Package 'metaVAR'

August 5, 2024

Title Multivariate Meta-Analysis of Vector Autoregressive Model Coefficients

Version 0.0.0.9000

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 *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 estimated parameters.

Author(s)

Ivan Jacob Agaloos Pesigan

confint.metavarmeta *Confidence Intervals for an Object of Class* metavarmeta

Description

Confidence Intervals for an Object of Class metavarmeta

Usage

```
## S3 method for class 'metavarmeta'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

object	Object of class metavarmeta.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
...	additional arguments.

Value

Returns a matrix of confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

`Meta` *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.

Usage

```
Meta(
  y,
  v,
  x = NULL,
  beta0_values = NULL,
  beta0_free = NULL,
  beta0_lbound = NULL,
  beta0_ubound = NULL,
  beta1_values = NULL,
  beta1_free = NULL,
  beta1_lbound = NULL,
  beta1_ubound = NULL,
  tau_values = NULL,
  tau_free = NULL,
  tau_lbound = NULL,
  tau_ubound = NULL,
  random = TRUE,
  diag = FALSE,
  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`.
`x`	An optional list. Each element of the list is a numeric vector of covariates for the mixed-effects model.

beta0_values	Numeric vector. Optional vector of starting values for beta0.
beta0_free	Logical vector. Optional vector of free (TRUE) parameters for beta0.
beta0_lbound	Numeric vector. Optional vector of lower bound values for beta0.
beta0_ubound	Numeric vector. Optional vector of upper bound values for beta0.
beta1_values	Numeric matrix. Optional matrix of starting values for beta1.
beta1_free	Logical matrix. Optional matrix of free (TRUE) parameters for beta1.
beta1_lbound	Numeric matrix. Optional matrix of lower bound values for beta1.
beta1_ubound	Numeric matrix. Optional matrix of upper bound values for beta1.
tau_values	Numeric matrix. Optional matrix of starting values for t(chol(tau_sqr)).
tau_free	Numeric matrix. Optional matrix of free (TRUE) parameters for t(chol(tau_sqr)).
tau_lbound	Numeric matrix. Optional matrix of lower bound values for t(chol(tau_sqr)).
tau_ubound	Numeric matrix. Optional matrix of upper bound values for t(chol(tau_sqr)).
random	Logical. If random = TRUE, estimates random effects. If random = FALSE, tau_sqr is a null matrix.
diag	Logical. If diag = TRUE, tau_sqr is a diagonal matrix. If diag = FALSE, tau_sqr is a symmetric matrix.
try	Positive integer. Number of extra optimization tries.
ncores	Positive integer. Number of cores to use.
...	Additional optional arguments to pass to mxTryHardctsem.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Cheung, M. W.-L. (2015). *Meta-analysis: A structural equation modeling approach*. Chichester, West Sussex: John Wiley & Sons, Inc.

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()

`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.

Usage

```
MetaVARMx(
  object,
  x = NULL,
  beta0_values = NULL,
  beta0_free = NULL,
  beta0_lbound = NULL,
  beta0_ubound = NULL,
  beta1_values = NULL,
  beta1_free = NULL,
  beta1_lbound = NULL,
  beta1_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 `fitDTVARMx::FitDTVARIDMx()` or `fitCTVARMx::FitCTVARIDMx()` functions.
`x`	An optional list. Each element of the list is a numeric vector of covariates for the mixed-effects model.
`beta0_values`	Numeric vector. Optional vector of starting values for `beta0`.
`beta0_free`	Logical vector. Optional vector of free (`TRUE`) parameters for `beta0`.
`beta0_lbound`	Numeric vector. Optional vector of lower bound values for `beta0`.
`beta0_ubound`	Numeric vector. Optional vector of upper bound values for `beta0`.

beta1_values	Numeric matrix. Optional matrix of starting values for beta1.
beta1_free	Logical matrix. Optional matrix of free (TRUE) parameters for beta1.
beta1_lbound	Numeric matrix. Optional matrix of lower bound values for beta1.
beta1_ubound	Numeric matrix. Optional matrix of upper bound values for beta1.
tau_values	Numeric matrix. Optional matrix of starting values for t(chol(tau_sqr)).
tau_free	Numeric matrix. Optional matrix of free (TRUE) parameters for t(chol(tau_sqr)).
tau_lbound	Numeric matrix. Optional matrix of lower bound values for t(chol(tau_sqr)).
tau_ubound	Numeric matrix. Optional matrix of upper bound values for t(chol(tau_sqr)).
random	Logical. If random = TRUE, estimates random effects. If random = FALSE, tau_sqr is a null matrix.
diag	Logical. If diag = TRUE, tau_sqr is a diagonal matrix. If diag = FALSE, tau_sqr is a symmetric matrix.
intercept	Logical. If intercept = TRUE, include estimates of the process intercept vector, if available. If intercept = FALSE, exclude estimates of the process intercept vector.
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 optimization tries.
ncores	Positive integer. Number of cores to use.
...	Additional optional arguments to pass to mxTryHardctsem.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Cheung, M. W.-L. (2015). *Meta-analysis: A structural equation modeling approach*. Chichester, West Sussex: John Wiley & Sons, Inc.

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 *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.

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` *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.

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` *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 sampling variance-covariance matrix of the estimated parameters.

Author(s)

Ivan Jacob Agaloos Pesigan

Index