Package ‘sfaR’

October 14, 2022

Version 0.1.1

Date 2022-05-04

Title Stochastic Frontier Analysis using R

Maintainer Yann Desjeux <yann.desjeux@inrae.fr>

Description Maximum likelihood estimation for stochastic frontier analysis (SFA) of production (profit) and cost functions.
The package includes several distributions for the one-sided error term (i.e. Rayleigh, Gamma, Weibull, lognormal, uniform, generalized exponential and truncated skewed Laplace) as well as the latent class stochastic frontier model (LCM) as described in Dakpo et al. (2021) <doi:10.1111/1477-9552.12422>.
Several possibilities in terms of optimization algorithms are proposed.

Depends R (>= 3.5.0)

BugReports https://r-forge.r-project.org/tracker/?group_id=2413

Imports dplyr, emdbook, fBasics, Formula, gsl, marqLevAlg, MASS, maxLik, methods, moments, nleqslv, numDeriv, primes, qrng, randtoolbox, trustOptim, ucminf

Suggests mlogit

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation no

Author K Hervé Dakpo [aut],
Yann Desjeux [aut, cre],
Laure Latruffe [aut]

Repository CRAN

Date/Publication 2022-05-05 14:40:02 UTC

R topics documented:

`sfaR-package` *sfaR: An R package for estimating stochastic frontier models*

Description

The **sfaR** package provides a set of tools (maximum likelihood - ML and maximum simulated likelihood - MSL) for various specifications of stochastic frontier analysis (SFA).

Two categories of important functions are available: `sfacross` and `lcmcross`, which estimate different types of frontiers and offer nine alternative optimization algorithms (i.e. "bfgs", "bhhh", "nr", "nm", "ucminf", "mla", "sr1", "sparse" and "nlminb").

lcmcross

`lcmcross` estimates latent class stochastic frontier models (LCM), which accounts for technological heterogeneity by splitting the observations into a maximum number of five classes. The classification operates based on a logit functional form that can be specified using some covariates (namely, the separating variables allowing the separation of observations in several classes). Only the half normal distribution is available for the one-sided error term. Heteroscedasticity in both error terms is possible. The choice of the number of classes can be guided by several information criteria (i.e. AIC, BIC or HQIC).

sfacross

sfacross estimates the frontier for cross-sectional data and allows for ten different distributions for the one-sided error term. These distributions include the exponential, the Gamma, the generalized exponential, the half normal, the lognormal, the truncated normal, the truncated skewed Laplace, the Rayleigh, the uniform and the Weibull distributions. In the case of the Gamma, lognormal and Weibull distributions, maximum simulated likelihood (MSL) is used with the possibility of four specific distributions to construct the draws: Halton, Generalized Halton, Sobol and uniform. Heteroscedasticity in both error terms can be implemented, in addition to heterogeneity in the truncated mean parameter in the case of the truncated normal and lognormal distributions. In addition, in the case of the truncated normal distribution, the scaling property can be estimated.

Bugreport

Any bug or suggestion can be reported using the sfaR's tracker facilities at: https://r-forge.r-project.org/projects/sfar/

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

---

coef *Extract coefficients of classic or latent class stochastic models*

Description

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

From on object of class 'sfacross' or 'lcmcross', it extracts only the estimated coefficients.

Usage

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

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

## S3 method for class 'sfacross'
coef(object, extraPar = FALSE, ...)

## S3 method for class 'lcmcross'
coef(object, extraPar = FALSE, ...)
```

Arguments

object — A classic or latent class stochastic frontier model returned by `sfacross` or `lcmcross`, or an object of class `'summary.sfacross'` or `'summary.lcmcross'`.

extraPar — Logical (default = `FALSE`). Only applies to objects of class `'sfacross'` or `'lcmcross'`. If `TRUE`, additional parameters are returned:

`sigmaSq` = `sigmauSq` + `sigmavSq`

`lambdaSq` = `sigmauSq`/`sigmavSq`

`sigmauSq` = $\exp{(Wu)}$ = $\exp{(\delta Z_u)}$

`sigmavSq` = $\exp{(Wv)}$ = $\exp{(\phi Z_v)}$

`sigma` = `sigmaSq`^0.5

`lambda` = `lambdaSq`^0.5

`sigmau` = `sigmauSq`^0.5

`sigmav` = `sigmavSq`^0.5

`gamma` = `sigmauSq`/(`sigmauSq` + `sigmavSq`)

... — Currently ignored.

Value

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

For objects of class `'sfacross'` or `'lcmcross'`, `coef` returns a numeric vector of the estimated coefficients. If `extraPar = TRUE`, additional parameters, detailed in the section 'Arguments', are also returned. In the case of object of class `'lcmcross'`, each additional parameter terminates with `"#"` that represents the class number.

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

See Also

`sfacross`, for the stochastic frontier analysis model fitting function.

`lcmcross`, for the latent class stochastic frontier analysis model fitting function.

Examples

```
## Using data on fossil fuel fired steam electric power generation plants in the U.S.
# Translog SFA (cost function) truncated normal with scaling property
tl_u_ts <- sfacross(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)),
    udist = "tnormal", muhet = ~ regu, uhet = ~ regu, data = utility, S = -1,
    scaling = TRUE, method = "mla")
  coef(tl_u_ts, extraPar = TRUE)
  coef(summary(tl_u_ts))
```

dairyspain *Data on Spanish Dairy Farms*

Description

This dataset contains six years of observations on 247 dairy farms in northern Spain, drawn from 1993-1998. The original data consist in the farm and year identifications, plus measurements on one output (i.e. milk), and four inputs (i.e. cows, land, labor and feed).

Usage

```
dairyspain
```

Format

A data frame with 1,482 observations on the following 29 variables.

FARM Farm identification.

AGEF Age of the farmer.

YEAR Year identification.

COWS Number of milking cows.

LAND Agricultural area.

MILK Milk production.

LABOR Labor.

FEED Feed.

YIT Log of `MILK`.

X1 Log of `COWS`.

X2 Log of `LAND`.

X3 Log of `LABOR`.

X4 Log of `FEED`.

X11 1/2 * `X1^2`.

X22 1/2 * `X2^2`.

X33 1/2 * `X3^2`.

X44 1/2 * `X4^2`.

X12 `X1` * `X2`.

X13 `X1` * `X3`.

X14 `X1` * `X4`.

X23 `X2` * `X3`.

X24 `X2` * `X4`.

X34 `X3` * `X4`.

YEAR93 Dummy for YEAR = 1993.

YEAR94 Dummy for YEAR = 1994.

YEAR95 Dummy for YEAR = 1995.

YEAR96 Dummy for YEAR = 1996.

YEAR97 Dummy for YEAR = 1997.

YEAR98 Dummy for YEAR = 1998.

Details

This dataset has been used in Alvarez *et al.* (2004). The data have been normalized so that the logs of the inputs sum to zero over the 1,482 observations.

Source

http://pages.stern.nyu.edu/~wgreene/Econometrics/oldPanelDataSets.htm

References

Alvarez, A., C. Arias, and W. Greene. 2004. Accounting for unobservables in production models: management and inefficiency. *Econometric Society*, **341**:1–20.

Examples

```
str(dairyspain)
summary(dairyspain)
```

---

efficiencies — *Compute conditional (in-)efficiency estimates of classic or latent class stochastic models*

Description

efficiencies returns (in-)efficiency estimates from classic or latent class stochastic frontier models estimated with sfacross or lcmcross.

Usage

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

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

Arguments

`object`	A classic or latent class stochastic frontier model returned by `sfacross` or `lcmcross`.
`level`	A number between between 0 and 0.9999 used for the computation of (in-)efficiency confidence intervals (defaut = `0.95`). Only used when `udist = "hnormal"`, `"exponential"`, `"tnormal"` or `"uniform"` in `sfacross` or `lcmcross`.
`...`	Currently ignored.

Details

The conditional inefficiency is obtained following Jondrow *et al.* (1982) and the conditional efficiency is computed following Battese and Coelli (1988). In some cases the conditional mode is also returned (Jondrow *et al.* 1982). The confidence interval is computed following Horrace and Schmidt (1996), Hjalmarsson *et al.* (1996), or Berra and Sharma (1999) (see 'Value' section).

In the case of the half normal distribution for the one-sided error term, the formulae are as follows (for notations, see the 'Details' section of `sfacross` or `lcmcross`):

- The conditional inefficiency is

$$E\left[u_i|\epsilon_i\right] = \mu_{i*} + \sigma_* \frac{\phi\left(\frac{\mu_{i*}}{\sigma_*}\right)}{\Phi\left(\frac{\mu_{i*}}{\sigma_*}\right)}$$

where

$$\mu_{i*} = \frac{-S\epsilon_i\sigma_u^2}{\sigma_u^2 + \sigma_v^2}$$

and

$$\sigma_*^2 = \frac{\sigma_u^2\sigma_v^2}{\sigma_u^2 + \sigma_v^2}$$

- The Battese and Coelli (1988) conditional efficiency is obtained by:

$$E\left[\exp\left(-u_i\right)|\epsilon_i\right] = \exp\left(-\mu_{i*} + \frac{1}{2}\sigma_*^2\right)\frac{\Phi\left(\frac{\mu_{i*}}{\sigma_*} - \sigma_*\right)}{\Phi\left(\frac{\mu_{i*}}{\sigma_*}\right)}$$

- The conditional mode is computed using:

$$M\left[u_i|\epsilon_i\right] = \mu_{i*} \quad For \quad \mu_{i*} > 0$$

and

$$M\left[u_i|\epsilon_i\right] = 0 \quad For \quad \mu_{i*} \leq 0$$

- The confidence intervals are obtained with:

$$\mu_{i*} + I_L\sigma_* \leq E\left[u_i|\epsilon_i\right] \leq \mu_{i*} + I_U\sigma_*$$

with $LB_i = \mu_{i*} + I_L\sigma_*$ and $UB_i = \mu_{i*} + I_U\sigma_*$

and

$$I_L = \Phi^{-1}\left\{1 - \left(1 - \frac{\alpha}{2}\right)\left[1 - \Phi\left(-\frac{\mu_{i*}}{\sigma_*}\right)\right]\right\}$$

and

$$I_U = \Phi^{-1}\left\{1 - \frac{\alpha}{2}\left[1 - \Phi\left(-\frac{\mu_{i*}}{\sigma_*}\right)\right]\right\}$$

Thus

$$\exp\left(-UB_i\right) \leq E\left[\exp\left(-u_i\right)|\epsilon_i\right] \leq \exp\left(-LB_i\right)$$

Value

A data frame that contains individual (in-)efficiency estimates. These are ordered in the same way as the corresponding observations in the dataset used for the estimation.

- For object of class `'sfacross'` the following elements are returned:

`u` Conditional inefficiency. In the case argument `udist` of sfacross is set to `"uniform"`, two conditional inefficiency estimates are returned: `u1` for the classic conditional inefficiency following Jondrow *et al.* (1982), and `u2` which is obtained when $\theta/\sigma_v \longrightarrow \infty$ (see Nguyen, 2010).

`uLB` Lower bound for conditional inefficiency. Only when the argument `udist` of sfacross is set to `"hnormal"`, `"exponential"`, `"tnormal"` or `"uniform"`.

`uUB` Upper bound for conditional inefficiency. Only when the argument `udist` of sfacross is set to `"hnormal"`, `"exponential"`, `"tnormal"` or `"uniform"`.

`teJLMS` $\exp\left(-u\right)$. When the argument `udist` of sfacross is set to `"uniform"`, `teJLMS1` = $\exp\left(-u1\right)$ and `teJLMS2` = $\exp\left(-u2\right)$. Only when `logDepVar = TRUE`.

`m` Conditional model. Only when the argument `udist` of sfacross is set to `"hnormal"`, `"exponential"`, `"tnormal"`, or `"rayleigh"`.

`teMO` $\exp\left(-m\right)$. Only when, in the function sfacross, `logDepVar = TRUE` and `udist = "hnormal"`, `"exponential"`, `"tnormal"`, `"uniform"`, or `"rayleigh"`.

`teBC` Battese and Coelli (1988) conditional efficiency. Only when, in the function sfacross, `logDepVar = TRUE` and `udist = "hnormal"`, `"exponential"`, `"tnormal"`, `"genexponential"`, `"rayleigh"`, or `"tslaplace"`. In the case `udist = "uniform"`, two conditional efficiency estimates are returned: `teBC1` which is the classic conditional efficiency following Battese and Coelli (1988) and `teBC2` when $\theta/\sigma_v \longrightarrow \infty$ (see Nguyen, 2010).

teBCLB — Lower bound for Battese and Coelli (1988) conditional efficiency. Only when, in the function sfacross, logDepVar = TRUE and udist = "hnormal", "exponential", "tnormal", or "uniform".

teBCUB — Upper bound for Battese and Coelli (1988) conditional efficiency. Only when, in the function sfacross, logDepVar = TRUE and udist = "hnormal", "exponential", "tnormal", or "uniform".

- For object of class 'lcmcross' the following elements are returned:

Group_c — Most probable class of each observation.

PosteriorProb_c — Highest posterior probability.

PosteriorProb_c# — Posterior probability associated to class #, regardless of Group_c.

PriorProb_c# — Prior probability associated to class #, regardless of Group_c.

u_c — Conditional inefficiency of the most probable class given the posterior probability.

teJLMS_c — $\exp{(-u_c)}$. Only when, in the function lcmcross, logDepVar = TRUE.

u_c# — Conditional inefficiency associated to class #, regardless of Group_c.

ineff_c# — Conditional inefficiency (u_c) for observations in class # only.

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

References

Battese, G.E., and T.J. Coelli. 1988. Prediction of firm-level technical efficiencies with a generalized frontier production function and panel data. *Journal of Econometrics*, **38**:387–399.

Bera, A.K., and S.C. Sharma. 1999. Estimating production uncertainty in stochastic frontier production function models. *Journal of Productivity Analysis*, **12**:187-210.

Hjalmarsson, L., S.C. Kumbhakar, and A. Heshmati. 1996. DEA, DFA and SFA: A comparison. *Journal of Productivity Analysis*, **7**:303-327.

Horrace, W.C., and P. Schmidt. 1996. Confidence statements for efficiency estimates from stochastic frontier models. *Journal of Productivity Analysis*, **7**:257-282.

Jondrow, J., C.A.K. Lovell, I.S. Materov, and P. Schmidt. 1982. On the estimation of technical inefficiency in the stochastic frontier production function model. *Journal of Econometrics*, **19**:233–238.

Nguyen, N.B. 2010. Estimation of technical efficiency in stochastic frontier analysis. PhD Dissertation, Bowling Green State University, August.

See Also

sfacross, for the stochastic frontier analysis model fitting function.

lcmcross, for the latent class stochastic frontier analysis model fitting function.

Examples

```
## Using data on fossil fuel fired steam electric power generation plants in the U.S.
# Translog SFA (cost function) truncated normal with scaling property
tl_u_ts <- sfacross(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)), udist = "tnormal",
    muhet = ~ regu, uhet = ~ regu, data = utility, S = -1, scaling = TRUE, method = "mla")
  eff.tl_u_ts <- efficiencies(tl_u_ts)
  head(eff.tl_u_ts)
  summary(eff.tl_u_ts)

cb_2c_h1 <- lcmcross(formula = ly ~ lk + ll + yr, thet = ~initStat, data = worldprod)
  eff.ccb_2c_h1 <- efficiencies(cb_2c_h1)
  table(eff.ccb_2c_h1$Group_c)
  summary(eff.ccb_2c_h1[,c("ineff_c1", "ineff_c2")])
 summary(eff.ccb_2c_h1[eff.ccb_2c_h1$Group_c == 1, c("PosteriorProb_c1", "PriorProb_c1")])
 summary(eff.ccb_2c_h1[eff.ccb_2c_h1$Group_c == 2, c("PosteriorProb_c2", "PriorProb_c2")])
```

---

electricity — *Data on U.S. electric power generation*

Description

This dataset is on electric power generation in the United States.

Usage

```
electricity
```

Format

A data frame with 123 observations on the following 9 variables.

firm Firm identification.

cost Total cost in 1970, MM USD.

output Output in million KwH.

lprice Labor price.

lshare Labor's cost share.

cprice Capital price.

cshare Capital's cost share.

fprice Fuel price.

fshare Fuel's cost share.

Details

The dataset is from Christensen and Greene (1976) and has also been used in Greene (1990).

Source

http://pages.stern.nyu.edu/~wgreene/Text/tables/tablelist5.htm

References

Christensen, L.R., and W.H. Greene. 1976. Economies of scale in US electric power generation. *The Journal of Political Economy*, **84**:655–676.

Greene, W.H. 1990. A Gamma-distributed stochastic frontier model. *Journal of Econometrics*, **46**:141–163.

Examples

```
str(electricity)
summary(electricity)
```

---

fitted — *Extract fitted frontier values of classic or latent class stochastic models*

Description

`fitted` returns the fitted frontier values from classic or latent class stochastic frontier models estimated with `sfacross` or `lcmcross`.

Usage

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

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

Arguments

`object`	A classic or latent class stochastic frontier model returned by `sfacross` or `lcmcross`.
`...`	Currently ignored.

Value

In the case of an object of class `'sfacross'`, a vector of fitted values is returned.

In the case of an object of class `'lcmcross'`, a data frame containing the fitted values for each class is returned where each variable terminates with `"_c#"`, `"#"` being the class number.

Note

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

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

See Also

sfacross, for the stochastic frontier analysis model fitting function.

lcmcross, for the latent class stochastic frontier analysis model fitting function.

Examples

```
## Using data on eighty-two countries production (DGP)
# LCM Cobb Douglas (production function) half normal distribution
cb_2c_h <- lcmcross(formula = ly ~ lk + ll + yr, udist = 'hnormal', data = worldprod)
  fit.cb_2c_h <- fitted(cb_2c_h)
  head(fit.cb_2c_h)
```

---

ic — *Extract information criteria of classic or latent class stochastic models*

Description

ic returns information criterion from classic or latent class stochastic frontier models estimated with sfacross or lcmcross.

Usage

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

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

Arguments

object	A classic or latent class stochastic frontier model returned by sfacross or lcmcross.
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.

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

See Also

sfacross, for the stochastic frontier analysis model fitting function.

lcmcross, for the latent class stochastic frontier analysis model fitting function.

Examples

```
## Using data on Swiss railway
# LCM (cost function) half normal distribution
cb_2c_u <- lcmcross(formula = LNCT ~ LNQ2 + LNQ3 + LNNET + LNPK + LNPL,
    udist = "hnormal", uhet = ~ 1, data = swissrailways, S = -1, method="ucminf")
  ic(cb_2c_u)
  ic(cb_2c_u, IC = "BIC")
  ic(cb_2c_u, IC = "HQIC")
```

---

lcmcross *Latent class stochastic frontier using cross-section data*

Description

lcmcross is a symbolic formula based function for the estimation of the latent class stochastic frontier model (LCM) in the case of cross-sectional or pooled cross-section data. The model is estimated using maximum likelihood (ML). See Orea and Kumbhakar (2004), Parmeter and Kumbhakar (2014, p282).

Only the half-normal distribution is possible for the one-sided error term. Nine optimization algorithms are available.

The function also accounts for heteroscedasticity in both one-sided and two-sided error terms, as in Reifschneider and Stevenson (1991), Caudill and Ford (1993), Caudill *et al.* (1995) and Hadri (1999).

The model can estimate up to five classes.

Usage

```
lcmcross(formula, uhet, vhet, thet, logDepVar = TRUE, data, subset, S = 1,
  udist = "hnormal", start = NULL, lcmClasses = 2, method = "bfgs", hessianType = 1,
  itermax = 2000, printInfo = FALSE, tol = 1e-12, gradtol = 1e-06, stepmax = 0.1,
  qac = "marquardt", initStart = FALSE, initAlg = "nlminb", initIter = 100,
  initFactorLB = 0.5, initFactorUB = 1.5)
```

Arguments

formula	A symbolic description of the model to be estimated based on the generic function formula (see section 'Details').
uhet	A one-part formula to account for heteroscedasticity in the one-sided error variance (see section 'Details').
vhet	A one-part formula to account for heteroscedasticity in the two-sided error variance (see section 'Details').
thet	A one-part formula to account for technological heterogeneity in the construction of the classes.
logDepVar	Logical. Informs whether the dependent variable is logged (TRUE) or not (FALSE). Default = TRUE.
data	The data frame containing the data.
subset	An optional vector specifying a subset of observations to be used in the optimization process.
S	If S = 1 (default), a production (profit) frontier is estimated: $\epsilon_i = v_i - u_i$. If S = -1, a cost frontier is estimated: $\epsilon_i = v_i + u_i$.
udist	Character string. Distribution specification for the one-sided error term. Only the half normal distribution "hnormal" (Aigner *et al.*, 1977, Meeusen and Vandenbroeck, 1977) is currently implemented.
start	Numeric vector. Optional starting values for the maximum likelihood (ML) estimation.
lcmClasses	Number of classes to be estimated (default = 2). A maximum of five classes can be estimated.
method	Optimization algorithm used for the estimation. Default = "bfgs". 9 algorithms are available:

- "bfgs", for Broyden-Fletcher-Goldfarb-Shanno (see maxBFGS)
- "bhhh", for Berndt-Hall-Hall-Hausman (see maxBHHH)
- "nr", for Newton-Raphson (see maxNR)
- "nm", for Nelder-Mead (see maxNM)
- "ucminf", implements a quasi-Newton type with BFGS updating of the inverse Hessian and soft line search with a trust region type monitoring of the input to the line search algorithm (see ucminf)
- "mla", for general-purpose optimization based on Marquardt-Levenberg algorithm (see mla)
- "sr1", for Symmetric Rank 1 (see trust.optim)

- "sparse", for trust regions and sparse Hessian (see trust.optim)
- "nlminb", for optimization using PORT routines (see nlminb)

hessianType
: Integer. If 1 (default), analytic Hessian is returned for all the distributions except "gamma", "lognormal" and "weibull" for which the numeric Hessian is returned. If 2, bhhh Hessian is estimated ($g'g$). If 3, robust Hessian is computed ($H^{-1}GH^{-1}$).

itermax
: Maximum number of iterations allowed for optimization. Default = 2000.

printInfo
: Logical. Print information during optimization. Default = FALSE.

tol
: Numeric. Convergence tolerance. Default = 1e-12.

gradtol
: Numeric. Convergence tolerance for gradient. Default = 1e-06.

stepmax
: Numeric. Step max for ucminf algorithm. Default = 0.1.

qac
: Character. Quadratic Approximation Correction for "bhhh" and "nr" algorithms. If "qac = stephalving", the step length is decreased but the direction is kept. If "qac = marquardt" (default), the step length is decreased while also moving closer to the pure gradient direction. See maxBHHH and maxNR.

initStart
: Logical. If TRUE, the model is jump-started using an alternative algorithm ("nlminb") within certain bounds. Default = FALSE.

initAlg
: Character. Algorithm used to jump-start the latent class model. Only "nlminb" is currently available.

initIter
: Maximum number of iterations for the algorihtm when initStart = TRUE. Default = 100.

initFactorLB
: A numeric value indicating by which factor the starting value should be multiplied to define the lower bounds for the jump-start algorithm. Default = 0.5.

initFactorUB
: A numeric value indicating by which factor the starting value should be multiplied to define the upper bounds for the jump-start algorithm. Default = 1.5.

Details

LCM is an estimation of a finite mixture of production functions:

$$y_i = \alpha_j + x_i'\beta_j + v_{i|j} - Su_{i|j}$$

$$\epsilon_{i|j} = v_{i|j} - Su_{i|j}$$

where i is the observation, j is the class, y is the output (cost, revenue, profit), x is the vector of main explanatory variables (inputs and other control variables), u is the one-sided error term with variance σ_u^2, and v is the two-sided error term with variance σ_v^2.

S = 1 in the case of production (profit) frontier function and S = -1 in the case of cost frontier function.

The contribution of observation i to the likelihood conditional on class j is defined as:

$$P(i|j) = \frac{2}{\sqrt{\sigma_{u|j}^2 + \sigma_{v|j}^2}} \phi\left(\frac{S\epsilon_{i|j}}{\sqrt{\sigma_{u|j}^2 + \sigma_{v|j}^2}}\right) \Phi\left(\frac{\mu_{i*|j}}{\sigma_{*|j}}\right)$$

where

$$\mu_{i*|j} = \frac{-S\epsilon_{i|j}\sigma_{u|j}^2}{\sigma_{u|j}^2 + \sigma_{v|j}^2}$$

and

$$\sigma_*^2 = \frac{\sigma_{u|j}^2\sigma_{v|j}^2}{\sigma_{u|j}^2 + \sigma_{v|j}^2}$$

The prior probability of using a particular technology can depend on some covariates (namely the variables separating the observations into classes) using a logit specification:

$$\pi(i,j) = \frac{\exp{(\theta_j' Z_h)}}{\sum_{m=1}^J \exp{(\theta_m' Z_h)}}$$

with Z_h the covariates, θ the coefficients estimated for the covariates, and $\exp(\theta'_J Z_h) = 1$.

The unconditional likelihood of observation i is simply the average over the J classes:

$P(i) = \sum_{m=1}^J \pi(i,m)P(i|m)$

The number of classes can be retained based on information criterion (see for instance `ic`).

Class assignment is based on the largest posterior probability. This probability is obtained using Bayes' rule, as follows for class j:

$$w\left(j|i\right) = \frac{P\left(i|j\right)\pi\left(i,j\right)}{\sum_{m=1}^J P\left(i|m\right)\pi\left(i,m\right)}$$

To accommodate heteroscedasticity in the variance parameters of the error terms, a single part (right) formula can also be specified. To impose the positivity on these parameters, the variances are modelled respectively as: $\sigma^2_{u|j} = \exp{(\delta_j' Z_u)}$ and $\sigma^2_{v|j} = \exp{(\phi_j' Z_v)}$, where Z_u and Z_v are the heteroscedasticity variables (inefficiency drivers in the case of Z_u) and δ and ϕ the coefficients. In the case of heterogeneity in the truncated mean μ, it is modelled as $\mu = \omega' Z_\mu$.

Value

`lcmcross` returns a list of class `'lcmcross'` containing the following elements:

`call`	The matched call.
`formula`	Multi parts formula describing the estimated model.
`S`	The argument `'S'`. See the section 'Arguments'.
`typeSfa`	Character string. "Latent Class Production/Profit Frontier, e = v - u" when `S = 1` and "Latent Class Cost Frontier, e = v + u" when `S = -1`.
`Nobs`	Number of observations used for optimization.
`nXvar`	Number of main explanatory variables.
`nZHvar`	Number of variables in the logit specification of the finite mixture model (i.e. number of covariates).
`logDepVar`	The argument `'logDepVar'`. See the section 'Arguments'.
`nuZUvar`	Number of variables explaining heteroscedasticity in the one-sided error term.

nvZVvar	Number of variables explaining heteroscedasticity in the two-sided error term.
nParm	Total number of parameters estimated.
udist	The argument 'udist'. See the section 'Arguments'.
startVal	Numeric vector. Starting value for ML estimation.
dataTable	A data frame (tibble format) containing information on data used for optimization along with residuals and fitted values of the OLS and ML estimations, and the individual observation log-likelihood.
InitHalf	When start = NULL. Initial ML estimation with half normal distribution for the one-sided error term. Model to construct the starting values for the latent class estimation. Object of class 'maxLik' and 'maxim' returned.
optType	The optimization algorithm used.
nIter	Number of iterations of the ML estimation.
optStatus	An optimization algorithm termination message.
startLoglik	Log-likelihood at the starting values.
nClasses	The number of classes estimated.
mlLoglik	Log-likelihood value of the ML estimation.
mlParam	Numeric vector. Parameters obtained from ML estimation.
gradient	Numeric vector. Each variable gradient of the ML estimation.
gradL_OBS	Matrix. Each variable individual observation gradient of the ML estimation.
gradientNorm	Numeric. Gradient norm of the ML estimation.
invHessian	The covariance matrix of the parameters obtained from the ML estimation.
hessianType	The argument 'hessianType'. See the section 'Arguments'.
mlDate	Date and time of the estimated model.

Note

In the case of panel data, lcmcross estimates a pooled cross-section where the probability of belonging to a class a priori is not permanent (not fixed over time).

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

References

Aigner, D., Lovell, C. A. K., and P. Schmidt. 1977. Formulation and estimation of stochastic frontier production function models. *Journal of Econometrics*, **6**(1), 21–37.

Caudill, S. B., and J. M. Ford. 1993. Biases in frontier estimation due to heteroscedasticity. *Economics Letters*, **41**(1), 17–20.

Caudill, S. B., Ford, J. M., and D. M. Gropper. 1995. Frontier estimation and firm-specific inefficiency measures in the presence of heteroscedasticity. *Journal of Business & Economic Statistics*, **13**(1), 105–111.

Hadri, K. 1999. Estimation of a doubly heteroscedastic stochastic frontier cost function. *Journal of Business & Economic Statistics*, **17**(3), 359–363.

Meeusen, W., and J. Vandenbroeck. 1977. Efficiency estimation from Cobb-Douglas production functions with composed error. *International Economic Review*, **18**(2), 435–445.

Orea, L., and S.C. Kumbhakar. 2004. Efficiency measurement using a latent class stochastic frontier model. *Empirical Economics*, **29**, 169–183.

Parmeter, C.F., and S.C. Kumbhakar. 2014. Efficiency analysis: A primer on recent advances. *Foundations and Trends in Econometrics*, **7**, 191–385.

Reifschneider, D., and R. Stevenson. 1991. Systematic departures from the frontier: A framework for the analysis of firm inefficiency. *International Economic Review*, **32**(3), 715–723.

See Also

`summary` for creating and printing summary results.

`coef` for extracting coefficients of the estimation.

`efficiencies` for computing (in-)efficiency estimates.

`fitted` for extracting the fitted frontier values.

`ic` for extracting information criteria.

`logLik` for extracting log-likelihood value(s) of the estimation.

`marginal` for computing marginal effects of inefficiency drivers.

`residuals` for extracting residuals of the estimation.

`vcov` for computing the variance-covariance matrix of the coefficients.

Examples

```
## Using data on eighty-two countries production (DGP)
# LCM Cobb Douglas (production function) half normal distribution
# Intercept and initStat used as separating variables
cb_2c_h1 <- lcmcross(formula = ly ~ lk + ll + yr, thet = ~initStat, data = worldprod)
  summary(cb_2c_h1)

# summary of the initial ML model
  summary(cb_2c_h1$InitHalf)

# same result by jump-starting the estimation
cb_2c_h2 <- lcmcross(formula = ly ~ lk + ll + yr, data = worldprod, initStart = TRUE)
  summary(cb_2c_h2)

# Only the intercept is used as the separating variable and only variable
# initStat is used as inefficiency driver
cb_2c_h3 <- lcmcross(formula = ly ~ lk + ll + yr, uhet = ~initStat, data = worldprod)
  summary(cb_2c_h3)
```

logLik — *Extract log-likelihood value of classic or latent class stochastic models*

Description

logLik extracts the log-likelihood value(s) from classic or latent class stochastic frontier models estimated with sfacross or lcmcross.

Usage

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

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

Arguments

object	A classic or latent class stochastic frontier model returned by sfacross or lcmcross.
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 an object of class 'logLik', which is either a numeric matrix with the log-likelihood value (logLik), the total number of observations (Nobs) and the number of free parameters (df), 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.

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

See Also

sfacross, for the stochastic frontier analysis model fitting function.

lcmcross, for the latent class stochastic frontier analysis model fitting function.

Examples

```
## Using data on fossil fuel fired steam electric power generation plants in the U.S.
# Translog SFA (cost function) truncated normal with scaling property
tl_u_ts <- sfacross(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)),
    udist = "tnormal", muhet = ~ regu, uhet = ~ regu, data = utility, S = -1,
    scaling = TRUE, method = "mla")
  logLik(tl_u_ts)

## Using data on eighty-two countries production (DGP)
# LCM Cobb Douglas (production function) half normal distribution
cb_2c_h <- lcmcross(formula = ly ~ lk + ll + yr, udist = "hnormal", data = worldprod, S = 1)
  logLik(cb_2c_h, individual = TRUE)
```

---

marginal — *Compute marginal effects of the inefficiency drivers in classic or latent class stochastic models*

Description

This function returns marginal effects of the inefficiency drivers from classic or latent class stochastic frontier models estimated with sfacross or lcmcross.

Usage

```
## S3 method for class 'sfacross'
marginal(object, ...)

## S3 method for class 'lcmcross'
marginal(object, ...)
```

Arguments

object	A classic or latent class stochastic frontier model returned by sfacross or lcmcross.
...	Currently ignored.

Details

marginal operates in the presence of exogenous variables that explain inefficiency, namely the inefficiency drivers ($uhet = Z_u$ or $muhet = Z_{mu}$).

Two components are computed for each variable: the marginal effects on the expected inefficiency ($\frac{\partial E[u]}{\partial Z_{(m)u}}$) and the marginal effects on the variance of inefficiency ($\frac{\partial V[u]}{\partial Z_{(m)u}}$).

The model also allows the Wang (2002) parametrization of μ and σ_u^2 by the same vector of exogenous variables. This double parameterization accounts for non-monotonic relationships between the inefficiency and its drivers.

Value

marginal returns a data frame containing the marginal effects of the Z_u variables on the expected inefficiency (each variable has the prefix "Eu_") and on the variance of the inefficiency (each variable has the prefix "Vu_") is returned.

In the case of the latent class model (LCM), each variable terminates with "_c#" where "#" is the class number.

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

References

Wang, H.J. 2002. Heteroscedasticity and non-monotonic efficiency effects of a stochastic frontier model. *Journal of Productivity Analysis*, **18**:241–253.

See Also

sfacross, for the stochastic frontier analysis model fitting function.

lcmcross, for the latent class stochastic frontier analysis model fitting function.

Examples

```
## Using data on fossil fuel fired steam electric power generation plants in the U.S.
# Translog SFA (cost function) truncated normal with scaling property
tl_u_ts <- sfacross(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)),
    udist = "tnormal", muhet = ~ regu + wl, uhet = ~ regu + wl, data = utility, S = -1,
    scaling = TRUE, method = "mla")
  marg.tl_u_ts <- marginal(tl_u_ts)
  summary(marg.tl_u_ts)

## Using data on eighty-two countries production (DGP)
# LCM Cobb Douglas (production function) half normal distribution
cb_2c_h <- lcmcross(formula = ly ~ lk + ll + yr, udist = "hnormal",
    data = worldprod, uhet = ~ initStat + h, S = 1, method = "mla")
  marg.cb_2c_h <- marginal(cb_2c_h)
  summary(marg.cb_2c_h)
```

---

residuals *Extract residuals of classic or latent class stochastic models*

Description

This function returns the residuals' values from classic or latent class stochastic frontier models estimated with sfacross or lcmcross.

Usage

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

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

Arguments

`object`	A classic or latent class stochastic frontier model returned by `sfacross` or `lcmcross`.
`...`	Currently ignored.

Value

When the `object` is of class `'sfacross'`, `residuals` returns a vector of residuals values.

When the `object` is of class `'lcmcross'`, `residuals` returns a data frame containing the residuals values for each latent class, where each variable terminates with `"_c#"`, `"#"` being the class number.

Note

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

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

See Also

`sfacross`, for the stochastic frontier analysis model fitting function.

`lcmcross`, for the latent class stochastic frontier analysis model fitting function.

Examples

```
## Using data on fossil fuel fired steam electric power generation plants in the U.S.
# Translog SFA (cost function) truncated normal with scaling property
tl_u_ts <- sfacross(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)),
    udist = 'tnormal', muhet = ~ regu, uhet = ~ regu, data = utility, S = -1,
    scaling = TRUE, method = 'mla')
  resid.tl_u_ts <- residuals(tl_u_ts)
  head(resid.tl_u_ts)

## Using data on eighty-two countries production (DGP)
# LCM Cobb Douglas (production function) half normal distribution
cb_2c_h <- lcmcross(formula = ly ~ lk + ll + yr, udist = 'hnormal', data = worldprod, S = 1)
  resid.cb_2c_h <- residuals(cb_2c_h)
  head(resid.cb_2c_h)
```

`ricephil` *Data on rice production in the Philippines*

Description

This dataset contains annual data collected from 43 smallholder rice producers in the Tarlac region of the Philippines between 1990 and 1997.

Usage

```
ricephil
```

Format

A data frame with 344 observations on the following 17 variables.

YEARDUM Time period (1= 1990, ..., 8 = 1997).

FARMERCODE Farmer code (1, ..., 43).

PROD Output (tonnes of freshly threshed rice).

AREA Area planted (hectares).

LABOR Labor used (man-days of family and hired labor).

NPK Fertiliser used (kg of active ingredients).

OTHER Other inputs used (Laspeyres index = 100 for Farm 17 in 1991).

PRICE Output price (pesos per kg).

AREAP Rental price of land (pesos per hectare).

LABORP Labor price (pesos per hired man-day).

NPKP Fertiliser price (pesos per kg of active ingredient).

OTHERP Price of other inputs (implicit price index).

AGE Age of the household head (years).

EDYRS Education of the household head (years).

HHSIZE Household size.

NADULT Number of adults in the household.

BANRAT Percentage of area classified as bantog (upland) fields.

Details

This dataset is published as supplement to Coelli *et al.* (2005). While most variables of this dataset were supplied by the International Rice Research Institute (IRRI), some were calculated by Coelli *et al.* (2005, see p. 325–326). The survey is described in Pandey *et al.* (1999).

Source

Supplementary files for Coelli *et al.* (2005), http://www.uq.edu.au/economics/cepa/crob2005/software/CROB2005.zip.

References

Coelli, T. J., Rao, D. S. P., O'Donnell, C. J., and Battese, G. E. 2005. *An Introduction to Efficiency and Productivity Analysis*, Springer, New York.

Pandey, S., Masciat, P., Velasco, L, and Villano, R. 1999. Risk analysis of a rainfed rice production system system in Tarlac, Central Luzon, Philippines. *Experimental Agriculture*, **35**:225–237.

Examples

```
str(ricephil)
summary(ricephil)
```

---

sfacross *Stochastic frontier estimation using cross-section data*

Description

sfacross is a symbolic formula-based function for the estimation of stochastic frontier models in the case of cross-sectional or pooled cross-section data, using maximum (simulated) likelihood - M(S)L.

The function accounts for heteroscedasticity in both one-sided and two-sided error terms as in Reifschneider and Stevenson (1991), Caudill and Ford (1993), Caudill *et al.* (1995) and Hadri (1999), but also heterogeneity in the mean of the pre-truncated distribution as in Kumbhakar *et al.* (1991), Huang and Liu (1994) and Battese and Coelli (1995).

Ten distributions are possible for the one-sided error term and nine optimization algorithms are available.

The truncated normal - normal distribution with scaling property as in Wang and Schmidt (2002) is also implemented.

Usage

```
sfacross(formula, muhet, uhet, vhet, logDepVar = TRUE, data, subset, S = 1,
  udist = "hnormal", scaling = FALSE, start = NULL, method = "bfgs", hessianType = 1,
  simType = "halton", Nsim = 100, prime = 2, burn = 10, antithetics = FALSE,
  seed = 12345, itermax = 2000, printInfo = FALSE, tol = 1e-12, gradtol = 1e-06,
  stepmax = 0.1, qac = "marquardt")
```

Arguments

formula	A symbolic description of the model to be estimated based on the generic function formula (see section 'Details').
muhet	A one-part formula to consider heterogeneity in the mean of the pre-truncated distribution (see section 'Details').
uhet	A one-part formula to consider heteroscedasticity in the one-sided error variance (see section 'Details').

vhet
: A one-part formula to consider heteroscedasticity in the two-sided error variance (see section 'Details').

logDepVar
: Logical. Informs whether the dependent variable is logged (TRUE) or not (FALSE). Default = TRUE.

data
: The data frame containing the data.

subset
: An optional vector specifying a subset of observations to be used in the optimization process.

S
: If S = 1 (default), a production (profit) frontier is estimated: $\epsilon_i = v_i - u_i$. If S = -1, a cost frontier is estimated: $\epsilon_i = v_i + u_i$.

udist
: Character string. Default = "hnormal". Distribution specification for the one-sided error term. 10 different distributions are available:

- "hnormal", for the half normal distribution (Aigner *et al.* 1977, Meeusen and Vandenbroeck 1977)
- "exponential", for the exponential distribution
- "tnormal" for the truncated normal distribution (Stevenson 1980)
- "rayleigh", for the Rayleigh distribution (Hajargasht 2015)
- "uniform", for the uniform distribution (Li 1996, Nguyen 2010)
- "gamma", for the Gamma distribution (Greene 2003)
- "lognormal", for the log normal distribution (Migon and Medici 2001, Wang and Ye 2020)
- "weibull", for the Weibull distribution (Tsionas 2007)
- "genexponential", for the generalized exponential distribution (Papadopoulos 2020)
- "tslaplace", for the truncated skewed Laplace distribution (Wang 2012).

scaling
: Logical. Only when udist = "tnormal" and scaling = TRUE, the scaling property model (Wang and Schmidt 2002) is estimated. Default = FALSE. (see section 'Details').

start
: Numeric vector. Optional starting values for the maximum likelihood (ML) estimation.

method
: Optimization algorithm used for the estimation. Default = "bfgs". 9 algorithms are available:

- "bfgs", for Broyden-Fletcher-Goldfarb-Shanno (see maxBFGS)
- "bhhh", for Berndt-Hall-Hall-Hausman (see maxBHHH)
- "nr", for Newton-Raphson (see maxNR)
- "nm", for Nelder-Mead (see maxNM)
- "ucminf", implements a quasi-Newton type with BFGS updating of the inverse Hessian and soft line search with a trust region type monitoring of the input to the line search algorithm (see ucminf)
- "mla", for general-purpose optimization based on Marquardt-Levenberg algorithm (see mla)
- "sr1", for Symmetric Rank 1 (see trust.optim)
- "sparse", for trust regions and sparse Hessian (see trust.optim)
- "nlminb", for optimization using PORT routines (see nlminb)

hessianType	Integer. If 1 (Default), analytic Hessian is returned for all the distributions except "gamma", "lognormal" and "weibull" for which the numeric Hessian is returned. If 2, bhhh Hessian is estimated ($g'g$). If 3, robust Hessian is computed ($H^{-1}GH^{-1}$).
simType	Character string. If simType = "halton" (Default), Halton draws are used for maximum simulated likelihood (MSL). If simType = "ghalton", Generalized-Halton draws are used for MSL. If simType = "sobol", Sobol draws are used for MSL. If simType = "uniform", uniform draws are used for MSL. (see section 'Details').
Nsim	Number of draws for MSL.
prime	Prime number considered for Halton and Generalized-Halton draws. Default = 2.
burn	Number of the first observations discarded in the case of Halton draws. Default = 10.
antithetics	Logical. Default = FALSE. If TRUE, antithetics counterpart of the uniform draws is computed. (see section 'Details').
seed	Numeric. Seed for the random draws.
itermax	Maximum number of iterations allowed for optimization. Default = 2000.
printInfo	Logical. Print information during optimization. Default = FALSE.
tol	Numeric. Convergence tolerance. Default = 1e-12.
gradtol	Numeric. Convergence tolerance for gradient. Default = 1e-06.
stepmax	Numeric. Step max for ucminf algorithm. Default = 0.1.
qac	Character. Quadratic Approximation Correction for "bhhh" and "nr" algorithms. If "stephalving", the step length is decreased but the direction is kept. If "marquardt" (default), the step length is decreased while also moving closer to the pure gradient direction. See maxBHHH and maxNR.

Details

The stochastic frontier model is defined as:

$$y_i = \alpha + \mathbf{x}_i'\beta + v_i - Su_i$$

$$\epsilon_i = v_i - Su_i$$

where i is the observation, j is the class, y is the output (cost, revenue, profit), x is the vector of main explanatory variables (inputs and other control variables), u is the one-sided error term with variance σ_u^2, and v is the two-sided error term with variance σ_v^2.

S = 1 in the case of production (profit) frontier function and S = -1 in the case of cost frontier function.

The model is estimated using maximum likelihood (ML) for most distributions except the Gamma, Weibull and log-normal distributions for which maximum simulated likelihood (MSL) is used. For this latter, several draws can be implemented namely Halton, Generalized Halton, Sobol and uniform. In the case of uniform draws, antithetics can also be computed: first Nsim/2 draws are obtained, then the Nsim/2 other draws are obtained as counterpart of one (1-draw).

To account for heteroscedasticity in the variance parameters of the error terms, a single part (right) formula can also be specified. To impose the positivity to these parameters, the variances are modelled as: $\sigma^2_u = \exp{(\delta' Z_u)}$ or $\sigma^2_v = \exp{(\phi' Z_v)}$, where Z_u and Z_v are the heteroscedasticity variables (inefficiency drivers in the case of Z_u) and δ and ϕ the coefficients. In the case of heterogeneity in the truncated mean μ, it is modelled as $\mu = \omega' Z_{\mu}$. The scaling property can be applied for the truncated normal distribution: $u \sim h(Z_u, \delta)u$ where u follows a truncated normal distribution $N^+(\tau, \exp{(cu)})$.

In the case of the truncated normal distribution, the convolution of u_i and v_i is:

$$f(\epsilon_i) = \frac{1}{\sqrt{\sigma_u^2 + \sigma_v^2}} \phi\left(\frac{S\epsilon_i + \mu}{\sqrt{\sigma_u^2 + \sigma_v^2}}\right) \Phi\left(\frac{\mu_{i*}}{\sigma_*}\right) / \Phi\left(\frac{\mu}{\sigma_u}\right)$$

where

$$\mu_{i*} = \frac{\mu\sigma_v^2 - S\epsilon_i\sigma_u^2}{\sigma_u^2 + \sigma_v^2}$$

and

$$\sigma_*^2 = \frac{\sigma_u^2\sigma_v^2}{\sigma_u^2 + \sigma_v^2}$$

In the case of the half normal distribution the convolution is obtained by setting $\mu = 0$.

Value

`sfacross` returns a list of class `'sfacross'` containing the following elements:

`call`	The matched call.
`formula`	The estimated model.
`S`	The argument `'S'`. See the section 'Arguments'.
`typeSfa`	Character string. "Stochastic Production/Profit Frontier, e = v - u" when `S = 1` and "Stochastic Cost Frontier, e = v + u" when `S = -1`.
`Nobs`	Number of observations used for optimization.
`nXvar`	Number of explanatory variables in the production or cost frontier.
`nmuZUvar`	Number of variables explaining heterogeneity in the truncated mean, only if `udist = "tnormal"` or `"lognormal"`.
`scaling`	The argument `'scaling'`. See the section 'Arguments'.
`logDepVar`	The argument `'logDepVar'`. See the section 'Arguments'.
`nuZUvar`	Number of variables explaining heteroscedasticity in the one-sided error term.
`nvZVvar`	Number of variables explaining heteroscedasticity in the two-sided error term.
`nParm`	Total number of parameters estimated.
`udist`	The argument `'udist'`. See the section 'Arguments'.
`startVal`	Numeric vector. Starting value for M(S)L estimation.

dataTable	A data frame (tibble format) containing information on data used for optimization along with residuals and fitted values of the OLS and M(S)L estimations, and the individual observation log-likelihood.
olsParam	Numeric vector. OLS estimates.
olsStder	Numeric vector. Standard errors of OLS estimates.
olsSigmasq	Numeric. Estimated variance of OLS random error.
olsLoglik	Numeric. Log-likelihood value of OLS estimation.
olsSkew	Numeric. Skewness of the residuals of the OLS estimation.
olsM3Okay	Logical. Indicating whether the residuals of the OLS estimation have the expected skewness.
CoelliM3Test	Coelli's test for OLS residuals skewness. (See Coelli, 1995).
AgostinoTest	D'Agostino's test for OLS residuals skewness. (See D'Agostino and Pearson, 1973).
optType	Optimization algorithm used.
nIter	Number of iterations of the ML estimation.
optStatus	Optimization algorithm termination message.
startLoglik	Log-likelihood at the starting values.
mlLoglik	Log-likelihood value of the M(S)L estimation.
mlParam	Parameters obtained from M(S)L estimation.
gradient	Each variable gradient of the M(S)L estimation.
gradL_OBS	Matrix. Each variable individual observation gradient of the M(S)L estimation.
gradientNorm	Gradient norm of the M(S)L estimation.
invHessian	Covariance matrix of the parameters obtained from the M(S)L estimation.
hessianType	The argument 'hessianType'. See the section 'Arguments'.
mlDate	Date and time of the estimated model.
simDist	The argument 'simDist', only if udist = "gamma", "lognormal" or , "weibull". See the section 'Arguments'.
Nsim	The argument 'Nsim', only if udist = "gamma", "lognormal" or , "weibull". See the section 'Arguments'.
FiMat	Matrix of random draws used for MSL, only if udist = "gamma", "lognormal" or , "weibull".

Note

For the Halton draws, the code is adapted from the **mlogit** package.

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

References

Aigner, D., Lovell, C. A. K., and Schmidt, P. 1977. Formulation and estimation of stochastic frontier production function models. *Journal of Econometrics*, **6**(1), 21–37.

Battese, G. E., and Coelli, T. J. 1995. A model for technical inefficiency effects in a stochastic frontier production function for panel data. *Empirical Economics*, **20**(2), 325–332.

Caudill, S. B., and Ford, J. M. 1993. Biases in frontier estimation due to heteroscedasticity. *Economics Letters*, **41**(1), 17–20.

Caudill, S. B., Ford, J. M., and Gropper, D. M. 1995. Frontier estimation and firm-specific inefficiency measures in the presence of heteroscedasticity. *Journal of Business & Economic Statistics*, **13**(1), 105–111.

Coelli, T. 1995. Estimators and hypothesis tests for a stochastic frontier function - a Monte-Carlo analysis. *Journal of Productivity Analysis*, **6**:247–268.

D'Agostino, R., and E.S. Pearson. 1973. Tests for departure from normality. Empirical results for the distributions of b_2 and $\sqrt{b_1}$. *Biometrika*, **60**:613–622.

Greene, W. H. 2003. Simulated likelihood estimation of the normal-Gamma stochastic frontier function. *Journal of Productivity Analysis*, **19**(2-3), 179–190.

Hadri, K. 1999. Estimation of a doubly heteroscedastic stochastic frontier cost function. *Journal of Business & Economic Statistics*, **17**(3), 359–363.

Hajargasht, G. 2015. Stochastic frontiers with a Rayleigh distribution. *Journal of Productivity Analysis*, **44**(2), 199–208.

Huang, C. J., and Liu, J.-T. 1994. Estimation of a non-neutral stochastic frontier production function. *Journal of Productivity Analysis*, **5**(2), 171–180.

Kumbhakar, S. C., Ghosh, S., and McGuckin, J. T. 1991) A generalized production frontier approach for estimating determinants of inefficiency in U.S. dairy farms. *Journal of Business & Economic Statistics*, **9**(3), 279–286.

Li, Q. 1996. Estimating a stochastic production frontier when the adjusted error is symmetric. *Economics Letters*, **52**(3), 221–228.

Meeusen, W., and Vandenbroeck, J. 1977. Efficiency estimation from Cobb-Douglas production functions with composed error. *International Economic Review*, **18**(2), 435–445.

Migon, H. S., and Medici, E. V. 2001. Bayesian hierarchical models for stochastic production frontier. Lacea, Montevideo, Uruguay.

Nguyen, N. B. 2010. Estimation of technical efficiency in stochastic frontier analysis. PhD dissertation, Bowling Green State University, August.

Papadopoulos, A. 2021. Stochastic frontier models using the generalized exponential distribution. *Journal of Productivity Analysis*, **55**:15–29.

Reifschneider, D., and Stevenson, R. 1991. Systematic departures from the frontier: A framework for the analysis of firm inefficiency. *International Economic Review*, **32**(3), 715–723.

Stevenson, R. E. 1980. Likelihood Functions for Generalized Stochastic Frontier Estimation. *Journal of Econometrics*, **13**(1), 57–66.

Tsionas, E. G. 2007. Efficiency measurement with the Weibull stochastic frontier. *Oxford Bulletin of Economics and Statistics*, **69**(5), 693–706.

Wang, K., and Ye, X. 2020. Development of alternative stochastic frontier models for estimating time-space prism vertices. *Transportation*.

Wang, H.J., and Schmidt, P. 2002. One-step and two-step estimation of the effects of exogenous variables on technical efficiency levels. *Journal of Productivity Analysis*, **18**:129–144.

Wang, J. 2012. A normal truncated skewed-Laplace model in stochastic frontier analysis. Master thesis, Western Kentucky University, May.

See Also

`summary` for creating and printing summary results.

`coef` for extracting coefficients of the estimation.

`efficiencies` for computing (in-)efficiency estimates.

`fitted` for extracting the fitted frontier values.

`ic` for extracting information criteria.

`logLik` for extracting log-likelihood value(s) of the estimation.

`marginal` for computing marginal effects of inefficiency drivers.

`residuals` for extracting residuals of the estimation.

`vcov` for computing the variance-covariance matrix of the coefficients.

`skewnessTest` for implementing skewness test.

Examples

```
## Using data on fossil fuel fired steam electric power generation plants in U.S.
# Translog (cost function) half normal with heteroscedasticity
tl_u_h <- sfacross(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)),
    udist = 'hnormal', uhet = ~ regu, data = utility, S = -1, method = 'bfgs')
  summary(tl_u_h)

# Translog (cost function) truncated normal with heteroscedasticity
tl_u_t <- sfacross(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)),
    udist = 'tnormal', muhet = ~ regu, data = utility, S = -1, method = 'bhhh')
  summary(tl_u_t)

# Translog (cost function) truncated normal with scaling property
tl_u_ts <- sfacross(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)),
    udist = 'tnormal', muhet = ~ regu, uhet = ~ regu, data = utility, S = -1,
    scaling = TRUE, method = 'mla')
  summary(tl_u_ts)

## Using data on Philippine rice producers
# Cobb Douglas (production function) generalized exponential, and Weibull distributions
    cb_p_ge <- sfacross(formula = log(PROD) ~ log(AREA) + log(LABOR) + log(NPK) +
```

```
    log(OTHER), udist = 'genexponential', data = ricephil, S = 1, method = 'bfgs')
  summary(cb_p_ge)

## Using data on U.S. electric utility industry
# Cost frontier Gamma distribution
tl_u_g <- sfacross(formula = log(cost/fprice) ~ log(output) + I(log(output)^2) +
    I(log(lprice/fprice)) + I(log(cprice/fprice)), udist = "gamma", uhet = ~ 1,
    data = electricity, S = -1, method = "bfgs", simType = "halton", Nsim = 200,
    hessianType = 2)
  summary(tl_u_g)
```

---

skewnessTest *Skewness test for stochastic frontier models*

Description

skewnessTest computes skewness test for stochastic frontier models (i.e. objects of class 'sfacross').

Usage

```
skewnessTest(object, test = "agostino")
```

Arguments

object An object of class 'sfacross', returned by sfacross.

test A character string specifying the test to implement. If "agostino" (default), D'Agostino skewness test is implemented (D'Agostino and Pearson, 1973). If "coelli", Coelli skewness test is implemented (Coelli, 1995).

Value

skewnessTest returns the results of either the D'Agostino's or the Coelli's skewness test.

Note

skewnessTest is currently only available for object of class 'sfacross'.

References

Coelli, T. 1995. Estimators and hypothesis tests for a stochastic frontier function - a Monte-Carlo analysis. *Journal of Productivity Analysis*, **6**:247–268.

D'Agostino, R., and E.S. Pearson. 1973. Tests for departure from normality. Empirical results for the distributions of b_2 and $\sqrt{b_1}$. *Biometrika*, **60**:613–622.

Examples

```
## Using data on fossil fuel fired steam electric power generation plants in the U.S.
# Translog SFA (cost function) truncated normal with scaling property
tl_u_ts <- sfacross(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)),
    udist = "tnormal", muhet = ~ regu, uhet = ~ regu, data = utility, S = -1,
    scaling = TRUE, method = "mla")
  skewnessTest(tl_u_ts)
  skewnessTest(tl_u_ts, test = "coelli")
```

---

summary *Summary of results for classic or latent class stochastic models*

Description

Create and print summary results for classic or latent class stochastic models returned by `sfacross` and `lcmcross`.

Usage

```
## S3 method for class 'sfacross'
summary(object, grad = FALSE, ci = FALSE, ...)

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

## S3 method for class 'lcmcross'
summary(object, grad = FALSE, ci = FALSE, ...)

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

Arguments

`object`	An object of either class `'sfacross'`, returned by the function `sfacross`, or class `'lcmcross'`, returned by the function `lcmcross`.
`grad`	Logical. Default = `FALSE`. If `TRUE`, the gradient for the maximum likelihood (ML) estimates of the different parameters is returned.
`ci`	Logical. Default = `FALSE`. If `TRUE`, the 95% confidence interval for the different parameters (OLS and ML estimates) is returned.
`...`	Currently ignored.
`x`	An object of either class `'summary.sfacross'` or `'summary.lcmcross'`.
`digits`	Numeric. Number of digits displayed in values.

Value

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

AIC	Akaike information criterion.
BIC	Bayesian information criterion.
HQIC	Hannan-Quinn information criterion.
sigmavSq	For object of class 'sfacross'. Variance of the two-sided error term (σ_v^2).
sigmauSq	For object of class 'sfacross'. Parametrization of the variance of the one-sided error term (σ_u^2).
Varu	For object of class 'sfacross'. Variance of the one-sided error term.
THETA	For object of class 'sfacross' with "udist = uniform". Θ value in the case the uniform distribution is defined as: $u_i \in [0, \Theta]$.
Eu	For object of class 'sfacross'. Expected unconditional inefficiency.
Expu	For object of class 'sfacross'. Expected unconditional efficiency.
olsRes	For object of class 'sfacross'. Matrix of OLS estimates, their standard errors, t-values, P-values, and when ci = TRUE their confidence intervals.
mlRes	Matrix of ML estimates, their standard errors, z-values, asymptotic P-values, and when grad = TRUE their gradient, ci = TRUE their confidence intervals.
chisq	For object of class 'sfacross'. Chi-square statistics of the difference between the stochastic frontier and the OLS.
df	Degree of freedom for the inefficiency model.

See Also

sfacross, for the stochastic frontier analysis model fitting function.

lcmcross, for the latent class stochastic frontier analysis model fitting function.

coef for extracting coefficients of the estimation.

efficiencies for computing (in-)efficiency estimates.

fitted for extracting the fitted frontier values.

ic for extracting information criteria.

logLik for extracting log-likelihood value(s) of the estimation.

marginal for computing marginal effects of inefficiency drivers.

residuals for extracting residuals of the estimation.

vcov for computing the variance-covariance matrix of the coefficients.

skewnessTest for implementing skewness test.

Examples

```
## Using data on fossil fuel fired steam electric power generation plants in the U.S.
# Translog SFA (cost function) truncated normal with scaling property
tl_u_ts <- sfacross(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)),
    udist = "tnormal", muhet = ~ regu, uhet = ~ regu, data = utility, S = -1,
    scaling = TRUE, method = "mla")
  summary(tl_u_ts, grad = TRUE, ci = TRUE)
```

---

swissrailways *Data on Swiss railways*

Description

This dataset is an unbalanced panel of 50 Swiss railway companies over the period 1985-1997.

Usage

```
swissrailways
```

Format

A data frame with 605 observations on the following 42 variables.

ID Firm identification.

YEAR Year identification.

NI Number of years observed.

STOPS Number of stops in network.

NETWORK Network length (in meters).

NARROW_T Dummy variable for railroads with narrow track.

RACK Dummy variable for 'rack rail' in network.

TUNNEL Dummy variable for network with tunnels over 300 meters on average.

T Time indicator, first year = 0.

Q2 Passenger output – passenger km.

Q3 Freight output – ton km.

CT Total cost (1,000 Swiss franc).

PL Labor price.

PE Electricity price.

PK Capital price.

VIRAGE 1 for railroads with curvy tracks.

LNCT Log of `CT/PE`.

LNQ2 Log of `Q2`.

LNQ3 Log of `Q3`.

LNNET Log of `NETWORK`/1000.

LNPL Log of `PL`/`PE`.

LNPE Log of `PE`.

LNPK Log of `PK`/`PE`.

LNSTOP Log of `STOPS`.

MLNQ2 Mean of `LNQ2`.

MLNQ3 Mean of `LNQ3`.

MLNNET Mean of `LNNET`.

MLNPL Mean of `LNPL`.

MLNPK Mean of `LNPK`.

MLNSTOP Mean of `LNSTOP`.

Details

The dataset is extracted from the annual reports of the Swiss Federal Office of Statistics on public transport companies and has been used in Farsi *et al.* (2005).

Source

http://pages.stern.nyu.edu/~wgreene/Text/Edition7/tablelist8new.htm

http://people.stern.nyu.edu/wgreene/Microeconometrics.htm

References

Farsi, M., M. Filippini, and W. Greene. 2005. Efficiency measurement in network industries: Application to the Swiss railway companies. *Journal of Regulatory Economics*, **28**:69–90.

Examples

```
str(swissrailways)
```

`utility` *Data on U.S. electricity generating plants*

Description

This dataset contains data on fossil fuel fired steam electric power generation plants in the United States between 1986 and 1996.

Usage

```
utility
```

Format

A data frame with 791 observations on the following 11 variables.

firm Plant identification.

year Year identification.

y Net-steam electric power generation in megawatt-hours.

regu Dummy variable which takes a value equal to 1 if the power plant is in a state which enacted legislation or issued a regulatory order to implement retail access during the sample period, and 0 otherwise.

k Capital stock.

labor Labor and maintenance.

fuel Fuel.

wl Labor price.

wf Fuel price.

wk Capital price.

tc Total cost.

Details

The dataset has been used in Kumbhakar *et al.* (2014).

Source

https://sites.google.com/site/sfbook2014/home/for-stata-v12-v13-v14

References

Kumbhakar, S.C., H.J. Wang, and A. Horncastle. 2014. *A Practitioner's Guide to Stochastic Frontier Analysis Using Stata*. Cambridge University Press.

Examples

```
str(utility)
summary(utility)
```

---

vcov *Compute variance-covariance matrix of classic or latent class stochastic models*

Description

vcov computes the variance-covariance matrix of the maximum likelihood (ML) coefficients of classic or latent class stochastic frontier models estimated by sfacross or lcmcross.

Usage

```
## S3 method for class 'sfacross'
vcov(object, extraPar = FALSE, ...)

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

Arguments

`object`	A classic or latent class stochastic frontier model returned by `sfacross` or `lcmcross`

`extraPar`	Logical. Only available for non heteroscedastic models returned by `sfacross`. Default = `FALSE`. If `TRUE`, variances and covariances of additional parameters are returned:
	`sigmaSq` = `sigmauSq` + `sigmavSq`
	`lambdaSq` = `sigmauSq`/`sigmavSq`
	`sigmauSq` = $\exp(Wu) = \exp(\delta Z_u)$
	`sigmavSq` = $\exp(Wv) = \exp(\phi Z_v)$
	`sigma` = `sigmaSq`^0.5
	`lambda` = `lambdaSq`^0.5
	`sigmau` = `sigmauSq`^0.5
	`sigmav` = `sigmavSq`^0.5
	`gamma` = `sigmauSq`/(`sigmauSq` + `sigmavSq`)

`...`	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 or the robust Hessian matrix is evaluated.

The argument `extraPar`, is currently available for objects of class `'sfacross'`. When `"extraPar = TRUE"`, the variance-covariance of the additional parameters is obtained using the delta method.

Value

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

Author(s)

K Hervé Dakpo, Yann Desjeux and Laure Latruffe

See Also

`sfacross`, for the stochastic frontier analysis model fitting function.

`lcmcross`, for the latent class stochastic frontier analysis model fitting function.

Examples

```
## Using data on Spanish dairy farms
# Cobb Douglas (production function) half normal distribution
cb_s_h <- sfacross(formula = YIT ~ X1 + X2 + X3 + X4, udist = "hnormal",
    data = dairyspain, S = 1, method = "bfgs")
  vcov(cb_s_h)
  vcov(cb_s_h, extraPar = TRUE)


## Using data on eighty-two countries production (DGP)
# LCM Cobb Douglas (production function) half normal distribution
cb_2c_h <- lcmcross(formula = ly ~ lk + ll + yr, udist = "hnormal",
    data = worldprod, uhet = ~ initStat, S = 1)
  vcov(cb_2c_h)
```

---

worldprod — *Data on world production*

Description

This dataset provides information on production related variables for eighty-two countries over the period 1960–1987.

Usage

```
worldprod
```

Format

A data frame with 2,296 observations on the following 12 variables.

country Country name.

code Country identification.

yr Year identification.

y GDP in 1987 U.S. dollars.

k Physical capital stock in 1987 U.S. dollars.

l Labor (number of individuals in the workforce between the ages of 15 and 64).

h Human capital-adjusted labor.

ly Log of `y`.

lk Log of `k`.

ll Log of `l`.

lh Log of `h`.

initStat Log of the initial capital to labor ratio of each country, `lk - ll`, measured at the beginning of the sample period.

Details

The dataset is from the World Bank STARS database and has been used in Kumbhakar *et al.* (2014).

Source

https://sites.google.com/site/sfbook2014/home/for-stata-v12-v13-v14

References

Kumbhakar, S.C., H.J. Wang, and A. Horncastle. 2014. *A Practitioner's Guide to Stochastic Frontier Analysis Using Stata*. Cambridge University Press.

Examples

```
str(worldprod)
summary(worldprod)
```

Index