

# Package ‘fitDTVARMx’

July 8, 2024

**Title** Fit the Discrete-Time Vector Autoregressive Model

**Version** 0.0.0.9000

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

**URL** <https://github.com/jeksterslab/fitDTVARMx>,  
<https://jeksterslab.github.io/fitDTVARMx/>

**BugReports** <https://github.com/jeksterslab/fitDTVARMx/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.fittedvaridmx . . . . .	2
coef.fittedvarmx . . . . .	2
FitDTVARIDMx . . . . .	3
FitDTVARMx . . . . .	6
print.fittedvaridmx . . . . .	9
print.fittedvarmx . . . . .	10
summary.fittedvaridmx . . . . .	10
summary.fittedvarmx . . . . .	11
vcov.fittedvaridmx . . . . .	11
vcov.fittedvarmx . . . . .	12

---

coef.fitdtvaridmx	<i>Parameter Estimates</i>
-------------------	----------------------------

---

**Description**

Parameter Estimates

**Usage**

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

**Arguments**

- object            Object of class fitdtvaridmx.
- psi               Logical. If psi = TRUE, include estimates of the psi matrix, if available. If psi = FALSE, exclude estimates of the psi 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.fitdtvarmx	<i>Parameter Estimates</i>
-----------------	----------------------------

---

**Description**

Parameter Estimates

**Usage**

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

**Arguments**

object	Object of class fitdtdvarmx.
psi	Logical. If psi = TRUE, include estimates of the psi matrix, if available. If psi = FALSE, exclude estimates of the psi 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

---

FitDTVARIDMx

---

*Fit the First-Order Discrete-Time Vector Autoregressive Model by ID*


---

**Description**

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

**Usage**

```
FitDTVARIDMx(
  data,
  observed,
  id,
  beta_start = NULL,
  beta_lbound = NULL,
  beta_ubound = NULL,
  psi_diag = TRUE,
  psi_start = NULL,
  psi_lbound = NULL,
  psi_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

<code>data</code>	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.
<code>observed</code>	Character vector. A vector of character strings of the names of the observed variables in the data.
<code>id</code>	Character string. A character string of the name of the ID variable in the data.
<code>beta_start</code>	Numeric matrix. Optional starting values for beta.
<code>beta_lbound</code>	Numeric matrix. Optional lower bound for beta.
<code>beta_ubound</code>	Numeric matrix. Optional upper bound for beta.
<code>psi_diag</code>	Logical. If <code>psi_diag = TRUE</code> , <code>psi</code> is a diagonal matrix.
<code>psi_start</code>	Numeric matrix. Optional starting values for psi.
<code>psi_lbound</code>	Numeric matrix. Optional lower bound for psi.
<code>psi_ubound</code>	Optional upper bound for psi.
<code>theta_fixed</code>	Logical. If <code>theta_fixed = TRUE</code> , the measurement error matrix <code>theta</code> is fixed to zero. If <code>theta_fixed = FALSE</code> , estimate the diagonal measurement error matrix <code>theta</code> .
<code>theta_start</code>	Optional starting values for <code>theta</code> . Ignored if <code>theta_fixed = TRUE</code> .
<code>theta_lbound</code>	Optional lower bound for <code>theta</code> . Ignored if <code>theta_fixed = TRUE</code> .
<code>theta_ubound</code>	Optional upper bound for <code>theta</code> . Ignored if <code>theta_fixed = TRUE</code> .
<code>mu0_fixed</code>	Logical. If <code>mu0_fixed = TRUE</code> , initial mean vector <code>mu0</code> is fixed. If <code>mu0_fixed = FALSE</code> , initial mean vector <code>mu0</code> is estimated.
<code>mu0_start</code>	Optional starting values for <code>mu0</code> . If <code>mu0_fixed = TRUE</code> , <code>mu0_start</code> will be used as fixed values. If <code>mu0_fixed = FALSE</code> , <code>mu0_start</code> will be used as starting values.
<code>mu0_lbound</code>	Optional lower bound for <code>mu0</code> . Ignored if <code>mu0_fixed = TRUE</code> .
<code>mu0_ubound</code>	Optional upper bound for <code>mu0</code> . Ignored if <code>mu0_fixed = TRUE</code> .
<code>sigma0_fixed</code>	Logical. If <code>sigma0_fixed = TRUE</code> , initial mean vector <code>sigma0</code> is fixed. If <code>sigma0_fixed = FALSE</code> , initial mean vector <code>sigma0</code> is estimated.
<code>sigma0_diag</code>	Logical. If <code>sigma0_diag = TRUE</code> , <code>sigma0</code> is a diagonal matrix.
<code>sigma0_start</code>	Optional starting values for <code>sigma0</code> . If <code>sigma0_fixed = TRUE</code> , <code>sigma0_start</code> will be used as fixed values. If <code>sigma0_fixed = FALSE</code> , <code>sigma0_start</code> will be used as starting values.
<code>sigma0_lbound</code>	Optional lower bound for <code>sigma0</code> . Ignored if <code>sigma0_fixed = TRUE</code> .
<code>sigma0_ubound</code>	Optional upper bound for <code>sigma0</code> . Ignored if <code>sigma0_fixed = TRUE</code> .
<code>try</code>	Positive integer. Number of extra optimization tries.
<code>ncores</code>	Positive integer. Number of cores to use.

**Value**

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

**call** Function call.

**args** List of function arguments.

**fun** Function used ("FitDTVARIDMx").

**output** A list of fitted OpenMx models.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**References**

Hunter, M. D. (2017). State space modeling in an open source, modular, structural equation modeling environment. *Structural Equation Modeling: A Multidisciplinary Journal*, 25(2), 307–324. doi:[10.1080/10705511.2017.1369354](https://doi.org/10.1080/10705511.2017.1369354)

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., Estabrook, R., Bates, T. C., Maes, H. H., & Boker, S. M. (2015). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika*, 81(2), 535–549. doi:[10.1007/s1133601494358](https://doi.org/10.1007/s1133601494358)

**See Also**

Other DTVAR Functions: [FitDTVARmX\(\)](#)

**Examples**

```
## Not run:
# Generate data using the simStateSpace package-----
set.seed(42)
beta_mu <- matrix(
  data = c(
    0.7, 0.5, -0.1,
    0.0, 0.6, 0.4,
    0, 0, 0.5
  ),
  nrow = 3
)
beta_sigma <- diag(3 * 3)
beta <- simStateSpace::SimBetaN(
  n = 5,
  beta = beta_mu,
  vcov_beta_vec_1 = t(chol(beta_sigma))
)
sim <- simStateSpace::SimSSMVARIVary(
  n = 5,
  time = 100,
  mu0 = list(rep(x = 0, times = 3)),
  sigma0_1 = list(t(chol(diag(3)))),
  alpha = list(rep(x = 0, times = 3)),
```

```

    beta = beta,
    psi_l = list(t(chol(diag(3))))
  )
data <- as.data.frame(sim)

# Fit the model-----
library(fitDTVARMx)
fit <- FitDTVARIDMx(
  data = data,
  observed = c("y1", "y2", "y3"),
  id = "id"
)
print(fit)
summary(fit)
coef(fit)
vcov(fit)

## End(Not run)

```

---

FitDTVARMx

---

*Fit the First-Order Discrete-Time Vector Autoregressive Model*


---

## Description

Fit the First-Order Discrete-Time Vector Autoregressive Model

## Usage

```

FitDTVARMx(
  data,
  observed,
  id,
  beta_start = NULL,
  beta_lbound = NULL,
  beta_ubound = NULL,
  psi_diag = TRUE,
  psi_start = NULL,
  psi_lbound = NULL,
  psi_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

<code>data</code>	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.
<code>observed</code>	Character vector. A vector of character strings of the names of the observed variables in the data.
<code>id</code>	Character string. A character string of the name of the ID variable in the data.
<code>beta_start</code>	Numeric matrix. Optional starting values for beta.
<code>beta_lbound</code>	Numeric matrix. Optional lower bound for beta.
<code>beta_ubound</code>	Numeric matrix. Optional upper bound for beta.
<code>psi_diag</code>	Logical. If <code>psi_diag = TRUE</code> , <code>psi</code> is a diagonal matrix.
<code>psi_start</code>	Numeric matrix. Optional starting values for psi.
<code>psi_lbound</code>	Numeric matrix. Optional lower bound for psi.
<code>psi_ubound</code>	Optional upper bound for psi.
<code>theta_fixed</code>	Logical. If <code>theta_fixed = TRUE</code> , the measurement error matrix <code>theta</code> is fixed to zero. If <code>theta_fixed = FALSE</code> , estimate the diagonal measurement error matrix <code>theta</code> .
<code>theta_start</code>	Optional starting values for <code>theta</code> . Ignored if <code>theta_fixed = TRUE</code> .
<code>theta_lbound</code>	Optional lower bound for <code>theta</code> . Ignored if <code>theta_fixed = TRUE</code> .
<code>theta_ubound</code>	Optional upper bound for <code>theta</code> . Ignored if <code>theta_fixed = TRUE</code> .
<code>mu0_fixed</code>	Logical. If <code>mu0_fixed = TRUE</code> , initial mean vector <code>mu0</code> is fixed. If <code>mu0_fixed = FALSE</code> , initial mean vector <code>mu0</code> is estimated.
<code>mu0_start</code>	Optional starting values for <code>mu0</code> . If <code>mu0_fixed = TRUE</code> , <code>mu0_start</code> will be used as fixed values. If <code>mu0_fixed = FALSE</code> , <code>mu0_start</code> will be used as starting values.
<code>mu0_lbound</code>	Optional lower bound for <code>mu0</code> . Ignored if <code>mu0_fixed = TRUE</code> .
<code>mu0_ubound</code>	Optional upper bound for <code>mu0</code> . Ignored if <code>mu0_fixed = TRUE</code> .
<code>sigma0_fixed</code>	Logical. If <code>sigma0_fixed = TRUE</code> , initial mean vector <code>sigma0</code> is fixed. If <code>sigma0_fixed = FALSE</code> , initial mean vector <code>sigma0</code> is estimated.
<code>sigma0_diag</code>	Logical. If <code>sigma0_diag = TRUE</code> , <code>sigma0</code> is a diagonal matrix.
<code>sigma0_start</code>	Optional starting values for <code>sigma0</code> . If <code>sigma0_fixed = TRUE</code> , <code>sigma0_start</code> will be used as fixed values. If <code>sigma0_fixed = FALSE</code> , <code>sigma0_start</code> will be used as starting values.

<code>sigma0_lbound</code>	Optional lower bound for <code>sigma0</code> . Ignored if <code>sigma0_fixed = TRUE</code> .
<code>sigma0_ubound</code>	Optional upper bound for <code>sigma0</code> . Ignored if <code>sigma0_fixed = TRUE</code> .
<code>try</code>	Positive integer. Number of extra optimization tries.
<code>ncores</code>	Positive integer. Number of cores to use.

### Value

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

**call** Function call.

**args** List of function arguments.

**fun** Function used ("FitDTVARMx").

**output** A fitted OpenMx model.

### Author(s)

Ivan Jacob Agaloos Pesigan

### References

Hunter, M. D. (2017). State space modeling in an open source, modular, structural equation modeling environment. *Structural Equation Modeling: A Multidisciplinary Journal*, 25(2), 307–324. doi:[10.1080/10705511.2017.1369354](https://doi.org/10.1080/10705511.2017.1369354)

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., Estabrook, R., Bates, T. C., Maes, H. H., & Boker, S. M. (2015). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika*, 81(2), 535–549. doi:[10.1007/s1133601494358](https://doi.org/10.1007/s1133601494358)

### See Also

Other DTVAR Functions: [FitDTVARIIDMx\(\)](#)

### Examples

```
## Not run:
# Generate data using the simStateSpace package-----
set.seed(42)
sim <- simStateSpace::SimSSMVARFixed(
  n = 5,
  time = 100,
  mu0 = rep(x = 0, times = 3),
  sigma0_l = t(chol(diag(3))),
  alpha = rep(x = 0, times = 3),
  beta = matrix(
    data = c(
      0.7, 0.5, -0.1,
      0.0, 0.6, 0.4,
      0, 0, 0.5
    ),
    nrow = 3
```



```

    ),
    psi_l = t(chol(diag(3)))
  )
data <- as.data.frame(sim)

# Fit the model-----
library(fitDTVARMx)
fit <- FitDTVARMx(
  data = data,
  observed = c("y1", "y2", "y3"),
  id = "id"
)
print(fit)
summary(fit)
coef(fit)
vcov(fit)

## End(Not run)

```

---

print.fitdtvaridmx	<i>Print Method for Object of Class fitdtvaridmx</i>
--------------------	--

---

## Description

Print Method for Object of Class fitdtvaridmx

## Usage

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

## Arguments

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

## Author(s)

Ivan Jacob Agaloos Pesigan

---

print.fitdtvarmx	<i>Print Method for Object of Class fitdtvarmx</i>
------------------	--

---

**Description**

Print Method for Object of Class fitdtvarmx

**Usage**

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

**Arguments**

x	an object of class fitdtvarmx.
...	further arguments.

**Author(s)**

Ivan Jacob Agaloos Pesigan

---

---

summary.fitdtvaridmx	<i>Summary Method for Object of Class fitdtvaridmx</i>
----------------------	--

---

**Description**

Summary Method for Object of Class fitdtvaridmx

**Usage**

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

**Arguments**

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

**Author(s)**

Ivan Jacob Agaloos Pesigan

---

summary.fitdvarmx	<i>Summary Method for Object of Class fitdvarmx</i>
-------------------	---

---

**Description**

Summary Method for Object of Class fitdvarmx

**Usage**

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

**Arguments**

object	an object of class fitdvarmx.
...	further arguments.

**Author(s)**

Ivan Jacob Agaloos Pesigan

---

vcov.fitdvaridmx	<i>Sampling Covariance Matrix of the Parameter Estimates</i>
------------------	--

---

**Description**

Sampling Covariance Matrix of the Parameter Estimates

**Usage**

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

**Arguments**

object	Object of class fitdvaridmx.
psi	Logical. If psi = TRUE, include estimates of the psi matrix, if available. If psi = FALSE, exclude estimates of the psi 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.fitdvarmx

*Sampling Covariance Matrix of the Parameter Estimates*

---

**Description**

Sampling Covariance Matrix of the Parameter Estimates

**Usage**

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

**Arguments**

object	Object of class fitdvarmx.
psi	Logical. If psi = TRUE, include estimates of the psi matrix, if available. If psi = FALSE, exclude estimates of the psi 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

## \* **DTVAR Functions**

FitDTVARIDMx, [3](#)

FitDTVARMx, [6](#)

## \* **fitDTVARMx**

FitDTVARIDMx, [3](#)

FitDTVARMx, [6](#)

## \* **fit**

FitDTVARIDMx, [3](#)

FitDTVARMx, [6](#)

## \* **methods**

coef.fitdtvaridmx, [2](#)

coef.fitdtvarmx, [2](#)

print.fitdtvaridmx, [9](#)

print.fitdtvarmx, [10](#)

summary.fitdtvaridmx, [10](#)

summary.fitdtvarmx, [11](#)

vcov.fitdtvaridmx, [11](#)

vcov.fitdtvarmx, [12](#)

coef.fitdtvaridmx, [2](#)

coef.fitdtvarmx, [2](#)

FitDTVARIDMx, [3](#), [8](#)

FitDTVARMx, [5](#), [6](#)

print.fitdtvaridmx, [9](#)

print.fitdtvarmx, [10](#)

summary.fitdtvaridmx, [10](#)

summary.fitdtvarmx, [11](#)

vcov.fitdtvaridmx, [11](#)

vcov.fitdtvarmx, [12](#)