Package ‘fitCTVARMx’

July 7, 2024

Title Fit the Continuous-Time Vector Autoregressive Model

Version 0.0.0.9000

Description Fit the continuous-time vector autoregressive model using the 'OpenMx' package.

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

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

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Depends R (>= 3.0.0), OpenMx

Imports stats

Suggests knitr, rmarkdown, testthat, simStateSpace

RoxygenNote 7.3.2

NeedsCompilation no

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

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

Contents

`coef.fitctvaridmx` *Parameter Estimates*

Description

Parameter Estimates

Usage

```
## S3 method for class 'fitctvaridmx'
coef(object, sigma = FALSE, theta = FALSE, ...)
```

Arguments

`object`	Object of class `fitctvaridmx`.
`sigma`	Logical. If `sigma = TRUE`, include estimates of the `sigma` matrix, if available. If `sigma = FALSE`, exclude estimates of the `sigma` matrix.
`theta`	Logical. If `theta = TRUE`, include estimates of the `theta` matrix, if available. If `theta = FALSE`, exclude estimates of the `theta` matrix.
`...`	additional arguments.

Value

Returns a list of vectors of parameter estimates.

Author(s)

Ivan Jacob Agaloos Pesigan

`coef.fitctvarmx` *Parameter Estimates*

Description

Parameter Estimates

Usage

```
## S3 method for class 'fitctvarmx'
coef(object, sigma = FALSE, theta = FALSE, ...)
```

Arguments

`object`	Object of class `fitctvarmx`.
`sigma`	Logical. If `sigma = TRUE`, include estimates of the `sigma` matrix, if available. If `sigma = FALSE`, exclude estimates of the `sigma` matrix.
`theta`	Logical. If `theta = TRUE`, include estimates of the `theta` matrix, if available. If `theta = FALSE`, exclude estimates of the `theta` matrix.
`...`	additional arguments.

Value

Returns a vector of parameter estimates.

Author(s)

Ivan Jacob Agaloos Pesigan

`FitCTVARIDMx` *Fit the First Order Continuous-Time Vector Autoregressive Model by ID*

Description

Fit the First Order Continuous-Time Vector Autoregressive Model by ID

Usage

```
FitCTVARIDMx(
  data,
  observed,
  id,
  time,
  phi_start = NULL,
  phi_lbound = NULL,
  phi_ubound = NULL,
  sigma_diag = TRUE,
  sigma_start = NULL,
  sigma_lbound = NULL,
  sigma_ubound = NULL,
  theta_fixed = TRUE,
  theta_start = NULL,
  theta_lbound = NULL,
  theta_ubound = NULL,
  mu0_fixed = TRUE,
  mu0_start = NULL,
  mu0_lbound = NULL,
  mu0_ubound = NULL,
```

```
    sigma0_fixed = TRUE,
    sigma0_diag = TRUE,
    sigma0_start = NULL,
    sigma0_lbound = NULL,
    sigma0_ubound = NULL,
    try = 1000,
    ncores = NULL
)
```

Arguments

data	Data frame. A data frame object of data for potentially multiple subjects that contain a column of subject ID numbers (i.e., an ID variable), and at least one column of observed values.
observed	Character vector. A vector of character strings of the names of the observed variables in the data.
id	Character string. A character string of the name of the ID variable in the data.
time	Character string. A character string of the name of the TIME variable in the data.
phi_start	Numeric matrix. Optional starting values for `phi`.
phi_lbound	Numeric matrix. Optional lower bound for `phi`.
phi_ubound	Numeric matrix. Optional upper bound for `phi`.
sigma_diag	Logical. If `sigma_diag = TRUE`, `sigma` is a diagonal matrix.
sigma_start	Numeric matrix. Optional starting values for `sigma`.
sigma_lbound	Numeric matrix. Optional lower bound for `sigma`.
sigma_ubound	Optional upper bound for `sigma`.
theta_fixed	Logical. If `theta_fixed = TRUE`, the measurement error matrix `theta` is fixed to zero. If `theta_fixed = FALSE`, estimate the diagonal measurement error matrix `theta`.
theta_start	Optional starting values for `theta`. Ignored if `theta_fixed = TRUE`.
theta_lbound	Optional lower bound for `theta`. Ignored if `theta_fixed = TRUE`.
theta_ubound	Optional upper bound for `theta`. Ignored if `theta_fixed = TRUE`.
mu0_fixed	Logical. If `mu0_fixed = TRUE`, initial mean vector `mu0` is fixed. If `mu0_fixed = FALSE`, initial mean vector `mu0` is estimated.
mu0_start	Optional starting values for `mu0`. If `mu0_fixed = TRUE`, `mu0_start` will be used as fixed values. If `mu0_fixed = FALSE`, `mu0_start` will be used as starting values.
mu0_lbound	Optional lower bound for `mu0`. Ignored if `mu0_fixed = TRUE`.
mu0_ubound	Optional upper bound for `mu0`. Ignored if `mu0_fixed = TRUE`.
sigma0_fixed	Logical. If `sigma0_fixed = TRUE`, initial mean vector `sigma0` is fixed. If `sigma0_fixed = FALSE`, initial mean vector `sigma0` is estimated.
sigma0_diag	Logical. If `sigma0_diag = TRUE`, `sigma0` is a diagonal matrix.

`sigma0_start`	Optional starting values for `sigma0`. If `sigma0_fixed = TRUE`, `sigma0_start` will be used as fixed values. If `sigma0_fixed = FALSE`, `sigma0_start` will be used as starting values.
`sigma0_lbound`	Optional lower bound for `sigma0`. Ignored if `sigma0_fixed = TRUE`.
`sigma0_ubound`	Optional upper bound for `sigma0`. Ignored if `sigma0_fixed = TRUE`.
`try`	Positive integer. Number of extra tries for `OpenMx::mxTryHard()`.
`ncores`	Positive integer. Number of cores to use.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other CTVAR Functions: `FitCTVARMx()`

Examples

```
## Not run:
# Generate data using the simStateSpace package------------------------------
set.seed(42)
phi_mu <- matrix(
  data = c(
    -0.357, 0.771, -0.450,
    0.0, -0.511, 0.729,
    0, 0, -0.693
  ),
  nrow = 3
)
phi_sigma <- diag(3 * 3)
phi <- simStateSpace::SimPhiN(
  n = 5,
  phi = phi_mu,
  vcov_phi_vec_l = t(chol(phi_sigma))
)
sim <- simStateSpace::SimSSMOUIVary(
  n = 5,
  time = 100,
  delta_t = 0.10,
  mu0 = list(rep(x = 0, times = 3)),
  sigma0_l = list(t(chol(diag(3)))),
  mu = list(rep(x = 0, times = 3)),
  phi = phi,
  sigma_l = list(t(chol(diag(3)))),
  nu = list(rep(x = 0, times = 3)),
  lambda = list(diag(3)),
  theta_l = list(matrix(data = 0, nrow = 3, ncol = 3))
)
data <- as.data.frame(sim)

# Fit the model--------------------------------------------------------------
```

```
library(fitCTVARMx)
fit <- FitCTVARIDMx(
  data = data,
  observed = c("y1", "y2", "y3"),
  id = "id"
)
print(fit)
summary(fit)
coef(fit)
vcov(fit)

## End(Not run)
```

FitCTVARMx *Fit the First-Order Continuous-Time Vector Autoregressive Model*

Description

Fit the First-Order Continuous-Time Vector Autoregressive Model

Usage

```
FitCTVARMx(
  data,
  observed,
  id,
  time,
  phi_start = NULL,
  phi_lbound = NULL,
  phi_ubound = NULL,
  sigma_diag = TRUE,
  sigma_start = NULL,
  sigma_lbound = NULL,
  sigma_ubound = NULL,
  theta_fixed = TRUE,
  theta_start = NULL,
  theta_lbound = NULL,
  theta_ubound = NULL,
  mu0_fixed = TRUE,
  mu0_start = NULL,
  mu0_lbound = NULL,
  mu0_ubound = NULL,
  sigma0_fixed = TRUE,
  sigma0_diag = TRUE,
  sigma0_start = NULL,
  sigma0_lbound = NULL,
  sigma0_ubound = NULL,
```

```
  try = 1000,
  ncores = NULL
)
```

Arguments

data	Data frame. A data frame object of data for potentially multiple subjects that contain a column of subject ID numbers (i.e., an ID variable), and at least one column of observed values.
observed	Character vector. A vector of character strings of the names of the observed variables in the data.
id	Character string. A character string of the name of the ID variable in the data.
time	Character string. A character string of the name of the TIME variable in the data.
phi_start	Numeric matrix. Optional starting values for `phi`.
phi_lbound	Numeric matrix. Optional lower bound for `phi`.
phi_ubound	Numeric matrix. Optional upper bound for `phi`.
sigma_diag	Logical. If `sigma_diag = TRUE`, `sigma` is a diagonal matrix.
sigma_start	Numeric matrix. Optional starting values for `sigma`.
sigma_lbound	Numeric matrix. Optional lower bound for `sigma`.
sigma_ubound	Optional upper bound for `sigma`.
theta_fixed	Logical. If `theta_fixed = TRUE`, the measurement error matrix `theta` is fixed to zero. If `theta_fixed = FALSE`, estimate the diagonal measurement error matrix `theta`.
theta_start	Optional starting values for `theta`. Ignored if `theta_fixed = TRUE`.
theta_lbound	Optional lower bound for `theta`. Ignored if `theta_fixed = TRUE`.
theta_ubound	Optional upper bound for `theta`. Ignored if `theta_fixed = TRUE`.
mu0_fixed	Logical. If `mu0_fixed = TRUE`, initial mean vector `mu0` is fixed. If `mu0_fixed = FALSE`, initial mean vector `mu0` is estimated.
mu0_start	Optional starting values for `mu0`. If `mu0_fixed = TRUE`, `mu0_start` will be used as fixed values. If `mu0_fixed = FALSE`, `mu0_start` will be used as starting values.
mu0_lbound	Optional lower bound for `mu0`. Ignored if `mu0_fixed = TRUE`.
mu0_ubound	Optional upper bound for `mu0`. Ignored if `mu0_fixed = TRUE`.
sigma0_fixed	Logical. If `sigma0_fixed = TRUE`, initial mean vector `sigma0` is fixed. If `sigma0_fixed = FALSE`, initial mean vector `sigma0` is estimated.
sigma0_diag	Logical. If `sigma0_diag = TRUE`, `sigma0` is a diagonal matrix.
sigma0_start	Optional starting values for `sigma0`. If `sigma0_fixed = TRUE`, `sigma0_start` will be used as fixed values. If `sigma0_fixed = FALSE`, `sigma0_start` will be used as starting values.
sigma0_lbound	Optional lower bound for `sigma0`. Ignored if `sigma0_fixed = TRUE`.
sigma0_ubound	Optional upper bound for `sigma0`. Ignored if `sigma0_fixed = TRUE`.
try	Positive integer. Number of extra tries for `OpenMx::mxTryHard()`.
ncores	Positive integer. Number of cores to use.

Value

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

call Function call.

args List of function arguments.

fun Function used ("FitCTVARMx").

output A fitted OpenMx model.

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other CTVAR Functions: `FitCTVARIDMx()`

Examples

```
## Not run:
# Generate data using the simStateSpace package------------------------------
set.seed(42)
sim <- simStateSpace::SimSSMOUFixed(
  n = 5,
  time = 100,
  delta_t = 0.10,
  mu0 = rep(x = 0, times = 3),
  sigma0_l = t(chol(diag(3))),
  mu = rep(x = 0, times = 3),
  phi = matrix(
    data = c(
      -0.357, 0.771, -0.450,
      0.0, -0.511, 0.729,
      0, 0, -0.693
    ),
    nrow = 3
  ),
  sigma_l = t(chol(diag(3))),
  nu = rep(x = 0, times = 3),
  lambda = diag(3),
  theta_l = matrix(data = 0, nrow = 3, ncol = 3)
)
data <- as.data.frame(sim)

# Fit the model--------------------------------------------------------------
library(fitCTVARMx)
fit <- FitCTVARMx(
  data = data,
  observed = c("y1", "y2", "y3"),
  id = "id"
)
print(fit)
```

```
summary(fit)
coef(fit)
vcov(fit)

## End(Not run)
```

`print.fitctvaridmx` *Print Method for Object of Class* `fitctvaridmx`

Description

Print Method for Object of Class `fitctvaridmx`

Usage

```
## S3 method for class 'fitctvaridmx'
print(x, means = TRUE, ...)
```

Arguments

`x`	an object of class `fitctvaridmx`.
`means`	Logical. If `means = TRUE`, return means. Otherwise, the function returns raw estimates.
`...`	further arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

`print.fitctvarmx` *Print Method for Object of Class* `fitctvarmx`

Description

Print Method for Object of Class `fitctvarmx`

Usage

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

Arguments

`x`	an object of class `fitctvarmx`.
`...`	further arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

---

summary.fitctvaridmx *Summary Method for Object of Class* `fitctvaridmx`

Description

Summary Method for Object of Class `fitctvaridmx`

Usage

```
## S3 method for class 'fitctvaridmx'
summary(object, means = TRUE, ...)
```

Arguments

`object`	an object of class `fitctvaridmx`.
`means`	Logical. If `means = TRUE`, return means. Otherwise, the function returns raw estimates.
`...`	further arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

---

summary.fitctvarmx *Summary Method for Object of Class* `fitctvarmx`

Description

Summary Method for Object of Class `fitctvarmx`

Usage

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

Arguments

`object`	an object of class `fitctvarmx`.
`...`	further arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

vcov.fitctvaridmx *Sampling Covariance Matrix of the Parameter Estimates*

Description

Sampling Covariance Matrix of the Parameter Estimates

Usage

```
## S3 method for class 'fitctvaridmx'
vcov(object, sigma = FALSE, theta = FALSE, ...)
```

Arguments

object	Object of class fitctvaridmx.
sigma	Logical. If sigma = TRUE, include estimates of the sigma matrix, if available. If sigma = FALSE, exclude estimates of the sigma matrix.
theta	Logical. If theta = TRUE, include estimates of the theta matrix, if available. If theta = FALSE, exclude estimates of the theta matrix.
...	additional arguments.

Value

Returns a list of sampling variance-covariance matrices.

Author(s)

Ivan Jacob Agaloos Pesigan

vcov.fitctvarmx *Sampling Covariance Matrix of the Parameter Estimates*

Description

Sampling Covariance Matrix of the Parameter Estimates

Usage

```
## S3 method for class 'fitctvarmx'
vcov(object, sigma = FALSE, theta = FALSE, ...)
```

Arguments

`object`	Object of class `fitctvarmx`.
`sigma`	Logical. If `sigma = TRUE`, include estimates of the `sigma` matrix, if available. If `sigma = FALSE`, exclude estimates of the `sigma` matrix.
`theta`	Logical. If `theta = TRUE`, include estimates of the `theta` matrix, if available. If `theta = FALSE`, exclude estimates of the `theta` matrix.
`...`	additional arguments.

Value

Returns a list of sampling variance-covariance matrices.

Author(s)

Ivan Jacob Agaloos Pesigan

Index