

# Package ‘metaVAR’

July 26, 2024

**Title** Multivariate Meta-Analysis of Vector Autoregressive Model Coefficients

**Version** 0.9.1

**Description** Estimates the mean vector and covariance matrix of the multivariate meta-analysis of vector autoregressive model coefficients.

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

**BugReports** <https://github.com/jeksterslab/metaVAR/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**Depends** R (>= 3.5.0), OpenMx

**Imports** numDeriv, Matrix, fitDTVARMx, fitCTVARMx

**Remotes** jeksterslab/fitDTVARMx, jeksterslab/fitCTVARMx

**Suggests** knitr, rmarkdown, testthat, simStateSpace, MASS

**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.metavarmeta . . . . .	2
Meta . . . . .	2
MetaVARMx . . . . .	4
print.metavarmeta . . . . .	6
summary.metavarmeta . . . . .	7
vcov.metavarmeta . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

coef.metavarmeta	<i>Estimated Parameter Method for an Object of Class metavarmeta</i>
------------------	--

---

### Description

Estimated Parameter Method for an Object of Class metavarmeta

### Usage

```
## S3 method for class 'metavarmeta'
coef(object, ...)
```

### Arguments

object	an object of class metavarmeta.
...	further arguments.

### Value

Returns a vector of the mean estimated parameters.

### Author(s)

Ivan Jacob Agaloos Pesigan

---

Meta	<i>Fit Multivariate Meta-Analysis</i>
------	---------------------------------------

---

### Description

This function estimates the mean and covariance matrix of a vector of coefficients using the estimated coefficients and sampling variance-covariance matrix from each individual.

### Usage

```
Meta(
  y,
  v,
  mu_start = NULL,
  mu_lbound = NULL,
  mu_ubound = NULL,
  sigma_l_start = NULL,
  sigma_l_lbound = NULL,
  sigma_l_ubound = NULL,
  diag = FALSE,
  try = 1000,
  ncores = NULL
)
```

### Arguments

<code>y</code>	A list. Each element of the list is a numeric vector of estimated coefficients.
<code>v</code>	A list. Each element of the list is a sampling variance-covariance matrix of <code>y</code> .
<code>mu_start</code>	Numeric vector. Optional vector of starting values for <code>mu</code> .
<code>mu_lbound</code>	Numeric vector. Optional vector of lower bound values for <code>mu</code> .
<code>mu_ubound</code>	Numeric vector. Optional vector of upper bound values for <code>mu</code> .
<code>sigma_l_start</code>	Numeric matrix. Optional matrix of starting values for <code>t(chol(sigma))</code> .
<code>sigma_l_lbound</code>	Numeric matrix. Optional matrix of lower bound values for <code>t(chol(sigma))</code> .
<code>sigma_l_ubound</code>	Numeric matrix. Optional matrix of upper bound values for <code>t(chol(sigma))</code> .
<code>diag</code>	Logical. If <code>diag = TRUE</code> , <code>sigma</code> is a diagonal matrix. If <code>diag = FALSE</code> , <code>sigma</code> is a symmetric matrix.
<code>try</code>	Positive integer. Number of extra optimization tries.
<code>ncores</code>	Positive integer. Number of cores to use.

### Details

For  $i = \{1, \dots, n\}$ , the objective function used to estimate the mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$  of the random coefficients  $\mathbf{y}_i$  is given by

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \mathbf{y}_i, \mathbb{V}(\mathbf{y}_i)) = -\frac{1}{2} \left[ q \log(2\pi) + \log(|\mathbb{V}(\mathbf{y}_i) - \boldsymbol{\Sigma}|) + (\mathbf{y}_i - \boldsymbol{\mu})' (\mathbb{V}(\mathbf{y}_i) - \boldsymbol{\Sigma})^{-1} (\mathbf{y}_i - \boldsymbol{\mu}) \right]$$

where  $q$  is the number of unique elements in  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ , and  $\mathbb{V}(\mathbf{y}_i)$  is the sampling variance-covariance matrix of  $\mathbf{y}_i$ .

### Author(s)

Ivan Jacob Agaloos Pesigan

### References

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

### See Also

Other Meta-Analysis of VAR Functions: [MetaVARMx\(\)](#)

**Description**

This function estimates the mean and covariance matrix of a vector of coefficients using the estimated coefficients and sampling variance-covariance matrix from each individual.

**Usage**

```
MetaVARMx(
  object,
  mu_start = NULL,
  mu_lbound = NULL,
  mu_ubound = NULL,
  sigma_l_start = NULL,
  sigma_l_lbound = NULL,
  sigma_l_ubound = NULL,
  diag = FALSE,
  intercept = FALSE,
  noise = FALSE,
  error = FALSE,
  try = 1000,
  ncores = NULL
)
```

**Arguments**

object	Output of the <code>fitDTVARMx::FitDTVARIDMx()</code> or <code>fitCTVARMx::FitCTVARIDMx()</code> functions.
mu_start	Numeric vector. Optional vector of starting values for mu.
mu_lbound	Numeric vector. Optional vector of lower bound values for mu.
mu_ubound	Numeric vector. Optional vector of upper bound values for mu.
sigma_l_start	Numeric matrix. Optional matrix of starting values for $t(\text{chol}(\text{sigma}))$ .
sigma_l_lbound	Numeric matrix. Optional matrix of lower bound values for $t(\text{chol}(\text{sigma}))$ .
sigma_l_ubound	Numeric matrix. Optional matrix of upper bound values for $t(\text{chol}(\text{sigma}))$ .
diag	Logical. If <code>diag = TRUE</code> , sigma is a diagonal matrix. If <code>diag = FALSE</code> , sigma is a symmetric matrix.
intercept	Logical. If <code>intercept = TRUE</code> , include estimates of the process intercept vector, if available. If <code>intercept = FALSE</code> , exclude estimates of the process intercept vector.
noise	Logical. If <code>noise = TRUE</code> , include estimates of the process noise matrix, if available. If <code>noise = FALSE</code> , exclude estimates of the process noise matrix.

error	Logical. If error = TRUE, include estimates of the measurement error matrix, if available. If error = FALSE, exclude estimates of the measurement error matrix.
try	Positive integer. Number of extra optimization tries.
ncores	Positive integer. Number of cores to use.

### Details

For  $i = \{1, \dots, n\}$ , the objective function used to estimate the mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$  of the random coefficients  $\mathbf{y}_i$  is given by

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \mathbf{y}_i, \mathbb{V}(\mathbf{y}_i)) = -\frac{1}{2} \left[ q \log(2\pi) + \log(|\mathbb{V}(\mathbf{y}_i) - \boldsymbol{\Sigma}|) + (\mathbf{y}_i - \boldsymbol{\mu})' (\mathbb{V}(\mathbf{y}_i) - \boldsymbol{\Sigma})^{-1} (\mathbf{y}_i - \boldsymbol{\mu}) \right]$$

where  $q$  is the number of unique elements in  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ , and  $\mathbb{V}(\mathbf{y}_i)$  is the sampling variance-covariance matrix of  $\mathbf{y}_i$ .

### Author(s)

Ivan Jacob Agaloos Pesigan

### References

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

### See Also

Other Meta-Analysis of VAR Functions: [Meta\(\)](#)

### Examples

```
## Not run:
# Generate data using the simStateSpace package-----
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_l = 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"
)
# Multivariate meta-analysis-----
library(metaVAR)
meta <- MetaVARMx(fit)
print(meta)
summary(meta)
coef(meta)
vcov(meta)

## End(Not run)

```

---

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

---

## Description

Print Method for Object of Class metavarmeta

## Usage

```

## S3 method for class 'metavarmeta'
print(x, alpha = 0.05, digits = 4, ...)

```

## Arguments

x	an object of class metavarmeta.
alpha	Numeric vector. Significance level $\alpha$ .
digits	Integer indicating the number of decimal places to display.
...	further arguments.

## Author(s)

Ivan Jacob Agaloos Pesigan

---

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

---

**Description**

Summary Method for Object of Class metavarmeta

**Usage**

```
## S3 method for class 'metavarmeta'
summary(object, alpha = 0.05, digits = 4, ...)
```

**Arguments**

object	an object of class metavarmeta.
alpha	Numeric vector. Significance level $\alpha$ .
digits	Integer indicating the number of decimal places to display.
...	further arguments.

**Author(s)**

Ivan Jacob Agaloos Pesigan

---

vcov.metavarmeta	<i>Variance-Covariance Matrix Method for an Object of Class metavarmeta</i>
------------------	---

---

**Description**

Variance-Covariance Matrix Method for an Object of Class metavarmeta

**Usage**

```
## S3 method for class 'metavarmeta'
vcov(object, ...)
```

**Arguments**

object	an object of class metavarmeta.
...	further arguments.

**Value**

Returns the variance-covariance matrix of the estimated parameters.

**Author(s)**

Ivan Jacob Agaloos Pesigan

# Index

## \* Meta-Analysis of VAR Functions

Meta, [2](#)

MetaVARMx, [4](#)

## \* metaVAR

Meta, [2](#)

MetaVARMx, [4](#)

## \* meta

Meta, [2](#)

MetaVARMx, [4](#)

## \* methods

coef.metavarmeta, [2](#)

print.metavarmeta, [6](#)

summary.metavarmeta, [7](#)

vcov.metavarmeta, [7](#)

coef.metavarmeta, [2](#)

fitCTVARMx::FitCTVARIDMx(), [4](#)

fitDTVARMx::FitDTVARIIDMx(), [4](#)

Meta, [2](#), [5](#)

MetaVARMx, [3](#), [4](#)

print.metavarmeta, [6](#)

summary.metavarmeta, [7](#)

vcov.metavarmeta, [7](#)