Package ‘metafrontieR’

February 28, 2026

Title Metafrontier Analysis Routines

Version 1.0.0

Author Sulman Olieko Owili [aut, cre]

Maintainer Sulman Olieko Owili <oliekosulman@gmail.com>

Description An R package for implementing various metafrontier analyses for productivity/performance benchmarking, assessing efficiencies and technology gaps for firms operating under different technologies. It contains routines for implementing (i) the deterministic envelope proposed by O'Donnell et al. (2008) <doi:10.1007/s00181-007-0119-4> via linear and quadratic programming, as well as (ii) the stochastic metafrontier proposed by Huang et al. (2014) <doi:10.1007/s11123-014-0402-2>. The package also has functionalities for implementing latent class stochastic metafrontier analysis and sample selection correction stochastic metafrontier models. The package depends on sfaR by Dakpo et al. (2024) <https://github.com/hdakpo/sfaR>.

License GPL (>= 3)

URL https://github.com/SulmanOlieko/metafrontieR

BugReports https://github.com/SulmanOlieko/metafrontieR/issues

Depends R (>= 3.5.0),
sfaR

Imports stats

Suggests lmtest

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Contents

coef *Extract coefficients of stochastic metafrontier models*

Description

From an object of class 'summary.sfametafrontier', coef extracts the coefficients, their standard errors, z-values, and (asymptotic) P-values.

From on object of class 'sfametafrontier', it extracts only the estimated coefficients.

Usage

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

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

Arguments

`object`	A stochastic metafrontier model returned by sfametafrontier, or an object of class 'summary.sfametafrontier'.
`...`	Currently ignored.

Value

For objects of class 'summary.sfametafrontier', coef returns a matrix with four columns. Namely, the estimated coefficients, their standard errors, z-values, and (asymptotic) P-values.

For objects of class 'sfametafrontier', coef returns a numeric vector of the estimated coefficients.

See Also

sfametafrontier, for the stochastic metafrontier analysis model fitting function using cross-sectional or pooled data.

efficiencies — *Compute conditional (in-)efficiency estimates of stochastic metafrontier models*

Description

efficiencies returns (in-)efficiency estimates of models estimated with sfametafrontier.

Usage

```
## S3 method for class 'sfametafrontier'
efficiencies(object, level = 0.95, newData = NULL, ...)
```

Arguments

object	A stochastic metafrontier model returned by sfametafrontier.
level	A number between between 0 and 0.9999 used for the computation of (in-)efficiency confidence intervals (defaut = 0.95).
newData	Optional data frame that is used to calculate the efficiency estimates. If NULL (the default), the efficiency estimates are calculated for the observations that were used in the estimation.
...	Currently ignored.

Details

The metatechnology ratio (MTR) and metafrontier efficiencies are computed as follows:

- **Group Efficiency:** $TE_{it}^g = \exp(-u_{it})$, evaluated natively by the group-specific frontier (e.g. using Jondrow et al., 1982 or Battese and Coelli, 1988).
- **Metatechnology Ratio (MTR):** Evaluates the distance between the group-specific frontier and the global metafrontier.
  - For deterministic envelopes ("lp", "qp"):
    $$MTR_{it} = \exp\left(-\max\left\{S \times (\ln \hat{y}_{it}^* - \ln \hat{y}_{it}^g), 0\right\}\right)$$
    where $S = 1$ for production/profit and $S = -1$ for cost.
  - For Huang et al. (2014) stochastic approach:
    $$MTR_{it} = \exp(-U_{it})$$
    directly estimated as the technical efficiency from the second-stage SFA regression where the dependent variable is the group-fitted values $\ln \hat{y}_{it}^g$.
  - For O'Donnell et al. (2008) stochastic approach: $MTR_{it} = TE_{it}^*/TE_{it}^g$, potentially vulnerable to bounding issues if $TE_{it}^* > TE_{it}^g$.
- **Metafrontier Efficiency:** $TE_{it}^* = TE_{it}^g \times MTR_{it}$.

Value

A data frame containing individual (in-)efficiency estimates, ordered corresponding to the original data used for estimation. The specific columns vary depending on the model and `sfaApproach`, but generally include:

`u_group`	Group-specific inefficiency estimates.
`TE_group_JLMS`	Group-specific technical efficiency estimates using Jondrow et al. (1982).
`TE_group_BC`	Group-specific technical efficiency estimates using Battese and Coelli (1988).
`u_meta`	Metafrontier inefficiency estimates.
`TE_meta_JLMS`	Metafrontier technical efficiency estimates using JLMS.
`TE_meta_BC`	Metafrontier technical efficiency estimates using BC.
`MTR_JLMS`	Metatechnology ratio based on JLMS estimates.
`MTR_BC`	Metatechnology ratio based on BC estimates.

References

Battese, G. E., Rao, D. S. P., and O'Donnell, C. J. 2004. A metafrontier production function for estimation of technical efficiencies and technology gaps for firms operating under different technologies. *Journal of Productivity Analysis*, **21**(1), 91–103. https://doi.org/10.1023/B:PROD.0000012454.06094.29

Huang, C. J., Huang, T.-H., and Liu, N.-H. 2014. A new approach to estimating the metafrontier production function based on a stochastic frontier framework. *Journal of Productivity Analysis*, **42**(3), 241–254. https://doi.org/10.1007/s11123-014-0402-2

O'Donnell, C. J., Rao, D. S. P., and Battese, G. E. 2008. Metafrontier frameworks for the study of firm-level efficiencies and technology ratios. *Empirical Economics*, **34**(2), 231–255. https://doi.org/10.1007/s00181-007-0119-4

See Also

`sfametafrontier`, for the stochastic metafrontier analysis model fitting function using cross-sectional or pooled data.

`fitted` — *Extract fitted values of stochastic metafrontier models*

Description

`fitted` returns the fitted frontier values from stochastic metafrontier models estimated with `sfametafrontier`.

Usage

```
## S3 method for class 'sfametafrontier'
fitted(object, ...)
```

Arguments

`object`	A stochastic metafrontier model returned by `sfametafrontier`.
`...`	Currently ignored.

Value

A vector of fitted values is returned.

Note

The fitted values are ordered in the same way as the corresponding observations in the dataset used for the estimation.

See Also

sfametafrontier, for the stochastic metafrontier analysis model fitting function using cross-sectional or pooled data.

---

ic *Extract information criteria of stochastic metafrontier models*

Description

ic returns information criterion from stochastic metafrontier models estimated with sfametafrontier.

Usage

```
## S3 method for class 'sfametafrontier'
ic(object, IC = "AIC", ...)
```

Arguments

object A stochastic metafrontier model returned by sfametafrontier.

IC Character string. Information criterion measure. Three criteria are available:

- 'AIC' for Akaike information criterion (default)
- 'BIC' for Bayesian information criterion
- 'HQIC' for Hannan-Quinn information criterion

.

... Currently ignored.

Details

The different information criteria are computed as follows:

- AIC: $-2 \log LL + 2 * K$
- BIC: $-2 \log LL + \log N * K$
- HQIC: $-2 \log LL + 2 \log [\log N] * K$

where LL is the maximum likelihood value, K the number of parameters estimated and N the number of observations.

Value

ic returns the value of the information criterion (AIC, BIC or HQIC) of the maximum likelihood coefficients.

See Also

`sfametafrontier`, for the stochastic metafrontier analysis model fitting function using cross-sectional or pooled data.

`logLik` *Extract log-likelihood value of stochastic metafrontier models*

Description

`logLik` extracts the log-likelihood value(s) from stochastic metafrontier models estimated with `sfametafrontier`.

Usage

```
## S3 method for class 'sfametafrontier'
logLik(object, individual = FALSE, ...)
```

Arguments

`object`	A stochastic metafrontier model returned by `sfametafrontier`.
`individual`	Logical. If `FALSE` (default), the sum of all observations' log-likelihood values is returned. If `TRUE`, a vector of each observation's log-likelihood value is returned.
`...`	Currently ignored.

Value

`logLik` returns either an object of class `'logLik'`, which is the log-likelihood value with the total number of observations (`nobs`) and the number of free parameters (`df`) as attributes, when `individual = FALSE`, or a list of elements, containing the log-likelihood of each observation (`logLik`), the total number of observations (`Nobs`) and the number of free parameters (`df`), when `individual = TRUE`.

See Also

`sfametafrontier`, for the stochastic metafrontier analysis model fitting function using cross-sectional or pooled data.

`nobs` *Extract total number of observations used in frontier models*

Description

This function extracts the total number of 'observations' from a fitted point frontier model.

Usage

```
## S3 method for class 'sfametafrontier'
nobs(object, ...)
```

Arguments

`object`	a `sfametafrontier` object for which the number of total observations is to be extracted.
`...`	Currently ignored.

Details

`nobs` gives the number of observations actually used by the estimation procedure.

Value

A single number, normally an integer.

See Also

`sfametafrontier`, for the stochastic metafrontier analysis model fitting function using cross-sectional or pooled data

---

`residuals` *Extract residuals of stochastic metafrontier models*

Description

This function returns the residuals' values from stochastic metafrontier models estimated with `sfametafrontier`.

Usage

```
## S3 method for class 'sfametafrontier'
residuals(object, ...)
```

Arguments

`object`	A stochastic metafrontier model returned by `sfametafrontier`.
`...`	Currently ignored.

Value

`residuals` returns a vector of residuals values.

Note

The residuals values are ordered in the same way as the corresponding observations in the dataset used for the estimation.

See Also

`sfametafrontier`, for the stochastic metafrontier analysis model fitting function using cross-sectional or pooled data.

sfametafrontier *Stochastic metafrontier estimation*

Description

sfametafrontier estimates a stochastic metafrontier model following the theoretical frameworks of Battese, Rao, and O'Donnell (2004) and O'Donnell, Rao, and Battese (2008). It additionally implements the two-stage stochastic approach of Huang, Huang, and Liu (2014). Three types of group-level frontier models from the sfaR package are supported: standard stochastic frontier analysis (sfacross), sample selection SFA (sfaselectioncross), and latent class SFA (sfalcmcross).

Usage

```
sfametafrontier(
  formula,
  muhet,
  uhet,
  vhet,
  thet,
  logDepVar = TRUE,
  data,
  subset,
  group = NULL,
  S = 1L,
  udist = "hnormal",
  scaling = FALSE,
  groupType = "sfacross",
  metaMethod = "lp",
  sfaApproach = "huang",
  selectionF = NULL,
  lcmClasses = 2L,
  whichStart = 2L,
  initAlg = "nm",
  initIter = 100L,
  lType = "ghermite",
  Nsub = 100L,
  uBound = Inf,
  intol = 1e-06,
  method = "bfgs",
  hessianType = 1L,
  simType = "halton",
  Nsim = 100L,
  prime = 2L,
  burn = 10L,
  antithetics = FALSE,
  seed = 12345L,
  itermax = 2000L,
  printInfo = FALSE,
  tol = 1e-12,
  gradtol = 1e-06,
  stepmax = 0.1,
```

```
    qac = "marquardt",
    ...
  )

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

Arguments

`formula` A symbolic description of the frontier model.

`muhet` A one-part formula for heterogeneity in the mean of the pre-truncated distribution (only for `groupType = "sfacross"`). This formula is used to capture structural differences in the modeled efficiency mean.

`uhet` A one-part formula for heteroscedasticity in the one-sided error term. To accommodate heteroscedasticity in the variance parameters, the variances are modeled as: $\sigma_u^2 = \exp(\delta' Z_u)$, where Z_u are the inefficiency drivers and δ are the coefficients.

`vhet` A one-part formula for heteroscedasticity in the two-sided error term. Modeled as $\sigma_v^2 = \exp(\phi' Z_v)$, where Z_v are the heteroscedasticity variables and ϕ the coefficients.

`thet` A one-part formula for technological heterogeneity in LCM class construction (only for `groupType = "sfalcmcross"`). The variables specified are used in the logit formulation of the finite mixture model to compute the prior class membership probabilities.

`logDepVar` Logical. Whether the dependent variable is logged. Default `TRUE`.

`data` The data frame.

`subset` An optional subset vector.

`group` Character string. Name of the column in `data` that identifies the technology groups. Must have at least 2 unique values.

`S` `1` (default) for production/profit frontier; `-1` for cost frontier.

`udist` Character string. Distribution for the one-sided error term. For `groupType = "sfacross"`: all 10 distributions supported (see `sfacross`). For `groupType = "sfaselectioncross"` or `"sfalcmcross"`: only `'hnormal'`.

`scaling` Logical. Scaling property model for `groupType = "sfacross"` when `udist = 'tnormal'`. If `TRUE`, the scaling property is used to model the one-sided error conditional on the inefficiency drivers Z_u (e.g. $u = h(Z_u, \delta)u^*$ where u^* is a homoscedastic random variable). Default `FALSE`.

`groupType` Character string. Type of model used for each group's frontier. `"sfacross"` (default) estimates a standard cross-sectional SFA, `"sfaselectioncross"` estimates a sample selection SFA adjusting for bias via a generalized Heckman approach, or `"sfalcmcross"` estimates a latent class SFA estimating a pooled mixture model.

`metaMethod` Character string. Method for estimating the metafrontier.

- `"lp"` (default): Deterministic envelope (column-wise maximum of group frontier values evaluated at all observations).
- `"qp"`: Constrained OLS of the envelope on X.
- `"sfa"`: Second-stage pooled SFA. The approach depends on `sfaApproach`.

`sfaApproach`	Character string. Only relevant when `metaMethod = "sfa"`. `"huang"` (default): uses each observation's own group fitted value as the meta-stage dependent variable (Huang et al., 2014). `"ordonnell"`: uses the deterministic envelope of group betas evaluated at all observations (O'Donnell et al., 2008).
`selectionF`	A two-sided formula (e.g. `selected ~ z1 + z2`) or a named list of formulas (one per group) specifying the sample selection equation. Only used when `groupType = "sfaselectioncross"`.
`lcmClasses`	Integer (2–5). Number of latent classes for `groupType = "sfalcmcross"`. Default `2`. If `group` is not specified, `sfametafrontier` automatically splits the data evenly into `lcmClasses` classes.
`whichStart`	Integer. Strategy for obtaining initial values for LCM optimization. `whichStart = 1` uses initialized values for each class, whereas `whichStart = 2` (default) estimates a pooled base homoscedastic model to provide uniform starting points.
`initAlg`	Character. Initialization algorithm for LCM. A string specifying the non-gradient or gradient method. Options include `"nm"` for Nelder-Mead (default), `"bhhh"`, `"bfgs"`, `"cg"`, or `"sann"`.
`initIter`	Integer. Initialization iterations for LCM. Default `100`.
`lType`	Character. Likelihood type for selection model. Options include `"ghermite"` for Gauss-Hermite quadrature (default) or `"msl"` for Maximum Simulated Likelihood.
`Nsub`	Integer. Number of quadrature nodes/subdivisions for the selection model integration when `lType = "ghermite"`. Default `100`.
`uBound`	Numeric. Upper bound for integration in the selection model. Default `Inf`.
`intol`	Numeric. Integration tolerance for the selection model. Default `1e-6`.
`method`	Optimization algorithm for group models. Default `'bfgs'`. Other options include `"bhhh"`, `"nr"`, `"nm"`, `"cg"`, and `"sann"`.
`hessianType`	Integer (`1` or `2`). Determines how the Hessian matrix is computed for standard SFA models. Default `1` uses analytic (when available) or finite-difference numerical Hessians.
`simType`	Character. Simulation type for MSL. Options are `'halton'` (default), `'generalized_halton'`, `'sobol'`, or `'random'`.
`Nsim`	Integer. Number of MSL draws. Default `100`.
`prime`	Integer. Prime number for Halton draws. Default `2`.
`burn`	Integer. Number of initial Halton draws discarded. Default `10`.
`antithetics`	Logical. If `TRUE`, evaluates antithetic draws to reduce variance in MSL. Default `FALSE`.
`seed`	Numeric seed. Default `12345`.
`itermax`	Maximum iterations. Default `2000`.
`printInfo`	Logical. Default `FALSE`.
`tol`	Convergence tolerance. Default `1e-12`.
`gradtol`	Gradient tolerance. Default `1e-06`.
`stepmax`	Step max for `ucminf`. Default `0.1`.
`qac`	QAC for `'bhhh'`/`'nr'`. Default `'marquardt'`.
`...`	Additional arguments passed to the second-stage SFA call when `metaMethod = "sfa"`.
`x`	An object of class `'sfametafrontier'` (for printing).

Details

The stochastic metafrontier analysis workflow comprises two sequential stages. Suppose we have N observations divided into G technology groups. The underlying group frontier for firm i in group g at time t is specified as:

$$y_{it} = f(x_{it}, \beta_{(g)})e^{v_{it}-u_{it}}$$

where $f(\cdot)$ is the deterministic frontier (e.g., translog or Cobb-Douglas), x_{it} represents the input vector (or output vector for cost models), $\beta_{(g)}$ is the technology parameter vector for group g, v_{it} is the statistical noise, and $u_{it} \geq 0$ represents technical inefficiency.

1. **Group-specific estimations:** The data is split into G groups based on the `group` variable (unless `groupType="sfalcmcross"` with `group` omitted, in which case a single pooled latent class model is estimated to map observations to underlying groups automatically). For each explicitly or implicitly defined group, a frontier model is estimated natively by maximizing the log-likelihood function.
2. **Metafrontier estimation:** The global metafrontier $f(x_{it}, \beta^*)$ envelopes all group frontiers $f(x_{it}, \beta_{(g)})$. The methodologies are:
   - **Linear Programming (LP):** Fits a deterministic envelope over the fitted group frontier values by minimizing the absolute sum of deviations (Battese et al., 2004). The objective is to find a single parameter vector β^* such that the metafrontier predictions are completely uniformly larger than the group-specific predictions:
   $$\min_{\beta^*} \sum_i \sum_t \left| \ln f(x_{it}, \beta^*) - \ln \hat{f}(x_{it}, \hat{\beta}_{(g)}) \right|$$
   subject to $\ln f(x_{it}, \beta^*) \geq \ln \hat{f}(x_{it}, \hat{\beta}_{(g)})$.
   - **Quadratic Programming (QP):** Similar to LP, but minimizes the sum of squared deviations between the metafrontier and the underlying group frontiers.
   - **Stochastic Metafrontier (Huang et al., 2014):** Formulates the relationship between the metafrontier and the group frontier stochastically. Because the true group frontiers are unobservable, Huang et al. replace them with their estimates:
   $$\ln \hat{f}(x_{it}, \hat{\beta}_{(g)}) = \ln f(x_{it}, \beta^*) + V_{it} - U_{it}$$
   where V_{it} absorbs the statistical noise mapping from the first-stage estimation (representing the estimation error of the group frontier), and $U_{it} \geq 0$ captures the technology gap between the group frontier and the global metafrontier. This method relies directly on estimating a second-stage stochastic frontier using the group-specific fitted values $\ln \hat{y}_{it}$ as the dependent variable.
   - **Stochastic Metafrontier (O'Donnell et al., 2008):** Derives a stochastic metafrontier by fundamentally fitting an SFA over the pre-computed, deterministically enveloped values across all observations. Specifically, the dependent variable in this secondary stage relies on the theoretical values extracted via mathematical programming.
3. **Efficiencies and Metatechnology Ratios (MTR):**
   - Group-specific Technical Efficiency (TE_group): $TE^g_{it} = exp(-u_{it})$.
   - Metafrontier Technical Efficiency (TE_meta): TE^*_{it} = $exp(-u_{it} - U_{it})$.
   - Metatechnology Ratio (MTR): The ratio mapping the distance from the group frontier to the potential metafrontier: $MTR_{it} = \frac{TE^*_{it}}{TE^g_{it}} = exp(-U_{it})$.

When `groupType = "sfaselectioncross"`, only the selected observations (`selectDum == 1`) participate in the metafrontier; the MTR evaluates as `NA` for non-selected observations. When `groupType = "sfalcmcross"`, the best-posterior-class fitted value $y^{c^*}_{it}$ is routed to the metafrontier algorithm.

Value

sfametafrontier returns a list of class 'sfametafrontier' containing the following elements:

- `groupModels` A list containing the fitted frontier models for each group (class 'sfacross', 'sfaselectioncross', or 'sfalcmcross').
- `metaSfaObj` The fitted metafrontier model object. If metaMethod = "sfa", this is an object of class 'sfacross' representing the second-stage stochastic frontier. If metaMethod = "lp" or "qp", this contains optimization statistics and coefficients from the deterministic envelope.
- `metaRes` A matrix or data frame of the estimated metafrontier coefficients, standard errors, z-values, and p-values.
- `efficiencies` A list containing efficiency estimates, including group-specific technical efficiencies (TE_group), metafrontier technical efficiencies (TE_meta), and the metatechnology ratios (MTR).
- `formula` The formula used for the frontier.
- `metaMethod` The metafrontier estimation method used ("lp", "qp", or "sfa").
- `sfaApproach` The SFA approach used in the second stage ("huang" or "ordonnell").
- `groupType` The type of group-level models estimated ("sfacross", "sfaselectioncross", or "sfalcmcross").
- `groups` A character vector containing the names of the unique groups identified in the data.

References

Battese, G. E., Rao, D. S. P., and O'Donnell, C. J. 2004. A metafrontier production function for estimation of technical efficiencies and technology gaps for firms operating under different technologies. *Journal of Productivity Analysis*, **21**(1), 91–103. https://doi.org/10.1023/B:PROD.0000012454.06094.29

Huang, C. J., Huang, T.-H., and Liu, N.-H. 2014. A new approach to estimating the metafrontier production function based on a stochastic frontier framework. *Journal of Productivity Analysis*, **42**(3), 241–254. https://doi.org/10.1007/s11123-014-0402-2

O'Donnell, C. J., Rao, D. S. P., and Battese, G. E. 2008. Metafrontier frameworks for the study of firm-level efficiencies and technology ratios. *Empirical Economics*, **34**(2), 231–255. https://doi.org/10.1007/s00181-007-0119-4

See Also

sfacross, sfaselectioncross, sfalcmcross

Examples

```
## Not run:
## 1. Standard SFA Metafrontier Models
data("ricephil", package = "sfaR")
ricephil$group <- cut(ricephil$AREA,
  breaks = quantile(ricephil$AREA, probs = c(0, 1 / 3, 2 / 3, 1), na.rm = TRUE),
  labels = c("small", "medium", "large"),
  include.lowest = TRUE
)

# Linear Programming (LP) Metafrontier
```

```
meta_lp <- sfametafrontier(
  formula = log(PROD) ~ log(AREA) + log(LABOR) + log(NPK),
  data = ricephil, group = "group", S = 1, udist = "hnormal",
  metaMethod = "lp"
)
summary(meta_lp)

# Quadratic Programming (QP) Metafrontier
meta_qp <- sfametafrontier(
  formula = log(PROD) ~ log(AREA) + log(LABOR) + log(NPK),
  data = ricephil, group = "group", S = 1, udist = "hnormal",
  metaMethod = "qp"
)

# Huang (2014) Two-stage SFA Metafrontier
meta_huang <- sfametafrontier(
  formula = log(PROD) ~ log(AREA) + log(LABOR) + log(NPK),
  data = ricephil, group = "group", S = 1, udist = "hnormal",
  metaMethod = "sfa", sfaApproach = "huang"
)

# O'Donnell (2008) Stochastic Metafrontier on LP base
data("utility", package = "sfaR")
meta_ordonnell <- sfametafrontier(
  formula = log(tc / wf) ~ log(y) + I(1 / 2 * (log(y))^2) +
    log(wl / wf) + log(wk / wf) + I(1 / 2 * (log(wl / wf))^2) + I(1 / 2 * (log(wk / wf))^2) +
     I(log(wl / wf) * log(wk / wf)) + I(log(y) * log(wl / wf)) + I(log(y) * log(wk / wf)),
  data = utility, group = "regu", S = -1, udist = "hnormal",
  metaMethod = "sfa", sfaApproach = "ordonnell"
)

## 2. Latent Class Metafrontier (LCM) Models - Unobserved Groups
# LP Metafrontier using 2 posterior classes built automatically
meta_lcm_lp <- sfametafrontier(
  formula = log(tc / wf) ~ log(y) + I(1 / 2 * (log(y))^2) +
    log(wl / wf) + log(wk / wf) + I(1 / 2 * (log(wl / wf))^2) + I(1 / 2 * (log(wk / wf))^2) +
     I(log(wl / wf) * log(wk / wf)) + I(log(y) * log(wl / wf)) + I(log(y) * log(wk / wf)),
  data = utility, S = -1, groupType = "sfalcmcross", lcmClasses = 2,
  metaMethod = "lp"
)
summary(meta_lcm_lp)

# Huang (2014) stochastic metafrontier on LCM classes
meta_lcm_huang <- sfametafrontier(
  formula = log(tc / wf) ~ log(y) + I(1 / 2 * (log(y))^2) +
    log(wl / wf) + log(wk / wf) + I(1 / 2 * (log(wl / wf))^2) + I(1 / 2 * (log(wk / wf))^2) +
     I(log(wl / wf) * log(wk / wf)) + I(log(y) * log(wl / wf)) + I(log(y) * log(wk / wf)),
  data = utility, S = -1, groupType = "sfalcmcross", lcmClasses = 2,
  metaMethod = "sfa", sfaApproach = "huang"
)

## 3. Sample Selection Metafrontier Models
ricephil$laterSurvey <- as.integer(ricephil$YEARDUM > 3)
# LP Metafrontier using generalized sample selection handling bias
meta_sel_lp <- sfametafrontier(
  formula = log(PROD) ~ log(AREA) + log(LABOR) + log(NPK),
  data = ricephil, group = "group", S = 1, groupType = "sfaselectioncross",
```

```
  selectionF = laterSurvey ~ EDYRS + AGE, metaMethod = "lp"
)
summary(meta_sel_lp)

## End(Not run)
```

---

summary *Summary of results for stochastic metafrontier models*

Description

Create and print summary results for stochastic metafrontier models returned by `sfametafrontier`.

Usage

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

## S3 method for class 'summary.sfametafrontier'
print(x, digits = max(3, getOption("digits") - 2), ...)
```

Arguments

`object`	An object of class `'sfametafrontier'` returned by the function `sfametafrontier`.
`...`	Currently ignored.
`x`	An object of class `'summary.sfametafrontier'`.
`digits`	Numeric. Number of digits displayed in values.

Value

The `summary` method returns a list of class `'summary.sfametafrontier'` that contains the same elements as an object returned by `sfametafrontier` with the following additional elements:

`AIC`	Akaike information criterion.
`BIC`	Bayesian information criterion.
`HQIC`	Hannan-Quinn information criterion.
`metaRes`	Matrix of metafrontier estimates, their standard errors, z-values, and asymptotic P-values.
`effStats`	A list of efficiency statistics including group means and class membership probabilities.
`grpSummaries`	A list of summary objects for each group model.

See Also

`sfametafrontier`, for the stochastic metafrontier analysis model fitting function for cross-sectional or pooled data.

vcov *Compute variance-covariance matrix of stochastic metafrontier models*

Description

vcov computes the variance-covariance matrix of the maximum likelihood (ML) coefficients from stochastic metafrontier models estimated with sfametafrontier.

Usage

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

Arguments

object	A stochastic metafrontier model returned by sfametafrontier.
...	Currently ignored

Details

The variance-covariance matrix is obtained by the inversion of the negative Hessian matrix. Depending on the distribution and the 'hessianType' option, the analytical/numeric Hessian or the bhhh Hessian is evaluated.

Value

The variance-covariance matrix of the maximum likelihood coefficients is returned.

See Also

sfametafrontier, for the stochastic metafrontier analysis model fitting function using cross-sectional or pooled data.

Index