in = burn_in
  )

  # list to matrix
  mat <- Sim2Matrix(ssm, long = TRUE)
  str(mat)
  head(mat)
  mat <- Sim2Matrix(ssm, long = FALSE)
  str(mat)
  head(mat)
```

SimSSM *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

```
SimSSM(
  mu0,
  sigma0_sqrt,
```

```
  alpha,
  beta,
  psi_sqrt,
  nu,
  lambda,
  theta_sqrt,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  time,
  burn_in = 0
)
```

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 ($\boldsymbol{\Theta}$).
gamma_y	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the observed variables at time `t` ($\boldsymbol{\Gamma}_{\mathbf{y}}$).
gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the latent variables at time `t` ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
x	Numeric matrix. The matrix of observed covariates in `type = 1` or `type = 2`. The number of rows should be equal to `time + burn_in`.
type	Integer. State space model type.
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

Type 0:

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, $\boldsymbol{\eta}_t$ is a vector of latent random variables, and $\boldsymbol{\varepsilon}_t$ is a vector of random measurement errors, at time t. $\boldsymbol{\nu}$ is a vector of intercepts, $\boldsymbol{\Lambda}$ 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. $\boldsymbol{\alpha}$ is a vector of intercepts, $\boldsymbol{\beta}$ is a matrix of autoregression and cross regression coefficients, and $\boldsymbol{\Psi}$ is the covariance matrix of $\boldsymbol{\zeta}_t$.

Type 1:

The measurement model is given by

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

The dynamic structure is given by

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

where $\mathbf{x}_t$ is a vector of covariates at time t, and $\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$ is the coefficient matrix linking the covariates to the latent variables.

Type 2:

The measurement model is given by

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

where $\boldsymbol{\Gamma}_{\mathbf{y}}$ is the coefficient matrix linking the covariates to the observed variables.

The dynamic structure is given by

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

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.
- `x`: A `t` by `j` matrix of values for the covariates.
- `time`: A vector of discrete time points from 0 to `t - 1`.
- `id`: A vector of ones.

References

Chow, S.-M., Ho, M. R., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and differences between structural equation modeling and state-space modeling techniques. *Structural Equation Modeling: A Multidisciplinary Journal*, 17(2), 303–332. doi:10.1080/10705511003661553

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), Sim2Matrix(), SimSSMFixed(), SimSSMLinGrowth(), SimSSMOUFixed(), SimSSMOUVary(), SimSSMOU(), SimSSMVARFixed(), SimSSMVARVary(), SimSSMVAR(), SimSSMVary()

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
iden <- diag(k)
iden_sqrt <- chol(iden)
null_vec <- rep(x = 0, times = k)
mu0 <- null_vec
sigma0_sqrt <- iden_sqrt
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi_sqrt <- iden_sqrt
nu <- null_vec
lambda <- iden
theta_sqrt <- chol(diag(x = 0.50, nrow = k))
time <- 50
burn_in <- 0
gamma_y <- gamma_eta <- 0.10 * diag(k)
x <- matrix(
  data = rnorm(n = k * (time + burn_in)),
  ncol = k
)

# Type 0
ssm <- SimSSM(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  type = 0,
  time = time,
  burn_in = burn_in
)

str(ssm)

# Type 1
ssm <- SimSSM(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
```

```
    psi_sqrt = psi_sqrt,
    nu = nu,
    lambda = lambda,
    theta_sqrt = theta_sqrt,
    gamma_eta = gamma_eta,
    x = x,
    type = 1,
    time = time,
    burn_in = burn_in
  )

  str(ssm)

  # Type 2
  ssm <- SimSSM(
    mu0 = mu0,
    sigma0_sqrt = sigma0_sqrt,
    alpha = alpha,
    beta = beta,
    psi_sqrt = psi_sqrt,
    nu = nu,
    lambda = lambda,
    theta_sqrt = theta_sqrt,
    gamma_y = gamma_y,
    gamma_eta = gamma_eta,
    x = x,
    type = 2,
    time = time,
    burn_in = burn_in
  )

  str(ssm)
```

SimSSMFixed *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

```
SimSSMFixed(
  n,
  mu0,
  sigma0_sqrt,
  alpha,
```

```
  beta,
  psi_sqrt,
  nu,
  lambda,
  theta_sqrt,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  time,
  burn_in = 0
)
```

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 ($\boldsymbol{\alpha}$).
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 ($\boldsymbol{\Theta}$).
gamma_y	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the observed variables at time `t` ($\boldsymbol{\Gamma}_{\mathbf{y}}$).
gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the latent variables at time `t` ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
x	A list of length `n` of numeric matrices. Each element of the list is a matrix of observed covariates in `type = 1` or `type = 2`. The number of rows in each matrix should be equal to `time + burn_in`.
type	Integer. State space model type.
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

Type 0:

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, $\boldsymbol{\eta}_{i,t}$ is a vector of latent random variables, and $\boldsymbol{\varepsilon}_{i,t}$ is a vector of random measurement errors, at time t and individual i. $\boldsymbol{\nu}$ is a vector of intercepts, $\boldsymbol{\Lambda}$ 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. $\boldsymbol{\alpha}$ is a vector of intercepts, $\boldsymbol{\beta}$ is a matrix of autoregression and cross regression coefficients, and $\boldsymbol{\Psi}$ is the covariance matrix of $\boldsymbol{\zeta}_{i,t}$.

Type 1:

The measurement model is given by

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

The dynamic structure is given by

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

where $\mathbf{x}_{i,t}$ is a vector of covariates at time t and individual i, and $\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$ is the coefficient matrix linking the covariates to the latent variables.

Type 2:

The measurement model is given by

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

where $\boldsymbol{\Gamma}_{\mathbf{y}}$ is the coefficient matrix linking the covariates to the observed variables.

The dynamic structure is given by

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

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.
- `x`: A `t` by `j` matrix of values for the covariates.
- `time`: A vector of discrete time points from 1 to `t`.
- `id`: A vector of ID numbers of length `t`.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Chow, S.-M., Ho, M. R., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and differences between structural equation modeling and state-space modeling techniques. *Structural Equation Modeling: A Multidisciplinary Journal*, 17(2), 303–332. doi:10.1080/10705511003661553

See Also

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

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 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 = 0.50, nrow = k)
psi_sqrt <- iden_sqrt
nu <- null_vec
lambda <- iden
theta_sqrt <- chol(diag(x = 0.50, nrow = k))
time <- 50
burn_in <- 0
gamma_y <- gamma_eta <- 0.10 * diag(k)
x <- lapply(
  X = seq_len(n),
  FUN = function(i) {
    return(
      matrix(
        data = rnorm(n = k * (time + burn_in)),
        ncol = k
      )
    )
  }
)

# Type 0
ssm <- SimSSMFixed(
  n = n,
  mu0 = mu0,
```

```
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  type = 0,
  time = time,
  burn_in = burn_in
)

str(ssm)

# Type 1
ssm <- SimSSMFixed(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  time = time,
  burn_in = burn_in
)

str(ssm)

# Type 2
ssm <- SimSSMFixed(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  type = 2,
  time = time,
  burn_in = burn_in
)
```

```
str(ssm)
```

`SimSSMLinGrowth` *Simulate Data from a Linear Growth Curve Model*

Description

This function simulates data from a linear growth curve model.

Usage

```
SimSSMLinGrowth(
  n,
  mu0,
  sigma0_sqrt,
  theta_sqrt,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  time
)
```

Arguments

`n`	Positive integer. Number of individuals.
`mu0`	Numeric vector. A vector of length two. The first element is the mean of the intercept, and the second element is the mean of the slope.
`sigma0_sqrt`	Numeric matrix. Cholesky decomposition of the covariance matrix of the intercept and the slope.
`theta_sqrt`	Numeric. Square root of the common measurement error variance.
`gamma_y`	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to `y` at time `t` ($\boldsymbol{\Gamma}_{\mathbf{y}}$).
`gamma_eta`	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the latent variables (intercept and slope) at time `t` ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
`x`	A list of length `n` of numeric matrices. Each element of the list is a matrix of observed covariates in `type = 1` or `type = 2`. The number of rows in each matrix should be equal to `time`.
`type`	Integer. State space model type.
`time`	Positive integer. Number of time points to simulate.

Details

Type 0:

The measurement model is given by

$$y_{i,t} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \eta_{0_{i,t}} \\ \eta_{1_{i,t}} \end{pmatrix} + \boldsymbol{\varepsilon}_{i,t} \quad \text{with} \quad \boldsymbol{\varepsilon}_{i,t} \sim \mathcal{N}\left(0, \theta^2\right)$$

where $y_{i,t}$, $\eta_{0_{i,t}}$, $\eta_{1_{i,t}}$, and $\boldsymbol{\varepsilon}_{i,t}$ are random variables and and θ^2 is a model parameter. $y_{i,t}$ is a vector of observed random variables at time t and individual i, $\eta_{0_{i,t}}$ and $\eta_{1_{i,t}}$ form 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, and θ^2 is the variance of $\boldsymbol{\varepsilon}$.

The dynamic structure is given by

$$\begin{pmatrix} \eta_{0_{i,t}} \\ \eta_{1_{i,t}} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_{0_{i,t-1}} \\ \eta_{1_{i,t-1}} \end{pmatrix}.$$

The mean vector and covariance matrix of the intercept and slope are captured in the mean vector and covariance matrix of the initial condition given by

$$\boldsymbol{\mu}_{\boldsymbol{\eta}|0} = \begin{pmatrix} \mu_{\eta_0} \\ \mu_{\eta_1} \end{pmatrix} \quad \text{and,}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\eta}|0} = \begin{pmatrix} \sigma^2_{\eta_0} & \sigma_{\eta_0,\eta_1} \\ \sigma_{\eta_1,\eta_0} & \sigma^2_{\eta_1} \end{pmatrix}.$$

Type 1:

The measurement model is given by

$$y_{i,t} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \eta_{0_{i,t}} \\ \eta_{1_{i,t}} \end{pmatrix} + \boldsymbol{\varepsilon}_{i,t} \quad \text{with} \quad \boldsymbol{\varepsilon}_{i,t} \sim \mathcal{N}\left(0, \theta^2\right).$$

The dynamic structure is given by

$$\begin{pmatrix} \eta_{0_{i,t}} \\ \eta_{1_{i,t}} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_{0_{i,t-1}} \\ \eta_{1_{i,t-1}} \end{pmatrix} + \boldsymbol{\Gamma}_{\boldsymbol{\eta}} \mathbf{x}_{i,t}$$

where $\mathbf{x}_{i,t}$ is a vector of covariates at time t and individual i, and $\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$ is the coefficient matrix linking the covariates to the latent variables.

Type 2:

The measurement model is given by

$$y_{i,t} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \eta_{0_{i,t}} \\ \eta_{1_{i,t}} \end{pmatrix} + \boldsymbol{\Gamma}_{\mathbf{y}} \mathbf{x}_{i,t} + \boldsymbol{\varepsilon}_{i,t} \quad \text{with} \quad \boldsymbol{\varepsilon}_{i,t} \sim \mathcal{N}\left(0, \theta^2\right)$$

where $\boldsymbol{\Gamma}_{\mathbf{y}}$ is the coefficient matrix linking the covariates to the observed variables.

The dynamic structure is given by

$$\begin{pmatrix} \eta_{0_{i,t}} \\ \eta_{1_{i,t}} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_{0_{i,t-1}} \\ \eta_{1_{i,t-1}} \end{pmatrix} + \boldsymbol{\Gamma}_{\boldsymbol{\eta}} \mathbf{x}_{i,t}.$$

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.
- x: A t by j matrix of values for the covariates.
- time: A vector of discrete time points from 1 to t.
- id: A vector of ID numbers of length t.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Chow, S.-M., Ho, M. R., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and differences between structural equation modeling and state-space modeling techniques. *Structural Equation Modeling: A Multidisciplinary Journal*, 17(2), 303–332. doi:10.1080/10705511003661553

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), Sim2Matrix(), SimSSMFixed(), SimSSMOUFixed(), SimSSMOUVary(), SimSSMOU(), SimSSMVARFixed(), SimSSMVARVary(), SimSSMVAR(), SimSSMVary(), SimSSM()

Examples

```
# prepare parameters
set.seed(42)
n <- 5
mu0 <- c(0.615, 1.006)
sigma0 <- matrix(
  data = c(
    1.932,
    0.618,
    0.618,
    0.587
  ),
  nrow = 2
)
sigma0_sqrt <- chol(sigma0)
theta <- 0.6
theta_sqrt <- sqrt(theta)
time <- 10
gamma_y <- matrix(data = 0.10, nrow = 1, ncol = 2)
gamma_eta <- matrix(data = 0.10, nrow = 2, ncol = 2)
x <- lapply(
  X = seq_len(n),
  FUN = function(i) {
    return(
```

```
      matrix(
        data = rnorm(n = 2 * time),
        ncol = 2
      )
    )
  }
)

# Type 0
ssm <- SimSSMLinGrowth(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  theta_sqrt = theta_sqrt,
  type = 0,
  time = time
)

str(ssm)

# Type 1
ssm <- SimSSMLinGrowth(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  theta_sqrt = theta_sqrt,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  time = time
)

str(ssm)

# Type 2
ssm <- SimSSMLinGrowth(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  theta_sqrt = theta_sqrt,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  type = 2,
  time = time
)

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,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  delta_t,
  time,
  burn_in = 0
)
```

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 ($\boldsymbol{\Theta}$).
`gamma_y`	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the observed variables at time `t` ($\boldsymbol{\Gamma}_{\mathbf{y}}$).

gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
x	Numeric matrix. The matrix of observed covariates in type = 1 or type = 2. The number of rows should be equal to time + burn_in.
type	Integer. State space model type.
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

Type 0:
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, $\boldsymbol{\eta}_t$ is a vector of latent random variables, and $\boldsymbol{\varepsilon}_t$ is a vector of random measurement errors, at time t. $\boldsymbol{\nu}$ is a vector of intercepts, $\boldsymbol{\Lambda}$ 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.

Type 1:
The measurement model is given by

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

The dynamic structure is given by

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

where $\mathbf{x}_t$ is a vector of covariates at time t, and $\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$ is the coefficient matrix linking the covariates to the latent variables.

Type 2:
The measurement model is given by

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

where $\boldsymbol{\Gamma}_{\mathbf{y}}$ is the coefficient matrix linking the covariates to the observed variables.
The dynamic structure is given by

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

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 continuous time points of length t starting from 0 with delta_t increments.
- id: A vector of ones.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Chow, S.-M., Losardo, D., Park, J., & Molenaar, P. C. M. (2023). Continuous-time dynamic models: Connections to structural equation models and other discrete-time models. In R. H. Hoyle (Ed.), Handbook of structural equation modeling (2nd ed.). The Guilford Press.

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(), SimSSMFixed(), SimSSMLinGrowth(), SimSSMOUFixed(), SimSSMOUVary(), SimSSMVARFixed(), SimSSMVARVary(), SimSSMVAR(), SimSSMVary(), SimSSM()

Examples

```
# prepare parameters
set.seed(42)
p <- k <- 2
iden <- diag(p)
iden_sqrt <- chol(iden)
mu0 <- c(-3.0, 1.5)
sigma0_sqrt <- iden_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
gamma_y <- gamma_eta <- 0.10 * diag(k)
x <- matrix(
  data = rnorm(n = k * (time + burn_in)),
  ncol = k
```

```
)

# Type 0
ssm <- SimSSMOU(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  mu = mu,
  phi = phi,
  sigma_sqrt = sigma_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  type = 0,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

str(ssm)

# Type 1
ssm <- SimSSMOU(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  mu = mu,
  phi = phi,
  sigma_sqrt = sigma_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

str(ssm)

# Type 2
ssm <- SimSSMOU(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  mu = mu,
  phi = phi,
  sigma_sqrt = sigma_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
```

```
    type = 2,
    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,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  delta_t,
  time,
  burn_in = 0
)
```

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 ($\boldsymbol{\Theta}$).
gamma_y	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the observed variables at time `t` ($\boldsymbol{\Gamma}_{\mathbf{y}}$).
gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the latent variables at time `t` ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
x	Numeric matrix. The matrix of observed covariates in `type = 1` or `type = 2`. The number of rows should be equal to `time + burn_in`.
type	Integer. State space model type.
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

Type 0:

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, $\boldsymbol{\eta}_{i,t}$ is a vector of latent random variables, and $\boldsymbol{\varepsilon}_{i,t}$ is a vector of random measurement errors, at time t and individual i. $\boldsymbol{\nu}$ is a vector of intercepts, $\boldsymbol{\Lambda}$ 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.

Type 1:

The measurement model is given by

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

The dynamic structure is given by

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

where $\mathbf{x}_{i,t}$ is a vector of covariates at time t and individual i, and $\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$ is the coefficient matrix linking the covariates to the latent variables.

Type 2:
The measurement model is given by

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

where $\boldsymbol{\Gamma}_{\mathbf{y}}$ is the coefficient matrix linking the covariates to the observed variables.
The dynamic structure is given by

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

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.
- `x`: A `t` by `j` matrix of values for the covariates.
- `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

Chow, S.-M., Losardo, D., Park, J., & Molenaar, P. C. M. (2023). Continuous-time dynamic models: Connections to structural equation models and other discrete-time models. In R. H. Hoyle (Ed.), Handbook of structural equation modeling (2nd ed.). The Guilford Press.

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()`, `SimSSMFixed()`, `SimSSMLinGrowth()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVAR()`, `SimSSMVary()`, `SimSSM()`

Examples

```
# prepare parameters
set.seed(42)
p <- k <- 2
iden <- diag(p)
iden_sqrt <- chol(iden)
```

```
n <- 5
mu0 <- c(-3.0, 1.5)
sigma0_sqrt <- iden_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
gamma_y <- gamma_eta <- 0.10 * diag(k)
x <- lapply(
  X = seq_len(n),
  FUN = function(i) {
    return(
      matrix(
        data = rnorm(n = k * (time + burn_in)),
        ncol = k
      )
    )
  }
)

# Type 0
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,
  type = 0,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

str(ssm)

# Type 1
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,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

str(ssm)

# Type 2
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,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  type = 2,
  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,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  delta_t,
  time,
  burn_in = 0
)
```

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 ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\|0}$).
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 ($\boldsymbol{\Theta}$).
gamma_y	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the observed variables at time `t` ($\boldsymbol{\Gamma}_{\mathbf{y}}$).
gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the latent variables at time `t` ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
x	Numeric matrix. The matrix of observed covariates in `type = 1` or `type = 2`. The number of rows should be equal to `time + burn_in`.
type	Integer. State space model type.
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`, `gamma_y`, or `gamma_eta`) 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.
- `x`: A `t` by `j` matrix of values for the covariates.
- `time`: A vector of discrete time points from 1 to `t`.
- `id`: A vector of ID numbers of length `t`.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Chow, S.-M., Losardo, D., Park, J., & Molenaar, P. C. M. (2023). Continuous-time dynamic models: Connections to structural equation models and other discrete-time models. In R. H. Hoyle (Ed.), Handbook of structural equation modeling (2nd ed.). The Guilford Press.

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()`, `SimSSMFixed()`, `SimSSMLinGrowth()`, `SimSSMOUFixed()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVAR()`, `SimSSMVary()`, `SimSSM()`

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
  gamma_y <- gamma_eta <- list(0.10 * diag(k))
  x <- lapply(
    X = seq_len(n),
    FUN = function(i) {
      return(
        matrix(
          data = rnorm(n = k * (time + burn_in)),
          ncol = k
        )
      )
    }
  )

  # Type 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,
    type = 0,
    delta_t = delta_t,
    time = time,
    burn_in = burn_in
  )

  str(ssm)

  # Type 1
  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,
    gamma_eta = gamma_eta,
    x = x,
    type = 1,
    delta_t = delta_t,
    time = time,
    burn_in = burn_in
  )

  str(ssm)

  # Type 2
  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,
    gamma_y = gamma_y,
    gamma_eta = gamma_eta,
    x = x,
    type = 2,
    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,
  gamma_eta = NULL,
  x = NULL,
  time = 0,
  burn_in = 0
)
```

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 ($\boldsymbol{\Sigma}_{\boldsymbol{\eta}\mid 0}$).
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 ($\boldsymbol{\Psi}$).
gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the latent variables at time `t` ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
x	Numeric matrix. The matrix of observed covariates in `type = 1` or `type = 2`. The number of rows should be equal to `time + burn_in`.
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 time $t-1$, and $\boldsymbol{\zeta}_t$ is a vector of dynamic noise at time t. $\boldsymbol{\alpha}$ is a vector of intercepts, $\boldsymbol{\beta}$ is a matrix of autoregression and cross regression coefficients, and $\boldsymbol{\Psi}$ is the covariance matrix of $\boldsymbol{\zeta}_t$.

Note that when `gamma_eta` and `x` are not `NULL`, the dynamic structure is given by

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

where $\mathbf{x}_t$ is a vector of covariates at time t, and $\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$ is the coefficient matrix linking the covariates to the latent variables.

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.
- `x`: A `t` by `j` matrix of values for the covariates.
- `time`: A vector of discrete time points from 0 to `t - 1`.
- `id`: A vector of ones.

References

Chow, S.-M., Ho, M. R., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and differences between structural equation modeling and state-space modeling techniques. *Structural Equation Modeling: A Multidisciplinary Journal*, 17(2), 303–332. doi:10.1080/10705511003661553

See Also

Other Simulation of State Space Models Data Functions: `OU2SSM()`, `Sim2Matrix()`, `SimSSMFixed()`, `SimSSMLinGrowth()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARFixed()`, `SimSSMVARVary()`, `SimSSMVary()`, `SimSSM()`

Examples

```
# prepare parameters
set.seed(42)
k <- 3
iden <- diag(k)
iden_sqrt <- chol(iden)
null_vec <- rep(x = 0, times = k)
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
gamma_eta <- 0.10 * diag(k)
x <- matrix(
  data = rnorm(n = k * (time + burn_in)),
  ncol = k
)

# No covariates
ssm <- SimSSMVAR(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  time = time,
```

```
  burn_in = burn_in
)

str(ssm)

# With covariates
ssm <- SimSSMVAR(
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  gamma_eta = gamma_eta,
  x = x,
  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,
  gamma_eta = NULL,
  x = NULL,
  time = 0,
  burn_in = 0
)
```

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 ($\boldsymbol{\alpha}$).
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 ($\boldsymbol{\Psi}$).
gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time `t` to the latent variables at time `t` ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
x	A list of length `n` of numeric matrices. Each element of the list is a matrix of observed covariates in `type = 1` or `type = 2`. The number of rows in each matrix should be equal to `time + burn_in`.
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.
- `x`: A `t` by `j` matrix of values for the covariates.
- `time`: A vector of discrete time points from 1 to `t`.
- `id`: A vector of ID numbers of length `t`.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Chow, S.-M., Ho, M. R., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and differences between structural equation modeling and state-space modeling techniques. *Structural Equation Modeling: A Multidisciplinary Journal*, 17(2), 303–332. doi:10.1080/10705511003661553

See Also

Other Simulation of State Space Models Data Functions: `OU2SSM()`, `Sim2Matrix()`, `SimSSMFixed()`, `SimSSMLinGrowth()`, `SimSSMOUFixed()`, `SimSSMOUVary()`, `SimSSMOU()`, `SimSSMVARVary()`, `SimSSMVAR()`, `SimSSMVary()`, `SimSSM()`

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
gamma_eta <- 0.10 * diag(k)
x <- lapply(
  X = seq_len(n),
  FUN = function(i) {
    return(
      matrix(
        data = rnorm(n = k * (time + burn_in)),
        ncol = k
      )
    )
  }
)

# No covariates
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)

# With covariates
ssm <- SimSSMVARFixed(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  gamma_eta = gamma_eta,
  x = x,
```

```
  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,
  gamma_eta = NULL,
  x = NULL,
  time = 0,
  burn_in = 0
)
```

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 ($\boldsymbol{\Psi}$).
gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).

x	A list of length n of numeric matrices. Each element of the list is a matrix of observed covariates in type = 1 or type = 2. The number of rows in each matrix should be equal to time + burn_in.
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, or gamma_eta) 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.
- x: A t by j matrix of values for the covariates.
- time: A vector of discrete time points from 1 to t.
- id: A vector of ID numbers of length t.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Chow, S.-M., Ho, M. R., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and differences between structural equation modeling and state-space modeling techniques. *Structural Equation Modeling: A Multidisciplinary Journal*, 17(2), 303–332. doi:10.1080/10705511003661553

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), Sim2Matrix(), SimSSMFixed(), SimSSMLinGrowth(), SimSSMOUFixed(), SimSSMOUVary(), SimSSMOU(), SimSSMVARFixed(), SimSSMVAR(), SimSSMVary(), SimSSM()

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
gamma_eta <- list(0.10 * diag(k))
x <- lapply(
  X = seq_len(n),
  FUN = function(i) {
    return(
      matrix(
        data = rnorm(n = k * (time + burn_in)),
        ncol = k
      )
    )
  }
)

# No covariates
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)

# With covariates
ssm <- SimSSMVARVary(
  n = n,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  gamma_eta = gamma_eta,
  x = x,
  time = time,
  burn_in = burn_in
)
```

```
str(ssm)
```

SimSSMVary — *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

```
SimSSMVary(
  n,
  type,
  mu0,
  sigma0_sqrt,
  alpha,
  beta,
  psi_sqrt,
  nu,
  lambda,
  theta_sqrt,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  time = 0,
  burn_in = 0
)
```

Arguments

`n`	Positive integer. Number of individuals.
`type`	Integer. State space model type.
`mu0`	List of numeric vectors. Mean of initial latent variable values ($\boldsymbol{\mu}_{\boldsymbol{\eta} \mid 0}$).
`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 ($\boldsymbol{\Theta}$).
gamma_y	Numeric matrix. Matrix relating the values of the covariate matrix at time t to the observed variables at time t ($\boldsymbol{\Gamma}_{\mathbf{y}}$).
gamma_eta	Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t ($\boldsymbol{\Gamma}_{\boldsymbol{\eta}}$).
x	A list of length n of numeric matrices. Each element of the list is a matrix of observed covariates in type = 1 or type = 2. The number of rows in each matrix should be equal to time + burn_in.
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, theta_sqrt, gamma_y, or gamma_eta) 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.
- x: A t by j matrix of values for the covariates.
- time: A vector of discrete time points from 1 to t.
- id: A vector of ID numbers of length t.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Chow, S.-M., Ho, M. R., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and differences between structural equation modeling and state-space modeling techniques. *Structural Equation Modeling: A Multidisciplinary Journal*, 17(2), 303–332. doi:10.1080/10705511003661553

See Also

Other Simulation of State Space Models Data Functions: OU2SSM(), Sim2Matrix(), SimSSMFixed(), SimSSMLinGrowth(), SimSSMOUFixed(), SimSSMOUVary(), SimSSMOU(), SimSSMVARFixed(), SimSSMVARVary(), SimSSMVAR(), SimSSM()

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
gamma_y <- gamma_eta <- list(0.10 * diag(k))
x <- lapply(
  X = seq_len(n),
  FUN = function(i) {
    return(
      matrix(
        data = rnorm(n = k * (time + burn_in)),
        ncol = k
      )
    )
  }
)

# Type 0
ssm <- SimSSMVary(
  n = n,
  type = 0,
  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)

# Type 1
ssm <- SimSSMVary(
  n = n,
  type = 1,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  gamma_eta = gamma_eta,
  x = x,
  time = time,
  burn_in = burn_in
)

str(ssm)

# Type 2
ssm <- SimSSMVary(
  n = n,
  type = 2,
  mu0 = mu0,
  sigma0_sqrt = sigma0_sqrt,
  alpha = alpha,
  beta = beta,
  psi_sqrt = psi_sqrt,
  nu = nu,
  lambda = lambda,
  theta_sqrt = theta_sqrt,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  time = time,
  burn_in = burn_in
)

str(ssm)
```

Index

∗ **Simulation of State Space Models Data Functions**
OU2SSM, 2
Sim2Matrix, 3
SimSSM, 5
SimSSMFixed, 9
SimSSMLinGrowth, 14
SimSSMOU, 18
SimSSMOUFixed, 22
SimSSMOUVary, 26
SimSSMVAR, 30
SimSSMVARFixed, 33
SimSSMVARVary, 36
SimSSMVary, 39
∗ **growth**
SimSSMLinGrowth, 14
∗ **misc**
Sim2Matrix, 3
∗ **ou**
OU2SSM, 2
SimSSMOU, 18
SimSSMOUFixed, 22
SimSSMOUVary, 26
∗ **simStateSpace**
OU2SSM, 2
Sim2Matrix, 3
SimSSM, 5
SimSSMFixed, 9
SimSSMLinGrowth, 14
SimSSMOU, 18
SimSSMOUFixed, 22
SimSSMOUVary, 26
SimSSMVAR, 30
SimSSMVARFixed, 33
SimSSMVARVary, 36
SimSSMVary, 39
∗ **sim**
OU2SSM, 2
SimSSM, 5
SimSSMFixed, 9
SimSSMLinGrowth, 14
SimSSMOU, 18
SimSSMOUFixed, 22
SimSSMOUVary, 26
SimSSMVAR, 30
SimSSMVARFixed, 33
SimSSMVARVary, 36
SimSSMVary, 39
∗ **ssm**
SimSSM, 5
SimSSMFixed, 9
SimSSMVary, 39
∗ **var**
SimSSMVAR, 30
SimSSMVARFixed, 33
SimSSMVARVary, 36

OU2SSM, 2, *4*, *8*, *12*, *16*, *20*, *24*, *28*, *32*, *34*, *37*, *40*

Sim2Matrix, *3*, 3, *8*, *12*, *16*, *20*, *24*, *28*, *32*, *34*, *37*, *40*
SimSSM, *3, 4*, 5, *12*, *16*, *20*, *24*, *28*, *32*, *34*, *37*, *40*
SimSSM(), *3, 4*
SimSSMFixed, *3, 4*, *8*, 9, *16*, *20*, *24*, *28*, *32*, *34*, *37*, *40*
SimSSMFixed(), *3, 4*
SimSSMLinGrowth, *3, 4*, *8*, *12*, 14, *20*, *24*, *28*, *32*, *34*, *37*, *40*
SimSSMOU, *3, 4*, *8*, *12*, *16*, 17, *24*, *28*, *32*, *34*, *37*, *40*
SimSSMOU(), *3, 4*
SimSSMOUFixed, *3, 4*, *8*, *12*, *16*, *20*, 22, *28*, *32*, *34*, *37*, *40*
SimSSMOUFixed(), *3, 4*
SimSSMOUVary, *3, 4*, *8*, *12*, *16*, *20*, *24*, 26, *32*, *34*, *37*, *40*
SimSSMOUVary(), *3, 4*

SimSSMVAR, *3, 4*, *8*, *12*, *16*, *20*, *24*, *28*, 30, *34*, *37*, *40*
SimSSMVAR(), *3, 4*
SimSSMVARFixed, *3, 4*, *8*, *12*, *16*, *20*, *24*, *28*, *32*, 33, *37*, *40*
SimSSMVARFixed(), *3, 4*
SimSSMVARVary, *3, 4*, *8*, *12*, *16*, *20*, *24*, *28*, *32*, *34*, 36, *40*
SimSSMVARVary(), *3, 4*
SimSSMVary, *3, 4*, *8*, *12*, *16*, *20*, *24*, *28*, *32*, *34*, *37*, 39
SimSSMVary(), *3, 4*