

Package ‘simStateSpace’

January 14, 2024

Title Simulate Data from State Space Models

Version 1.1.0

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](https://doi.org/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 7.3.0

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:

| | |
|---------------------------------------|---|
| as.data.frame.simstatespace | 2 |
| as.matrix.simstatespace | 4 |
| OU2SSM | 5 |
| plot.simstatespace | 7 |

| | |
|--------------------------------|-----------|
| print.simstatespace | 8 |
| SimSSM | 10 |
| SimSSMFixed | 14 |
| SimSSMIVary | 18 |
| SimSSMLinGrowth | 22 |
| SimSSMLinGrowthIVary | 26 |
| SimSSMOU | 29 |
| SimSSMOUFixed | 33 |
| SimSSMOUIVary | 38 |
| SimSSMVAR | 42 |
| SimSSMVARFixed | 45 |
| SimSSMVARIVary | 48 |
| Index | 52 |

| |
|--|
| as.data.frame.simstatespace |
| <i>Coerce an Object of Class simstatespace to a Data Frame</i> |

Description

Coerce an Object of Class simstatespace to a Data Frame

Usage

```
## S3 method for class 'simstatespace'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  eta = FALSE,
  long = TRUE,
  ...
)
```

Arguments

| | |
|-----------|---|
| x | Object of class simstatespace. |
| row.names | NULL or character vector giving the row names for the data frame. Missing values are not allowed. |
| optional | Logical. If TRUE, setting row names and converting column names is optional. |
| 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. |
| ... | Additional arguments. |

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```

# prepare parameters
set.seed(42)
k <- p <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0 <- iden
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi <- iden
nu <- null_vec
lambda <- iden
theta <- 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 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  time = time,
  burn_in = burn_in
)

head(as.data.frame(ssm))
head(as.data.frame(ssm, long = FALSE))

```

as.matrix.simstatespace

Coerce an Object of Class simstatespace to a Matrix

Description

Coerce an Object of Class simstatespace to a Matrix

Usage

```
## S3 method for class 'simstatespace'
as.matrix(x, eta = FALSE, long = TRUE, ...)
```

Arguments

| | |
|------|---|
| x | Object of class simstatespace. |
| 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. |
| ... | Additional arguments. |

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0 <- iden
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi <- iden
nu <- null_vec
lambda <- iden
theta <- 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 <- SimSSMFxed(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  time = time,
  burn_in = burn_in
)

head(as.matrix(ssm))
head(as.matrix(ssm, long = FALSE))

```

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.

Usage

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

Arguments

| | |
|---------|--|
| mu | Numeric vector. The long-term mean or equilibrium level (μ). |
| phi | Numeric matrix. The rate of mean reversion, determining how quickly the variable returns to its mean (Φ). |
| sigma | Numeric matrix. The matrix of volatility or randomness in the process (Σ). |
| delta_t | Numeric. Time interval (δ_t). |

Details

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

$$\beta = \exp(-\Phi \Delta_t)$$

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

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

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: [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [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 <- matrix(
  data = c(2.79, 0.06, 0.06, 3.27),
  nrow = p
)
delta_t <- 0.10

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

plot.simstatespace *Plot Method for an Object of Class simstatespace*

Description

Plot Method for an Object of Class simstatespace

Usage

```
## S3 method for class 'simstatespace'
plot(x, id = NULL, time = NULL, eta = FALSE, type = "b", ...)
```

Arguments

| | |
|------|---|
| x | Object of class simstatespace. |
| id | Numeric vector. Optional id numbers to plot. If id = NULL, plot all available data. |
| time | Numeric vector. Optional time points to plot. If time = NULL, plot all available data. |
| eta | Logical. If eta = TRUE, plot the latent variables. If eta = FALSE, plot the observed variables. |
| type | Character indicating the type of plotting; actually any of the types as in plot.default() . |
| ... | Additional arguments. |

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0 <- iden
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi <- iden
nu <- null_vec
lambda <- iden
theta <- 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 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  time = time,
  burn_in = burn_in
)

plot(ssm)

```

`print.simstatespace` *Print Method for an Object of Class simstatespace*

Description

Print Method for an Object of Class `simstatespace`

Usage

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

Arguments

`x` Object of Class `simstatespace`.
`...` Additional arguments.

Value

Prints simulated data in long format.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0 <- iden
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi <- iden
nu <- null_vec
lambda <- iden
theta <- 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 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  time = time,
  burn_in = burn_in
```

```
)
print(ssm)
```

SimSSM

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

Description

This function simulates data from a state space model.

Usage

```
SimSSM(
  mu0,
  sigma0,
  alpha,
  beta,
  psi,
  nu,
  lambda,
  theta,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  time,
  burn_in = 0
)
```

Arguments

| | |
|---------|--|
| mu0 | Numeric vector. Mean of initial latent variable values ($\mu_{\eta 0}$). |
| sigma0 | Numeric matrix. The covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| 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 | Numeric matrix. The process noise covariance matrix (Ψ). |
| nu | Numeric vector. Vector of intercepts for the measurement model (ν). |
| lambda | Numeric matrix. Factor loading matrix linking the latent variables to the observed variables (Λ). |
| theta | Numeric matrix. The measurement error covariance matrix (Θ). |
| gamma_y | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the observed variables at time t (Γ_y). |

| | |
|-----------|---|
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_η). |
| x | Numeric matrix. The matrix of observed covariates in <code>type = 1</code> or <code>type = 2</code> . The number of rows should be equal to <code>time + burn_in</code> . |
| type | Integer. State space model type. See Details for more information. |
| 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

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

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

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

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

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

The dynamic structure is given by

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

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

Type 2:

The measurement model is given by

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

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

The dynamic structure is given by

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

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length `n`. `data` is a list with the following elements:
 - `id`: A vector of ones of length `t`.
 - `time`: A vector of time points of length `t`.
 - `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.
- `fun`: Function used.

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](https://doi.org/10.1080/10705511003661553)

See Also

Other Simulation of State Space Models Data Functions: [OU2SSM\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#)

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
mu0 <- null_vec
sigma0 <- iden
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi <- iden
nu <- null_vec
lambda <- iden
theta <- diag(x = 0.50, nrow = k)
time <- 1000
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 = sigma0,  
  alpha = alpha,  
  beta = beta,  
  psi = psi,  
  nu = nu,  
  lambda = lambda,  
  theta = theta,  
  type = 0,  
  time = time,  
  burn_in = burn_in  
)  
  
plot(ssm)  
  
# Type 1  
ssm <- SimSSM(  
  mu0 = mu0,  
  sigma0 = sigma0,  
  alpha = alpha,  
  beta = beta,  
  psi = psi,  
  nu = nu,  
  lambda = lambda,  
  theta = theta,  
  gamma_eta = gamma_eta,  
  x = x,  
  type = 1,  
  time = time,  
  burn_in = burn_in  
)  
  
plot(ssm)  
  
# Type 2  
ssm <- SimSSM(  
  mu0 = mu0,  
  sigma0 = sigma0,  
  alpha = alpha,  
  beta = beta,  
  psi = psi,  
  nu = nu,  
  lambda = lambda,  
  theta = theta,  
  gamma_y = gamma_y,  
  gamma_eta = gamma_eta,  
  x = x,  
  type = 2,  
  time = time,  
  burn_in = burn_in  
)
```

```
plot(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,
  alpha,
  beta,
  psi,
  nu,
  lambda,
  theta,
  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 ($\mu_{\eta 0}$). |
| sigma0 | Numeric matrix. The covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| 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 | Numeric matrix. The process noise covariance matrix (Ψ). |
| nu | Numeric vector. Vector of intercepts for the measurement model (ν). |
| lambda | Numeric matrix. Factor loading matrix linking the latent variables to the observed variables (Λ). |
| theta | Numeric matrix. The measurement error covariance matrix (Θ). |

| | |
|-----------|---|
| gamma_y | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the observed variables at time t (Γ_y). |
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_η). |
| 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. See Details for more information. |
| 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

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

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

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

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

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

The dynamic structure is given by

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

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

Type 2:

The measurement model is given by

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

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

The dynamic structure is given by

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

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length `n`. Each element of data is a list with the following elements:
 - `id`: A vector of ID numbers of length `t`.
 - `time`: A vector time points of length `t`.
 - `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.
- `fun`: Function used.

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](https://doi.org/10.1080/10705511003661553)

See Also

Other Simulation of State Space Models Data Functions: [OU2SSM\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
set.seed(42)
k <- p <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0 <- iden
alpha <- null_vec
beta <- diag(x = 0.50, nrow = k)
psi <- iden
nu <- null_vec
lambda <- iden
theta <- 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 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  time = time,
  burn_in = burn_in
)

```

```
plot(ssm)
```

```

# Type 1
ssm <- SimSSMFixed(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  nu = nu,
  lambda = lambda,
  theta = theta,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  time = time,
  burn_in = burn_in
)

```

```
plot(ssm)
```

```

# Type 2
ssm <- SimSSMFixed(
  n = n,

```

```

    mu0 = mu0,
    sigma0 = sigma0,
    alpha = alpha,
    beta = beta,
    psi = psi,
    nu = nu,
    lambda = lambda,
    theta = theta,
    gamma_y = gamma_y,
    gamma_eta = gamma_eta,
    x = x,
    type = 2,
    time = time,
    burn_in = burn_in
)

plot(ssm)

```

SimSSMIVary

Simulate Data using a State Space Model Parameterization for $n > 1$ Individuals (Individual-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

```

SimSSMIVary(
  n,
  mu0,
  sigma0,
  alpha,
  beta,
  psi,
  nu,
  lambda,
  theta,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  time,
  burn_in = 0
)

```

Arguments

| | |
|-----------|---|
| n | Positive integer. Number of individuals. |
| mu0 | List of numeric vectors. Each element of the list is the mean of initial latent variable values ($\mu_{\eta 0}$). |
| sigma0 | List of numeric matrices. Each element of the list is the covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| alpha | List of numeric vectors. Each element of the list is the vector of intercepts for the dynamic model (α). |
| beta | List of numeric matrices. Each element of the list is the transition matrix relating the values of the latent variables at time $t - 1$ to those at time t (β). |
| psi | List of numeric matrices. Each element of the list is the process noise covariance matrix (Ψ). |
| nu | List of numeric vectors. Each element of the list is the vector of intercepts for the measurement model (ν). |
| lambda | List of numeric matrices. Each element of the list is the factor loading matrix linking the latent variables to the observed variables (Λ). |
| theta | List of numeric matrices. Each element of the list is the measurement error covariance matrix (Θ). |
| gamma_y | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the observed variables at time t (Γ_y). |
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_η). |
| 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. See Details for more information. |
| 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, alpha, beta, psi, nu, lambda, theta, gamma_y, or gamma_eta) is less than n , the function will cycle through the available values.

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length n . Each element of data is a list with the following elements:

- id: A vector of ID numbers of length t.
- time: A vector time points of length t.
- 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.
- fun: Function used.

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\(\)](#), [SimSSMFixed\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
# In this example, beta varies across individuals
set.seed(42)
k <- p <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- list(null_vec)
sigma0 <- list(iden)
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 <- list(iden)
nu <- list(null_vec)
lambda <- list(iden)
theta <- list(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 <- SimSSMIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  time = time,
  burn_in = burn_in
)

```

```
plot(ssm)
```

```

# Type 1
ssm <- SimSSMIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  nu = nu,
  lambda = lambda,
  theta = theta,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  time = time,
  burn_in = burn_in
)

```

```
plot(ssm)
```

```

# Type 2
ssm <- SimSSMIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,

```

```

    alpha = alpha,
    beta = beta,
    psi = psi,
    nu = nu,
    lambda = lambda,
    theta = theta,
    gamma_y = gamma_y,
    gamma_eta = gamma_eta,
    x = x,
    type = 2,
    time = time,
    burn_in = burn_in
)

plot(ssm)

```

SimSSMLinGrowth

Simulate Data from a Linear Growth Curve Model

Description

This function simulates data from a linear growth curve model for $n > 1$ individuals.

Usage

```

SimSSMLinGrowth(
  n,
  mu0,
  sigma0,
  theta,
  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 | Numeric matrix. The covariance matrix of the intercept and the slope. |
| theta | Numeric. 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 (Γ_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 (Γ_η). |
| 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 <code>time</code> . |
| type | Integer. State space model type. See Details for more information. |
| time | Positive integer. Number of time points to simulate. |

Details

Type 0:

The measurement model is given by

$$y_{i,t} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \eta_{0i,t} \\ \eta_{1i,t} \end{pmatrix} + \varepsilon_{i,t} \quad \text{with} \quad \varepsilon_{i,t} \sim \mathcal{N}(0, \theta)$$

where $y_{i,t}$, $\eta_{0i,t}$, $\eta_{1i,t}$, and $\varepsilon_{i,t}$ are random variables and θ is a model parameter. $y_{i,t}$ is a vector of observed random variables at time t and individual i , $\eta_{0i,t}$ and $\eta_{1i,t}$ form a vector of latent random variables at time t and individual i , and $\varepsilon_{i,t}$ is a vector of random measurement errors at time t and individual i . θ is the variance of ε .

The dynamic structure is given by

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

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

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

$$\Sigma_{\eta|0} = \begin{pmatrix} \sigma_{\eta_0}^2 & \sigma_{\eta_0, \eta_1} \\ \sigma_{\eta_1, \eta_0} & \sigma_{\eta_1}^2 \end{pmatrix}.$$

Type 1:

The measurement model is given by

$$y_{i,t} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \eta_{0i,t} \\ \eta_{1i,t} \end{pmatrix} + \varepsilon_{i,t} \quad \text{with} \quad \varepsilon_{i,t} \sim \mathcal{N}(0, \theta).$$

The dynamic structure is given by

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

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

Type 2:

The measurement model is given by

$$y_{i,t} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \eta_{0,i,t} \\ \eta_{1,i,t} \end{pmatrix} + \Gamma_y \mathbf{x}_{i,t} + \varepsilon_{i,t} \quad \text{with} \quad \varepsilon_{i,t} \sim \mathcal{N}(0, \theta)$$

where Γ_y is the coefficient matrix linking the covariates to the observed variables.

The dynamic structure is given by

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

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length `n`. Each element of `data` is a list with the following elements:
 - `id`: A vector of ID numbers of length `t`.
 - `time`: A vector time points of length `t`.
 - `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.
- `fun`: Function used.

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](https://doi.org/10.1080/10705511003661553)

See Also

Other Simulation of State Space Models Data Functions: [OU2SSM\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```

# prepare parameters
set.seed(42)
n <- 10
mu0 <- c(0.615, 1.006)
sigma0 <- matrix(
  data = c(
    1.932,
    0.618,
    0.618,
    0.587
  ),
  nrow = 2
)
theta <- 0.6
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 = sigma0,
  theta = theta,
  type = 0,
  time = time
)

plot(ssm)

# Type 1
ssm <- SimSSMLinGrowth(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  theta = theta,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  time = time
)

```

```

)

plot(ssm)

# Type 2
ssm <- SimSSMLinGrowth(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  theta = theta,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  type = 2,
  time = time
)

plot(ssm)

```

SimSSMLinGrowthIVary *Simulate Data from a Linear Growth Curve Model (Individual-Varying Parameters)*

Description

This function simulates data from a linear growth curve model for $n > 1$ individuals. In this model, the parameters can vary across individuals.

Usage

```

SimSSMLinGrowthIVary(
  n,
  mu0,
  sigma0,
  theta,
  gamma_y = NULL,
  gamma_eta = NULL,
  x = NULL,
  type = 0,
  time
)

```

Arguments

| | |
|-----|---|
| n | Positive integer. Number of individuals. |
| mu0 | A list of numeric vectors. Each element of the list is 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 | A list of numeric matrices. Each element of the list is the covariance matrix of the intercept and the slope. |
| theta | A list numeric values. Each element of the list is 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 (Γ_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 (Γ_η). |
| 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. See Details for more information. |
| time | Positive integer. Number of time points to simulate. |

Details

Parameters can vary across individuals by providing a list of parameter values. If the length of any of the parameters (μ_0 , σ_0 , μ , θ , γ_y , or γ_η) is less than n, the function will cycle through the available values.

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length n. Each element of data is a list with the following elements:
 - `id`: A vector of ID numbers of length t.
 - `time`: A vector time points of length t.
 - `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.
- `fun`: Function used.

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\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
# In this example, the mean vector of the intercept and slope vary.
# Specifically,
# there are two sets of values representing two latent classes.
set.seed(42)
n <- 10
mu0_1 <- c(0.615, 1.006) # lower starting point, higher growth
mu0_2 <- c(1.000, 0.500) # higher starting point, lower growth
mu0 <- list(mu0_1, mu0_2)
sigma0 <- list(
  matrix(
    data = c(
      1.932,
      0.618,
      0.618,
      0.587
    ),
    nrow = 2
  )
)
theta <- list(0.6)
time <- 10
gamma_y <- list(matrix(data = 0.10, nrow = 1, ncol = 2))
gamma_eta <- list(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 <- SimSSMLinGrowthIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  theta = theta,
  type = 0,
  time = time
)
```

```

plot(ssm)

# Type 1
ssm <- SimSSMLinGrowthIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  theta = theta,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  time = time
)

plot(ssm)

# Type 2
ssm <- SimSSMLinGrowthIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  theta = theta,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  type = 2,
  time = time
)

plot(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.

Usage

```

SimSSMOU(
  mu0,
  sigma0,
  mu,
  phi,
  sigma,

```

```

    nu,
    lambda,
    theta,
    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 ($\mu_{\eta 0}$). |
| sigma0 | Numeric matrix. The covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| mu | Numeric vector. The long-term mean or equilibrium level (μ). |
| phi | Numeric matrix. The rate of mean reversion, determining how quickly the variable returns to its mean (Φ). |
| sigma | Numeric matrix. The matrix of volatility or randomness in the process (Σ). |
| nu | Numeric vector. Vector of intercepts for the measurement model (ν). |
| lambda | Numeric matrix. Factor loading matrix linking the latent variables to the observed variables (Λ). |
| theta | Numeric matrix. The measurement error covariance matrix (Θ). |
| gamma_y | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the observed variables at time t (Γ_y). |
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_η). |
| 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. See Details for more information. |
| 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

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

where \mathbf{y}_t , $\boldsymbol{\eta}_t$, and ε_t are random variables and ν , Λ , and Θ are model parameters. \mathbf{y}_t is a vector of observed random variables, $\boldsymbol{\eta}_t$ is a vector of latent random variables, and ε_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

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

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 $d\mathbf{W}$ is a Wiener process or Brownian motion, which represents random fluctuations.

Type 1:

The measurement model is given by

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

The dynamic structure is given by

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

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

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

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

The dynamic structure is given by

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

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length `n`. `data` is a list with the following elements:
 - `id`: A vector of ones of length `t`.
 - `time`: A vector of time points of length `t`.
 - `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.
- `fun`: Function used.

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\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
set.seed(42)
p <- k <- 2
iden <- diag(p)
mu0 <- c(-3.0, 1.5)
sigma0 <- iden
mu <- c(5.76, 5.18)
phi <- matrix(data = c(0.10, -0.05, -0.05, 0.10), nrow = p)
sigma <- matrix(
  data = c(2.79, 0.06, 0.06, 3.27),
  nrow = p
)
nu <- rep(x = 0, times = k)
lambda <- diag(k)
theta <- diag(x = 0.50, nrow = k)
delta_t <- 0.10
time <- 1000
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 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  delta_t = delta_t,
```



```

    time = time,
    burn_in = burn_in
  )

plot(ssm)

# Type 1
ssm <- SimSSMOU(
  mu0 = mu0,
  sigma0 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,
  theta = theta,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

plot(ssm)

# Type 2
ssm <- SimSSMOU(
  mu0 = mu0,
  sigma0 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,
  theta = theta,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  type = 2,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

plot(ssm)

```

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.

Usage

```
SimSSMOUFixed(
  n,
  mu0,
  sigma0,
  mu,
  phi,
  sigma,
  nu,
  lambda,
  theta,
  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 ($\mu_{\eta 0}$). |
| sigma0 | Numeric matrix. The covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| mu | Numeric vector. The long-term mean or equilibrium level (μ). |
| phi | Numeric matrix. The rate of mean reversion, determining how quickly the variable returns to its mean (Φ). |
| sigma | Numeric matrix. The matrix of volatility or randomness in the process (Σ). |
| nu | Numeric vector. Vector of intercepts for the measurement model (ν). |
| lambda | Numeric matrix. Factor loading matrix linking the latent variables to the observed variables (Λ). |
| theta | Numeric matrix. The measurement error covariance matrix (Θ). |
| gamma_y | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the observed variables at time t (Γ_y). |
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_η). |
| x | Numeric matrix. The matrix of observed covariates in <code>type = 1</code> or <code>type = 2</code> . The number of rows should be equal to <code>time + burn_in</code> . |
| type | Integer. State space model type. See Details for more information. |

| | |
|---------|---|
| 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

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

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

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

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 $d\mathbf{W}$ is a Wiener process or Brownian motion, which represents random fluctuations.

Type 1:

The measurement model is given by

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

The dynamic structure is given by

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

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

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

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

The dynamic structure is given by

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

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length `n`. Each element of data is a list with the following elements:
 - `id`: A vector of ID numbers of length `t`.
 - `time`: A vector time points of length `t`.
 - `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.
- `fun`: Function used.

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](https://doi.org/10.1103/physrev.36.823)

See Also

Other Simulation of State Space Models Data Functions: [OU2SSM\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
set.seed(42)
p <- k <- 2
iden <- diag(p)
n <- 5
mu0 <- c(-3.0, 1.5)
sigma0 <- iden
mu <- c(5.76, 5.18)
phi <- matrix(data = c(0.10, -0.05, -0.05, 0.10), nrow = p)
sigma <- matrix(
  data = c(2.79, 0.06, 0.06, 3.27),
  nrow = p
)
nu <- rep(x = 0, times = k)
```

```

lambda <- diag(k)
theta <- 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 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

plot(ssm)

# Type 1
ssm <- SimSSMOUFixed(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,
  theta = theta,
  gamma_eta = gamma_eta,
  x = x,
  type = 1,
  delta_t = delta_t,
  time = time,

```

```

    burn_in = burn_in
  )

plot(ssm)

# Type 2
ssm <- SimSSMOUFixed(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,
  theta = theta,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  type = 2,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

plot(ssm)

```

SimSSMOUIVary

Simulate Data from an Ornstein–Uhlenbeck Model using a State Space Model Parameterization for $n > 1$ Individuals (Individual-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

```

SimSSMOUIVary(
  n,
  mu0,
  sigma0,
  mu,
  phi,
  sigma,
  nu,
  lambda,

```

```

    theta,
    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 ($\mu_{\eta 0}$). |
| sigma0 | Numeric matrix. The covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| mu | List of numeric vectors. Each element of the list is the long-term mean or equilibrium level (μ). |
| phi | List of numeric matrices. Each element of the list is the rate of mean reversion, determining how quickly the variable returns to its mean (Φ). |
| sigma | List of numeric matrices. Each element of the list is the matrix of volatility or randomness in the process (Σ). |
| nu | Numeric vector. Vector of intercepts for the measurement model (ν). |
| lambda | Numeric matrix. Factor loading matrix linking the latent variables to the observed variables (Λ). |
| theta | Numeric matrix. The measurement error covariance matrix (Θ). |
| gamma_y | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the observed variables at time t (Γ_y). |
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_{η}). |
| 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. See Details for more information. |
| 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, mu, phi, sigma, nu, lambda, theta, gamma_y, or gamma_eta) is less than n, the function will cycle through the available values.

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length `n`. Each element of data is a list with the following elements:
 - `id`: A vector of ID numbers of length `t`.
 - `time`: A vector time points of length `t`.
 - `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.
- `fun`: Function used.

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](https://doi.org/10.1103/physrev.36.823)

See Also

Other Simulation of State Space Models Data Functions: [OU2SSM\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
# In this example, phi varies across individuals
set.seed(42)
p <- k <- 2
iden <- diag(p)
n <- 5
mu0 <- list(c(-3.0, 1.5))
sigma0 <- list(iden)
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 <- list(
    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 <- list(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 <- SimSSMOUIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,
  theta = theta,
  type = 0,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

```

```
plot(ssm)
```

```

# Type 1
ssm <- SimSSMOUIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,

```

```

    theta = theta,
    gamma_eta = gamma_eta,
    x = x,
    type = 1,
    delta_t = delta_t,
    time = time,
    burn_in = burn_in
)

plot(ssm)

# Type 2
ssm <- SimSSMOUIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  mu = mu,
  phi = phi,
  sigma = sigma,
  nu = nu,
  lambda = lambda,
  theta = theta,
  gamma_y = gamma_y,
  gamma_eta = gamma_eta,
  x = x,
  type = 2,
  delta_t = delta_t,
  time = time,
  burn_in = burn_in
)

plot(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.

Usage

```

SimSSMVAR(
  mu0,
  sigma0,
  alpha,
  beta,

```

```

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

```

Arguments

| | |
|-----------|---|
| mu0 | Numeric vector. Mean of initial latent variable values ($\mu_{\eta 0}$). |
| sigma0 | Numeric matrix. The covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| 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 | Numeric matrix. The process noise covariance matrix (Ψ). |
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_{η}). |
| 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

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

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

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

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

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length `n`. `data` is a list with the following elements:
 - `id`: A vector of ones of length `t`.
 - `time`: A vector of time points of length `t`.
 - `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.
- `fun`: Function used.

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](https://doi.org/10.1080/10705511003661553)

See Also

Other Simulation of State Space Models Data Functions: [OU2SSM\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
set.seed(42)
k <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
mu0 <- null_vec
sigma0 <- iden
alpha <- null_vec
beta <- diag(x = 0.5, nrow = k)
psi <- iden
time <- 1000
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 = sigma0,
```

```

    alpha = alpha,
    beta = beta,
    psi = psi,
    time = time,
    burn_in = burn_in
  )

plot(ssm)

# With covariates
ssm <- SimSSMVAR(
  mu0 = mu0,
  sigma0 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  gamma_eta = gamma_eta,
  x = x,
  time = time,
  burn_in = burn_in
)

plot(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,
  alpha,
  beta,
  psi,
  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 ($\mu_{\eta 0}$). |
| sigma0 | Numeric matrix. The covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| 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 | Numeric matrix. The process noise covariance matrix (Ψ). |
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_{η}). |
| 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 <code>time + burn_in</code> . |
| 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 an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length n . Each element of `data` is a list with the following elements:
 - `id`: A vector of ID numbers of length t .
 - `time`: A vector time points of length t .
 - `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.
- `fun`: Function used.

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\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARIVary\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
set.seed(42)
k <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- null_vec
sigma0 <- iden
alpha <- null_vec
beta <- diag(x = 0.5, nrow = k)
psi <- iden
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 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  time = time,
  burn_in = burn_in
)

plot(ssm)

# With covariates
ssm <- SimSSMVARFixed(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
```

```

    alpha = alpha,
    beta = beta,
    psi = psi,
    gamma_eta = gamma_eta,
    x = x,
    time = time,
    burn_in = burn_in
)

plot(ssm)

```

| | |
|----------------|--|
| SimSSMVARIVary | <i>Simulate Data from a Vector Autoregressive Model using a State Space Model Parameterization for $n > 1$ Individuals (Individual-Varying Parameters)</i> |
|----------------|--|

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

```

SimSSMVARIVary(
  n,
  mu0,
  sigma0,
  alpha,
  beta,
  psi,
  gamma_eta = NULL,
  x = NULL,
  time = 0,
  burn_in = 0
)

```

Arguments

| | |
|--------|--|
| n | Positive integer. Number of individuals. |
| mu0 | List of numeric vectors. Each element of the list is the mean of initial latent variable values ($\mu_{\eta 0}$). |
| sigma0 | List of numeric matrices. Each element of the list is the covariance matrix of initial latent variable values ($\Sigma_{\eta 0}$). |
| alpha | List of numeric vectors. Each element of the list is the vector of intercepts for the dynamic model (α). |

| | |
|-----------|---|
| beta | List of numeric matrices. Each element of the list is the transition matrix relating the values of the latent variables at time $t - 1$ to those at time t (β). |
| psi | List of numeric matrices. Each element of the list is the process noise covariance matrix (Ψ). |
| gamma_eta | Numeric matrix. Matrix relating the values of the covariate matrix at time t to the latent variables at time t (Γ_{η}). |
| 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 (μ_0 , σ_0 , α , β , ψ , or γ_{η}) is less than n , the function will cycle through the available values.

Value

Returns an object of class `simstatespace` which is a list with the following elements:

- `call`: Function call.
- `args`: Function arguments.
- `data`: Generated data which is a list of length n . Each element of data is a list with the following elements:
 - `id`: A vector of ID numbers of length t .
 - `time`: A vector time points of length t .
 - `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.
- `fun`: Function used.

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\(\)](#), [SimSSMFixed\(\)](#), [SimSSMIVary\(\)](#), [SimSSMLinGrowthIVary\(\)](#), [SimSSMLinGrowth\(\)](#), [SimSSMOUFixed\(\)](#), [SimSSMOUIVary\(\)](#), [SimSSMOU\(\)](#), [SimSSMVARFixed\(\)](#), [SimSSMVAR\(\)](#), [SimSSM\(\)](#)

Examples

```
# prepare parameters
# In this example, beta varies across individuals
set.seed(42)
k <- 3
iden <- diag(k)
null_vec <- rep(x = 0, times = k)
n <- 5
mu0 <- list(null_vec)
sigma0 <- list(iden)
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 <- list(iden)
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 <- SimSSMVARIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  time = time,
  burn_in = burn_in
)
```

```
plot(ssm)

# With covariates
ssm <- SimSSMVARIVary(
  n = n,
  mu0 = mu0,
  sigma0 = sigma0,
  alpha = alpha,
  beta = beta,
  psi = psi,
  gamma_eta = gamma_eta,
  x = x,
  time = time,
  burn_in = burn_in
)

plot(ssm)
```

Index

* Simulation of State Space Models Data

Functions

OU2SSM, [5](#)
SimSSM, [10](#)
SimSSMFixed, [14](#)
SimSSMIVary, [18](#)
SimSSMLinGrowth, [22](#)
SimSSMLinGrowthIVary, [26](#)
SimSSMOU, [29](#)
SimSSMOUFixed, [33](#)
SimSSMOUIVary, [38](#)
SimSSMVAR, [42](#)
SimSSMVARFixed, [45](#)
SimSSMVARIVary, [48](#)

* growth

SimSSMLinGrowth, [22](#)
SimSSMLinGrowthIVary, [26](#)

* methods

as.data.frame.simstatespace, [2](#)
as.matrix.simstatespace, [4](#)
plot.simstatespace, [7](#)
print.simstatespace, [8](#)

* ou

OU2SSM, [5](#)
SimSSMOU, [29](#)
SimSSMOUFixed, [33](#)
SimSSMOUIVary, [38](#)

* simStateSpace

OU2SSM, [5](#)
SimSSM, [10](#)
SimSSMFixed, [14](#)
SimSSMIVary, [18](#)
SimSSMLinGrowth, [22](#)
SimSSMLinGrowthIVary, [26](#)
SimSSMOU, [29](#)
SimSSMOUFixed, [33](#)
SimSSMOUIVary, [38](#)
SimSSMVAR, [42](#)
SimSSMVARFixed, [45](#)

SimSSMVARIVary, [48](#)

* sim

OU2SSM, [5](#)
SimSSM, [10](#)
SimSSMFixed, [14](#)
SimSSMIVary, [18](#)
SimSSMLinGrowth, [22](#)
SimSSMLinGrowthIVary, [26](#)
SimSSMOU, [29](#)
SimSSMOUFixed, [33](#)
SimSSMOUIVary, [38](#)
SimSSMVAR, [42](#)
SimSSMVARFixed, [45](#)
SimSSMVARIVary, [48](#)

* ssm

SimSSM, [10](#)
SimSSMFixed, [14](#)
SimSSMIVary, [18](#)

* var

SimSSMVAR, [42](#)
SimSSMVARFixed, [45](#)
SimSSMVARIVary, [48](#)

as.data.frame.simstatespace, [2](#)
as.matrix.simstatespace, [4](#)

OU2SSM, [5](#), [12](#), [16](#), [20](#), [24](#), [28](#), [32](#), [36](#), [40](#), [44](#),
[47](#), [50](#)

plot.default(), [7](#)
plot.simstatespace, [7](#)
print.simstatespace, [8](#)

SimSSM, [6](#), [10](#), [16](#), [20](#), [24](#), [28](#), [32](#), [36](#), [40](#), [44](#),
[47](#), [50](#)
SimSSMFixed, [6](#), [12](#), [14](#), [20](#), [24](#), [28](#), [32](#), [36](#), [40](#),
[44](#), [47](#), [50](#)
SimSSMIVary, [6](#), [12](#), [16](#), [18](#), [24](#), [28](#), [32](#), [36](#), [40](#),
[44](#), [47](#), [50](#)
SimSSMLinGrowth, [6](#), [12](#), [16](#), [20](#), [22](#), [28](#), [32](#),
[36](#), [40](#), [44](#), [47](#), [50](#)

SimSSMLinGrowthIVary, [6](#), [12](#), [16](#), [20](#), [24](#), [26](#),
[32](#), [36](#), [40](#), [44](#), [47](#), [50](#)
SimSSMOU, [6](#), [12](#), [16](#), [20](#), [24](#), [28](#), [29](#), [36](#), [40](#), [44](#),
[47](#), [50](#)
SimSSMOUFixed, [6](#), [12](#), [16](#), [20](#), [24](#), [28](#), [32](#), [33](#),
[40](#), [44](#), [47](#), [50](#)
SimSSMOUIVary, [6](#), [12](#), [16](#), [20](#), [24](#), [28](#), [32](#), [36](#),
[38](#), [44](#), [47](#), [50](#)
SimSSMVAR, [6](#), [12](#), [16](#), [20](#), [24](#), [28](#), [32](#), [36](#), [40](#),
[42](#), [47](#), [50](#)
SimSSMVARFixed, [6](#), [12](#), [16](#), [20](#), [24](#), [28](#), [32](#), [36](#),
[40](#), [44](#), [45](#), [50](#)
SimSSMVARIVary, [6](#), [12](#), [16](#), [20](#), [24](#), [28](#), [32](#), [36](#),
[40](#), [44](#), [47](#), [48](#)