

# Package ‘metaVAR’

May 3, 2025

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

**Version** 0.0.0.9000

**Description** Fits fixed-, random-, or mixed-effects multivariate meta-analysis models  
using vector autoregressive model estimates from each individual.

**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** Matrix, fitDTVARMx

**Remotes** jeksterslab/fitDTVARMx

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

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Ivan Jacob Agaloos Pesigan [aut, cre, cph] (ORCID:  
<<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 . . . . .	7
summary.metavarmeta . . . . .	7
vcov.metavarmeta . . . . .	8

<b>Index</b>	<b>9</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 estimated parameters.

### Author(s)

Ivan Jacob Agaloos Pesigan

---

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

---

### Description

This function estimates fixed-, random-, or mixed-effects meta-analysis parameters using the estimated coefficients and sampling variance-covariance matrix from each individual.

### Usage

```
Meta(
  y,
  v,
  x = NULL,
  alpha_values = NULL,
  alpha_free = NULL,
  alpha_lbound = NULL,
  alpha_ubound = NULL,
  beta_values = NULL,
  beta_free = NULL,
```

```

    beta_lbound = NULL,
    beta_ubound = NULL,
    tau_values = NULL,
    tau_free = NULL,
    tau_lbound = NULL,
    tau_ubound = NULL,
    random = TRUE,
    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>x</code>	An optional list. Each element of the list is a numeric vector of covariates for the mixed-effects model.
<code>alpha_values</code>	Numeric vector. Optional vector of starting values for <code>alpha</code> .
<code>alpha_free</code>	Logical vector. Optional vector of free (TRUE) parameters for <code>alpha</code> .
<code>alpha_lbound</code>	Numeric vector. Optional vector of lower bound values for <code>alpha</code> .
<code>alpha_ubound</code>	Numeric vector. Optional vector of upper bound values for <code>alpha</code> .
<code>beta_values</code>	Numeric matrix. Optional matrix of starting values for <code>beta</code> .
<code>beta_free</code>	Logical matrix. Optional matrix of free (TRUE) parameters for <code>beta</code> .
<code>beta_lbound</code>	Numeric matrix. Optional matrix of lower bound values for <code>beta</code> .
<code>beta_ubound</code>	Numeric matrix. Optional matrix of upper bound values for <code>beta</code> .
<code>tau_values</code>	Numeric matrix. Optional matrix of starting values for <code>t(chol(tau_sqr))</code> .
<code>tau_free</code>	Numeric matrix. Optional matrix of free (TRUE) parameters for <code>t(chol(tau_sqr))</code> .
<code>tau_lbound</code>	Numeric matrix. Optional matrix of lower bound values for <code>t(chol(tau_sqr))</code> .
<code>tau_ubound</code>	Numeric matrix. Optional matrix of upper bound values for <code>t(chol(tau_sqr))</code> .
<code>random</code>	Logical. If <code>random = TRUE</code> , estimates random effects. If <code>random = FALSE</code> , <code>tau_sqr</code> is a null matrix.
<code>diag</code>	Logical. If <code>diag = TRUE</code> , <code>tau_sqr</code> is a diagonal matrix. If <code>diag = FALSE</code> , <code>tau_sqr</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.
<code>...</code>	Additional optional arguments to pass to <code>mTryHardctsem</code> .

### Author(s)

Ivan Jacob Agaloos Pesigan

## References

Cheung, M. W.-L. (2015). *Meta-analysis: A structural equation modeling approach*. Wiley. doi:[10.1002/9781118957813](https://doi.org/10.1002/9781118957813)

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

---

MetaVARMx

*Fit Multivariate Meta-Analysis*

---

## Description

This function estimates fixed-, random-, or mixed-effects meta-analysis parameters using the estimated coefficients and sampling variance-covariance matrix from each individual fitted using the [fitDTVARMx::FitDTVARIDMx\(\)](#) function.

## Usage

```
MetaVARMx(
  object,
  x = NULL,
  alpha_values = NULL,
  alpha_free = NULL,
  alpha_lbound = NULL,
  alpha_ubound = NULL,
  beta_values = NULL,
  beta_free = NULL,
  beta_lbound = NULL,
  beta_ubound = NULL,
  tau_values = NULL,
  tau_free = NULL,
  tau_lbound = NULL,
  tau_ubound = NULL,
  random = TRUE,
  diag = FALSE,
  intercept = FALSE,
  noise = FALSE,
  error = FALSE,
  try = 1000,
  ncores = NULL,
  ...
)
```

**Arguments**

object	Output of the <code>fitDTVARMx::FitDTVARIDMx()</code> function.
x	An optional list. Each element of the list is a numeric vector of covariates for the mixed-effects model.
alpha_values	Numeric vector. Optional vector of starting values for alpha.
alpha_free	Logical vector. Optional vector of free (TRUE) parameters for alpha.
alpha_lbound	Numeric vector. Optional vector of lower bound values for alpha.
alpha_ubound	Numeric vector. Optional vector of upper bound values for alpha.
beta_values	Numeric matrix. Optional matrix of starting values for beta.
beta_free	Logical matrix. Optional matrix of free (TRUE) parameters for beta.
beta_lbound	Numeric matrix. Optional matrix of lower bound values for beta.
beta_ubound	Numeric matrix. Optional matrix of upper bound values for beta.
tau_values	Numeric matrix. Optional matrix of starting values for $t(\text{chol}(\text{tau\_sqr}))$ .
tau_free	Numeric matrix. Optional matrix of free (TRUE) parameters for $t(\text{chol}(\text{tau\_sqr}))$ .
tau_lbound	Numeric matrix. Optional matrix of lower bound values for $t(\text{chol}(\text{tau\_sqr}))$ .
tau_ubound	Numeric matrix. Optional matrix of upper bound values for $t(\text{chol}(\text{tau\_sqr}))$ .
random	Logical. If <code>random = TRUE</code> , estimates random effects. If <code>random = FALSE</code> , <code>tau_sqr</code> is a null matrix.
diag	Logical. If <code>diag = TRUE</code> , <code>tau_sqr</code> is a diagonal matrix. If <code>diag = FALSE</code> , <code>tau_sqr</code> 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 <code>error = TRUE</code> , include estimates of the measurement error matrix, if available. If <code>error = FALSE</code> , exclude estimates of the measurement error matrix.
try	Positive integer. Number of extra optimization tries.
ncores	Positive integer. Number of cores to use.
...	Additional optional arguments to pass to <code>mxTryHardctsem</code> .

**Author(s)**

Ivan Jacob Agaloos Pesigan

**References**

- Cheung, M. W.-L. (2015). *Meta-analysis: A structural equation modeling approach*. Wiley. doi:10.1002/9781118957813
- 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_l = 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.

**Value**

Returns a matrix of estimates, standard errors, test statistics, degrees of freedom, p-values, and confidence intervals.

**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.

**Value**

Returns a matrix of estimates, standard errors, test statistics, degrees of freedom, p-values, and confidence intervals.

**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 sampling 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, [7](#)

summary.metavarmeta, [7](#)

vcov.metavarmeta, [8](#)

coef.metavarmeta, [2](#)

fitDTVARmX::FitDTVARIDmX(), [4](#), [5](#)

Meta, [2](#), [6](#)

MetaVARmX, [4](#), [4](#)

print.metavarmeta, [7](#)

summary.metavarmeta, [7](#)

vcov.metavarmeta, [8](#)