Package ‘metaVAR’

July 7, 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

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 — *Estimated Parameter Method for an Object of Class* `metavarmeta`

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 — *Fit Multivariate Meta-Analysis*

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

Arguments

`y`	A list. Each element of the list is a numeric vector of estimated coefficients.
`v`	A list. Each element of the list is a sampling variance-covariance matrix of `y`.
`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(chol(sigma))`.
`sigma_l_lbound`	Numeric matrix. Optional matrix of lower bound values for `t(chol(sigma))`.
`sigma_l_ubound`	Numeric matrix. Optional matrix of upper bound values for `t(chol(sigma))`.
`try`	Positive integer. Number of extra tries for `OpenMx::mxTryHard()`.
`ncores`	Positive integer. Number of cores to use.

Details

For $i = \{1, \cdots, n\}$, 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

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Meta-Analysis of VAR Functions: `MetaVARMx()`

`MetaVARMx` *Fit Multivariate Meta-Analysis*

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,
  noise = FALSE,
  error = FALSE,
  try = 1000,
  ncores = NULL
)
```

Arguments

`object`	Output of the `fitDTVARMx::FitDTVARIDMx()` or `fitCTVARMx::FitCTVARIDMx()` 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(chol(sigma))`.
`sigma_l_lbound`	Numeric matrix. Optional matrix of lower bound values for `t(chol(sigma))`.
`sigma_l_ubound`	Numeric matrix. Optional matrix of upper bound values for `t(chol(sigma))`.
`noise`	Logical. If `noise = TRUE`, include estimates of the process noise matrix, if available. If `noise = FALSE`, 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 tries for `OpenMx::mxTryHard()`.
`ncores`	Positive integer. Number of cores to use.

Details

For $i = \{1, \cdots, n\}$, 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

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

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

Author(s)

Ivan Jacob Agaloos Pesigan

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 *Print Method for Object of Class* `metavarmeta`

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 α.
`digits`	Integer indicating the number of decimal places to display.
`...`	further arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

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

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 α.
`digits`	Integer indicating the number of decimal places to display.
`...`	further arguments.

Author(s)

Ivan Jacob Agaloos Pesigan

vcov.metavarmeta *Variance-Covariance Matrix Method for an Object of Class* metavarmeta

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