Package 'frontier'

May 8, 2026

Version 1.1-8

Date 2020-04-15

Title Stochastic Frontier Analysis

Author Tim Coelli, Arne Henningsen

Maintainer Arne Henningsen <arne.henningsen@gmail.com>

Depends R (>= 2.15.0), micEcon (>= 0.6-14), lmtest (>= 0.9-24)

Suggests MCMCpack (>= 1.0-8), fdrtool (>= 1.2.6)

Imports moments (>= 0.11), stats (>= 2.15.0), Formula (>= 0.2-0), miscTools (>= 0.6-1), plm (>= 1.0-1)

Description Maximum Likelihood Estimation of Stochastic Frontier Production and Cost Functions. Two specifications are available: the error components specification with time-varying efficiencies (Battese and Coelli, 1992, <doi:10.1007/BF00158774>) and a model specification in which the firm effects are directly influenced by a number of variables (Battese and Coelli, 1995, <doi:10.1007/BF01205442>).

License GPL (>= 2)

URL http://frontier.r-forge.r-project.org/

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-04-17 16:10:03 UTC

Contents

`coef.front41Output` *Coefficients from Frontier 4.1*

Description

These methods return the coefficients and their covariance matrix from a model estimated by Frontier 4.1.

Usage

```
   ## S3 method for class 'front41Output'
coef( object, which = "MLE", ... )

   ## S3 method for class 'summary.front41Output'
coef( object, which = "MLE", ... )

   ## S3 method for class 'front41Output'
vcov( object, ... )
```

Arguments

`object`	an object of class `front41Output` or `summary.front41Output` (read/created by `front41ReadOutput` or `summary.front41Output`, respectively).
`which`	character string indication, which coefficients should be returned: either 'OLS' (from OLS estimation), 'GRID' (from grid search), or 'MLE' (from maximum likelihood estimation).
`...`	currently ignored.

Value

The `coef` method applied to an object of class `front410utput` returns a vector containing all coefficients estimated by Frontier 4.1.

The `coef` method applied to an object of class `summary.front410utput` returns a matrix containing the estimates, their standard errors, the t values and P values of all coefficients estimated by Frontier 4.1.

The `vcov` method returns the covariance matrix of all coefficients estimated by Frontier 4.1.

Author(s)

Arne Henningsen

See Also

`front41ReadOutput`

`coef.frontier` *coef method for class frontier*

Description

Extract the coefficients from stochastic frontier models returned by `frontier`.

Usage

```
## S3 method for class 'frontier'
coef( object, which = "mle", extraPar = FALSE, ... )
```

Arguments

`object`	an object of class `frontier` (returned by the function `frontier`).
`which`	character string. Which coefficients should be returned? ('start' for starting values provided by the user, 'ols' for coefficients estimated by OLS, 'grid' for coefficients obtained by the grid search, or 'mle' for coefficients estimated by Maximum Likelihood).
`extraPar`	logical. If `TRUE`, additional parameters are returned: `sigmaSqU` = `sigmaSq` * `gamma` (with $u \sim N^+($ `mu`, `sigmaSqU` $)$), `sigmaSqV` = `sigmaSq` * (1 - `gamma`) (with $v \sim$ N(0, `sigmaSqV`)), `sigma` = `sigmaSq`^0.5, `sigmaU` = `sigmaSqU`^0.5, `sigmaV` = `sigmaSqV`^0.5, `lambdaSq` = `sigmaSqU` / `sigmaSqV`, and `lambda` = `sigmaU` / `sigmaV`. Please note that `sigmaSqU` and `sigmaU` are not the variance and standard error, respectively, of u. If the model is an error components frontier and argument `timeEffect` is `FALSE`, also the following additional parameters are returned: `varU` = the variance of u, `sdU` = `varU`^0.5, and `gammaVar` = `varU` / (`varU` + `sigmaSqV`). Please note that the variance of u usually differs between observations if the model is an error component frontier with 'time effect' or an efficiency effects frontier.
`...`	currently unused.

Value

`coef.frontier` returns a named vector of the coefficients.

Author(s)

Arne Henningsen

See Also

`coef.summary.frontier` and `sfa`.

Examples

```
# example included in FRONTIER 4.1
data( front41Data )

sfaResult <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data )
coef( sfaResult, which = "ols" )
coef( sfaResult, which = "grid" )
coef( sfaResult )
```

---

coef.summary.frontier *coef method for class summary.frontier*

Description

Extract the coefficients, their standard errors, z-values or t-values, and (asymptotic) P-values from stochastic frontier models returned by the `summary` method for objects of class `frontier`.

Usage

```
## S3 method for class 'summary.frontier'
coef( object, which = "mle", ... )
```

Arguments

`object`	an object of class `summary.frontier` (returned by the `summary` method for objects of class `frontier`
`which`	character string. Which coefficients should be returned? ('ols' for coefficients estimated by OLS or 'mle' for coefficients estimated by Maximum Likelihood).
`...`	currently unused.

Details

The standard errors of the estimated parameters are taken from the direction matrix that is used in the final iteration of the Davidon-Fletcher-Powell procedure that is used for maximising the (log) likelihood function.

If argument `which` of this method is `"mle"` (the default) and argument `extraPar` of `summary.frontier` is set to `TRUE`, some additional parameters, their standard errors, z-values, and (asymptotic) P-values are returned (see documentation of `summary.frontier`, `coef.frontier`, or `vcov.frontier`). The standard errors of the additional parameters are obtained by the delta method. Please note that the delta method might provide poor approximations of the 'true' standard errors, because parameter σ^2 is left-censored and parameter γ is both left-censored and right-censored so that these parameters cannot be normally distributed.

Please note further that the t statistic and the z statistic are not reliable for testing the statistical signicance of σ^2, γ, and the 'additional parameters', because these parameters are censored and cannot follow a normal distribution or a t distribution.

Value

The `coef` method for objects of class `summary.frontier` returns a matrix, where the four columns contain the estimated coefficients, their standard errors, z-values or t-values, and (asymptotic) P-values.

Author(s)

Arne Henningsen

See Also

`coef.frontier`, `summary.frontier`, `vcov.frontier`, and `sfa`.

Examples

```
# example included in FRONTIER 4.1
data( front41Data )

sfaResult <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data )
coef( summary( sfaResult ), which = "ols" )
coef( summary( sfaResult ) )
coef( summary( sfaResult, extraPar = TRUE ) )
```

---

cooks.distance.frontier

Pseudo-Cook's Distance of Stochastic Frontier Models

Description

This method returns the Pseudo-Cook's distances from stochastic frontier models estimated with the **frontier** package (e.g., function `sfa`).

Usage

```
## S3 method for class 'frontier'
cooks.distance( model, target = "predict",
   asInData = FALSE, progressBar = TRUE, ... )
```

Arguments

model	a stochastic frontier model estimated with the **frontier** package (e.g. function sfa).
target	character string. If "predict", the returned values indicate the influence of individual observations on the predicted values; if "efficiencies", the returned values indicate the influence of individual observations on the efficiency estimates.
asInData	logical. If FALSE, the returned vector only includes observations that were used in the estimation; if TRUE, the length of the returned vector is equal to the total number of observations in the data set, where the values in the returned vector that correspond to observations that were not used in the estimation due to NA or infinite values are set to NA.
progressBar	logical. Should a progress bar be displayed while the Cook's distances are obtained?
...	additional arguments that arecurrently ignored if argument target is "predict" and that are passed to the efficiencies() method if argument target is "efficiencies".

Value

A vector of the Pseudo-Cook's distances for each observation that was used in the estimation that is provided as argument model.

Author(s)

Arne Henningsen

See Also

sfa, cooks.distance.

Examples

```
# example included in FRONTIER 4.1 (cross-section data)
data( front41Data )

# Cobb-Douglas production frontier
cobbDouglas <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data )
summary( cobbDouglas )

# Pseudo-Cook's distances for predicted values
cooks.distance( cobbDouglas )
```

```
# Pseudo-Cook's distances for efficiency estimates
cooks.distance( cobbDouglas, "efficiencies" )
```

`efficiencies` *Returning Efficiency Estimates*

Description

This method returns efficiency estimates from frontier models.

Usage

```
efficiencies( object, ... )
## Default S3 method:
efficiencies( object, ... )
```

Arguments

`object`	a frontier model.
`...`	further arguments for methods.

Details

This is a generic function. The default method just returns the element `effic` from `object`.

Author(s)

Arne Henningsen

See Also

`efficiencies.frontier`.

`efficiencies.frontier` *Returning Efficiency Estimates*

Description

This method returns efficiency estimates from stochastic frontier models estimated with `frontier`.

Usage

```
## S3 method for class 'frontier'
efficiencies( object, asInData = FALSE,
   logDepVar = TRUE, minusU = farrell, farrell = TRUE,
   margEff = FALSE, newdata = NULL, ... )
```

Arguments

object	a stochastic frontier model returned by frontier.
asInData	logical. If TRUE, the efficiency estimates are returned in the same order as the corresponding observations in the data set used for the estimation (see section 'value' below).
logDepVar	logical. Is the dependent variable logged?
minusU	logical. If TRUE (the default), the efficiencies are calculated by E[exp(-u)], i.e. Farrel-type efficiencies are returned for input-oriented models, Shepard-type efficiencies are returned for output-oriented models, and the returned efficiency estimates have values between zero and one, where a one indicates a fully efficient firm and a zero indicates a fully inefficient firm. If FALSE, the efficiencies are calculated by E[exp(u)], i.e. Shepard-type efficiencies are returned for input-oriented models, Farrell-type efficiencies are returned for output-oriented models, and the returned efficiency estimates have values larger than or equal to one, where a one indicates a fully efficient firm and plus infinity indicates a fully inefficient firm.
farrell	logical. This argument is only kept for backward compatibility and will be removed in the future.
margEff	logical. If TRUE, the marginal effects of the z variables (of an Efficiency Effects Frontier, EEF) on the efficiency measure are returned as an 'attribute' to the returned object (i.e. the efficiency estimates). These marginal effects are calculated by the formula derived in Olsen and Henningsen (2011), which was slightly adjusted for the differing model specifications. Currently, this feature is implemented only for models with logged dependent variables.
newdata	an optional data frame from which the values of explanatory variables and the dependent variable are taken to calculate the efficiency estimates. If this argument is NULL (the default), the efficiency estimates are calculated for the observations that were used in the estimation.
...	currently ignored.

Value

If argument asInData is FALSE (default), a matrix of efficiency estimates is returned, where each row corresponds to a firm (cross-section unit) and each column corresponds to a time period (only if efficiency estimates differ between time periods).

If argument asInData is TRUE, a vector of efficiency estimates is returned, where the efficiency estimates are in the same order as the corresponding observations in the data set used for the estimation.

If argument margEff is TRUE, and the model is an Efficiency Effects Frontier (EFF) with z variables, and the dependent variable is logged, the returned efficiency estimates have an attribute "margEff" that contains the marginal effects of the z variables on the efficiency measure.

If the dependent variable is logged, the marginal effect of the kth z variable on the efficiency is

$$\frac{\partial E[\exp(-\kappa\, u)]}{\partial z_{kit}} = \frac{\delta_k(1-\gamma)\exp\left(-\kappa\,\bar{\mu}_{it} + \frac{1}{2}\bar{\sigma}^2\right)}{\Phi\left(\frac{\bar{\mu}_{it}}{\bar{\sigma}}\right)}$$

$$\cdot \left(\frac{\phi\left(-\kappa\,\bar{\sigma} + \frac{\bar{\mu}_{it}}{\bar{\sigma}}\right)}{\bar{\sigma}} - \frac{\Phi\left(-\kappa\,\bar{\sigma} + \frac{\bar{\mu}_{it}}{\bar{\sigma}}\right)\,\phi\left(\frac{\bar{\mu}_{it}}{\bar{\sigma}}\right)}{\bar{\sigma}\,\Phi\left(\frac{\bar{\mu}_{it}}{\bar{\sigma}}\right)} - \kappa\,\Phi\left(-\kappa\,\bar{\sigma} + \frac{\bar{\mu}_{it}}{\bar{\sigma}}\right) \right),$$

where

$$\bar{\mu}_{it} = (1-\gamma)\,z_{it}'\delta - \tau\,\gamma\,\epsilon_{it},$$
$$\bar{\sigma}^2 = \gamma\,(1-\gamma)\,\sigma^2,$$

$\kappa = 1$ in case of Farrell efficiencies (i.e. efficiencies have values between between 0 and 1), whereas $\kappa = -1$ otherwise (i.e. efficiencies have values larger than 1), and $\tau = 1$ if inefficiency decreases the dependent variable, whereas $\tau = -1$ otherwise (see Olsen and Henningsen 2011).

If argument `asInData` is `FALSE`, this attribute is a 3-dimensional array, where the first dimension represents the individual firm, the second deminsion represents the time period, and the third dimension represents the z variables. In contrast, if argument `asInData` is `TRUE`, this attribute is a matrix, where the rows represent the observations and the columns represent the z variables.

Author(s)

Arne Henningsen

References

Olsen, Jakob Vesterlund and Arne Henningsen (2011): Investment utilization and farm efficiency in Danish agriculture. FOI working paper 2011/13, Institute of Food and Resource Economics, University of Copenhagen, http://EconPapers.repec.org/RePEc:foi:wpaper:2011_13.

See Also

`sfa`, `summary.frontier`, and `efficiencies`.

Examples

```
# rice producers in the Philippines (panel data)
data( "riceProdPhil" )
library( "plm" )
riceProdPhil <- pdata.frame( riceProdPhil, c( "FMERCODE", "YEARDUM" ) )

# Error Components Frontier (Battese & Coelli 1992), no time effect
rice <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil )
efficiencies( rice )
riceProdPhil$efficiencies <- efficiencies( rice, asInData = TRUE )

# efficiency of an 'average' farm
efficiencies( rice,
   newdata = data.frame( t( colMeans( riceProdPhil[ , -c(1,2) ] ) ) ) )

# Error Components Frontier (Battese & Coelli 1992), with time effect
riceTime <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil, timeEffect = TRUE )
efficiencies( riceTime )
riceProdPhil$efficienciesTime <- efficiencies( riceTime, asInData = TRUE )
```

```
# Technical Efficiency Effects Frontier (Battese & Coelli 1995)
rice2 <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ) |
   EDYRS + BANRAT - 1, data = riceProdPhil )
eff <- efficiencies( rice2, margEff = TRUE )
attr( eff, "margEff" )   # marginal effects
```

`elas.frontierQuad` *Elasticities of a Quadratic/Translog Frontier*

Description

Calculate the elasticities of a quadratic or translog frontier function.

Usage

```
## S3 method for class 'frontierQuad'
elas( object, data = NULL, dataLogged = TRUE,
   yObs = FALSE, ... )
```

Arguments

`object`	object of class `frontierQuad` (returned by `frontierQuad`).
`data`	dataframe containing the data; if it is not specified, the data frame that was used for the frontier estimation is used for calculating elasticities.
`dataLogged`	logical. Are the variables (specified in arguments `yName` and `xNames` and available in argument `data`) already logged? (If argument `dataLogged` is `TRUE`, the frontier function is of the translog form; if argument `dataLogged` is `FALSE`, the frontier function is quadratic).
`yObs`	logical. Use observed values of the endogenous variable. If `FALSE` (default) predicted values calculated by `quadFuncCalc` are used (ignored if argument `dataLogged` is `TRUE`).
`...`	currently ignored.

Details

This method internally calls the functions `translogEla` and `quadFuncEla`.

Value

See documentation of `translogEla` and `quadFuncEla`.

Author(s)

Arne Henningsen

See Also

frontierQuad, translogEla, and quadFuncEla.

Examples

```
# example included in FRONTIER 4.1 (cross-section data)
data( front41Data )
front41Data$logOutput  <- log( front41Data$output )
front41Data$logCapital <- log( front41Data$capital )
front41Data$logLabour  <- log( front41Data$labour )

translog <- frontierQuad( yName = "logOutput",
   xNames = c( "logCapital", "logLabour" ),
   data = front41Data )
elas( translog )
```

---

fitted.frontier *Fitted and Predicted (Frontier) Values*

Description

This method returns the fitted and predicted "frontier" values from stochastic frontier models estimated with the **frontier** package (e.g. function sfa).

Usage

```
## S3 method for class 'frontier'
fitted( object, asInData = FALSE, ... )

## S3 method for class 'frontier'
predict( object, newdata = NULL, asInData = TRUE, ... )
```

Arguments

object	a stochastic frontier model estimated with the **frontier** package (e.g. function sfa).
newdata	an optional data frame from which the explanatory variables are used to calculate the predicted "frontier" values. If this argument is NULL, the fitted values are returned.
asInData	logical. If TRUE, the fitted values are returned in the same order as the corresponding observations in the data set used for the estimation (see section 'value' below).
...	currently ignored.

Value

If argument asInData is FALSE, a matrix of the fitted or predicted values is returned, where each row corresponds to a firm (cross-section unit) and each column corresponds to a time period.

If argument asInData is TRUE, a vector of fitted or predicted values is returned, where the fitted values are in the same order as the corresponding observations in the data set used for the estimation or the data set specified by argument newdata.

Author(s)

Arne Henningsen

See Also

sfa, fitted, predict.

Examples

```
# rice producers in the Philippines (panel data)
data( "riceProdPhil" )
library( "plm" )
riceProdPhil <- pdata.frame( riceProdPhil, c( "FMERCODE", "YEARDUM" ) )

# Error Components Frontier (Battese & Coelli 1992), no time effect
rice <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil )
fitted( rice )
riceProdPhil$fitted <- fitted( rice, asInData = TRUE )

# Error Components Frontier (Battese & Coelli 1992), with time effect
riceTime <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil, timeEffect = TRUE )
fitted( riceTime )
riceProdPhil$fittedTime <- fitted( riceTime, asInData = TRUE )
```

---

front41Data *Data provided with Tim Coelli's Frontier 4.1*

Description

The front41Data data frame contains cross-sectional data of 60 firms.

Usage

```
data(front41Data)
```

Format

This data frame contains the following columns:

firm firm ID.

output output quantity (value added).

capital capital input quantity (quantity index).

labour labour input quantity (quantity index).

Source

Coelli, T. (1996) A Guide to FRONTIER Version 4.1: A Computer Program for Stochastic Frontier Production and Cost Function Estimation, CEPA Working Paper 96/08, http://www.uq.edu.au/economics/cepa/frontier.php, University of New England.

---

front41Est *Estimate a Stochastic Frontier Model by Frontier 4.1*

Description

Estimate a stochastic frontier model with a modified version of Tim Coelli's program Frontier 4.1 (NOTE: this program has to be installed separately!).

Usage

```
front41Est( command = ifelse( .Platform$OS.type == "windows",
   "front41.exe", "front41.bin" ), ... )
```

Arguments

command	command to call the modified version of FRONTIER 4.1 (see details).
...	arguments passed to front41WriteInput.

Details

Using the command front41Est requires the installation of a modified version of Tim Coelli's FRONTIER 4.1. It is available on http://frontier.r-forge.r-project.org/front41.html. as (FORTRAN) source code and (executable) binaries for GNU/Linux and MS-Windows.

Value

front41Est returns a list of class front41Output that is returned by front41ReadOutput with two additional elements:

input	object returned by front41WriteInput.
messages	messages returned by FRONTIER 4.1.

Author(s)

Arne Henningsen

References

Battese, G.E. and T. Coelli (1992), Frontier production functions, technical efficiency and panel data: with application to paddy farmers in India. *Journal of Productivity Analysis*, 3, 153-169.

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

Coelli, T. (1996) A Guide to FRONTIER Version 4.1: A Computer Program for Stochastic Frontier Production and Cost Function Estimation, CEPA Working Paper 96/08, http://www.uq.edu.au/economics/cepa/frontier.php, University of New England.

See Also

front41WriteInput, front41ReadOutput

Examples

```
data( front41Data )
front41Data$logOutput  <- log( front41Data$output )
front41Data$logCapital <- log( front41Data$capital )
front41Data$logLabour  <- log( front41Data$labour )

## Not run:
front41Est( data = front41Data, crossSectionName = "firm",
   yName = "logOutput", xNames = c( "logCapital", "logLabour" ) )

## End(Not run)
```

---

front41ReadOutput *Read output of Frontier 4.1*

Description

Read the output file of Tim Coelli's program Frontier 4.1 that performs stochastic frontier analysis.

Usage

```
front41ReadOutput( file = "front41.out" )

## S3 method for class 'front41Output'
print( x, efficiencies = FALSE, ... )
```

Arguments

file	character variable with the name of the file to read.
x	object of class front41Output (returned by front41ReadOutput.
efficiencies	logical. Print all efficiency estimates? (If FALSE, only the mean efficiency is printed.)
...	currently ignored.

Details

A modified version of Tim Coelli's FRONTIER 4.1 that can be used non-interactively is available on http://frontier.r-forge.r-project.org/front41.html. It can be called from within R using the system command (see example). This version is is available as (FORTRAN) source code and (executable) binaries for GNU/Linux and MS-Windows.

Value

a list of class front41Output containing following objects:

version	the version of Frontier 4.1 that produced the output.
insFile	name of the instruction file used by Frontier 4.1.
dtaFile	name of the data file used by Frontier 4.1.
modelType	model type: either 1 for 'Error Components Frontier' or 2 for 'Tech. Eff. Effects Frontier'.
modelTypeName	model type: 'Error Components Frontier' or 'Tech. Eff. Effects Frontier'.
functionType	function type: either 1 for 'production function' or 2 for 'cost function'.
functionTypeName	function type: 'production function' or 'cost function'.
logDepVar	logical. Is the dependent variable logged.
olsResults	results of the OLS estimation.
nXvars	number X variables (exogenous variables of the production or cost function.
olsLogl	log likelihood value of the OLS estimation.
gridResults	results of the grid search.
mleResults	results of the maximum likelihood estimation.
mleLogl	log likelihood value of the maximum likelihood estimation.
mleCov	coefficient covariance matrix of the maximum likelihood estimation.
lrTest	LR test of the one-sided error.
lrTestRestrict	number of restrictions of the LR test.
nIter	number of iterations.
maxIter	maximum number of iterations set.
nCross	number of cross-sections.
nPeriods	umber of time periods.

nObs	total number of observations.
nObsMissing	number of observations that are not in the panel.
efficiency	technical efficiency estimates.
meanEfficiency	mean efficiency.

Author(s)

Arne Henningsen

References

Battese, G.E. and T. Coelli (1992), Frontier production functions, technical efficiency and panel data: with application to paddy farmers in India. *Journal of Productivity Analysis*, 3, 153-169.

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

Coelli, T. (1996) A Guide to FRONTIER Version 4.1: A Computer Program for Stochastic Frontier Production and Cost Function Estimation, CEPA Working Paper 96/08, http://www.uq.edu.au/economics/cepa/frontier.php, University of New England.

See Also

front41WriteInput, front41Est

Examples

```
# read the output file that is provided with Frontier 4.1
outFile <- system.file( "front41/EG1.OUT", package = "frontier" )
sfa <- front41ReadOutput( outFile )
print( sfa, efficiencies = TRUE )

# perform an SFA and read the output
data( front41Data )
front41Data$logOutput  <- log( front41Data$output )
front41Data$logCapital <- log( front41Data$capital )
front41Data$logLabour  <- log( front41Data$labour )

front41WriteInput( front41Data, "firm", yName = "logOutput",
   xNames = c( "logCapital", "logLabour" ),
   path = tempdir(), insFile = "coelli.ins" )

## Not run:
system( paste0( "cd ", tempdir(), "; front41.bin coelli.ins" ) )
sfa <- front41ReadOutput( file.path( tempdir(), "coelli.out" ) )
summary( sfa )

## End(Not run)
```

front41WriteInput *Write input files for Frontier 4.1*

Description

Write an instruction file, a data file, and a start-up file for Tim Coelli's program Frontier 4.1 that performs stochastic frontier analysis.

Usage

```
front41WriteInput( data, crossSectionName, timePeriodName = NULL,
   yName, xNames = NULL, qxNames = NULL, zNames = NULL, quadHalf = TRUE,
   modelType = ifelse( is.null( zNames ), 1, 2 ), functionType = 1,
   logDepVar = TRUE, mu = FALSE, eta = FALSE, path = ".",
   insFile = "front41.ins", dtaFile = sub( "\\.ins$", ".dta", insFile ),
   outFile = sub( "\\.ins$", ".out", insFile ), startUpFile = "front41.000",
   iprint = 5, indic = 1, tol = 0.00001, tol2 = 0.001, bignum = 1.0E+16,
   step1 = 0.00001, igrid2 = 1, gridno = 0.1, maxit = 100, ite = 1 )
```

Arguments

data	data frame that contains the data.
crossSectionName	string: name of the cross section identifier.
timePeriodName	string: name of the time period identifier or NULL in case of cross-section data.
yName	string: name of the endogenous variable.
xNames	a vector of strings containing the names of the X variables (exogenous variables of the production or cost function).
qxNames	a vector of strings containing the names of the variables to construct quadratic and interaction terms. As a shortcut, this argument can be set to "all" for using all variables specified in argument xNames to get a full quadratic or translog model.
zNames	a vector of strings containing the names of the Z variables (variables explaining the efficiency level).
quadHalf	logical. Multiply the quadratic terms by one half?
modelType	model type: either 1 for an 'Error Components Frontier' or 2 for an 'Efficiency Effects Frontier'.
functionType	function type: either 1 for 'production function' or 2 for 'cost function'.
logDepVar	logical. Is the dependent variable logged.
mu	logical. Should a 'mu' (in case of an 'Error Components Frontier', i.e. modelType = 1) or a delta0 (in case of an 'Efficiency Effects Frontier', i.e. modelType = 2) be included in the estimation.
eta	logical. Should an 'eta' be included in the estimation (only in case of an 'Error Components Frontier', i.e. modelType = 1).

path	path in which the instruction file, the data file, and the start-up file should be written.
insFile	name of the instruction file.
dtaFile	name of the data file.
outFile	name of the output file.
startUpFile	name of the start-up file. If this argument is NULL, no start-up file is written.
iprint	numeric. Print info every iprint iterations; if this argument is 0, do not print.
indic	numeric. Use in unidimensional search procedure: indic = 2 says do not scale step length in unidimensional search; indic = 1 says scale (to length of last step) only if last step was smaller; indic = any other number says scale (to length of last step).
tol	numeric. Convergence tolerance (proportiona).
tol2	numeric. Tolerance used in uni-dimensional search procedure.
bignum	numeric. Used to set bounds on densities and distributions.
step1	numeric. Size of 1st step in search procedure.
igrid2	numeric. 1 = double accuracy, 0 = single accuracy.
gridno	numeric. Steps taken in single accuracy grid search on gamma.
maxit	numeric. Maximum number of iterations permitted
ite	numeric. 1 = print all efficiency estimates; 0 = print only the mean efficiency.

Details

A modified version of Tim Coelli's FRONTIER 4.1 that can be used non-interactively is available on http://frontier.r-forge.r-project.org/front41.html. It can be called from within R using the system command (see example). This version is is available as (FORTRAN) source code and (executable) binaries for GNU/Linux and MS-Windows.

Value

front41WriteInput writes an instruction file, a data file, and a start-up file for Frontier 4.1 to disk and it invisibly returns a list of class front41WriteInput. This list contains mainly the arguments with which front41WriteInput was called. An exception is element data, which is *not* the argument data but the data matrix that was written into the data file. Furthermore, in case of an Efficiency Effects Model, the element eta contains the number of Z variables. Additionally, the returned list contains following elements:

nCrossSection	number of cross section units.
nTimePeriods	number of time periods.
nTotalObs	total number of observations.
nXtotal	total number of X variables (including quadratic and interaction terms).
nZvars	number of Z variables.

Author(s)

Arne Henningsen

References

Battese, G.E. and T. Coelli (1992), Frontier production functions, technical efficiency and panel data: with application to paddy farmers in India. *Journal of Productivity Analysis*, 3, 153-169.

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

Coelli, T. (1996) A Guide to FRONTIER Version 4.1: A Computer Program for Stochastic Frontier Production and Cost Function Estimation, CEPA Working Paper 96/08, http://www.uq.edu.au/economics/cepa/frontier.php, University of New England.

See Also

front41ReadOutput, front41Est

Examples

```
data( front41Data )
front41Data$logOutput  <- log( front41Data$output )
front41Data$logCapital <- log( front41Data$capital )
front41Data$logLabour  <- log( front41Data$labour )

front41WriteInput( front41Data, "firm", yName = "logOutput",
   xNames = c( "logCapital", "logLabour" ),
   path = tempdir(), insFile = "coelli.ins" )

## Not run:
system( paste0( "cd ", tempdir(), "; front41.bin coelli.ins" ) )
sfa <- front41ReadOutput( file.path( tempdir(), "coelli.out" ) )
summary( sfa )

## End(Not run)
```

---

frontierQuad *Quadratic or Translog Frontiers*

Description

This is a convenient interface for estimating quadratic or translog stochastic frontier functions using frontier.

Usage

```
frontierQuad( yName, xNames, shifterNames = NULL, zNames = NULL,
   data, lrTests = FALSE, ... )
```

Arguments

yName	string: name of the endogenous variable.
xNames	a vector of strings containing the names of the X variables (exogenous variables of the production or cost function) that should be included as linear, quadratic, and interaction terms.
shifterNames	a vector of strings containing the names of the X variables that should be included as shifters only (not in quadratic or interaction terms).
zNames	a vector of strings containing the names of the Z variables (variables explaining the efficiency level).
data	a (panel) data frame that contains the data; if data is a usual data.frame, it is assumed that these are cross-section data; if data is a panel data frame (created with pdata.frame), it is assumed that these are panel data.
lrTests	logical. If TRUE, likelihood ratio tests are conducted to test the statistical significance of each X variable.
...	further arguments passed to frontier.

Value

frontierQuad returns a list of class frontierQuad (and frontier) containing the same elements as returned by frontier. If argument lrTest is set to TRUE, the returned object has a component lrTests that contains the results of likelihood-ratio tests of the statistical significance of each X variable.

Author(s)

Arne Henningsen

See Also

frontier.

Examples

```
# example included in FRONTIER 4.1 (cross-section data)
data( front41Data )
front41Data$logOutput  <- log( front41Data$output )
front41Data$logCapital <- log( front41Data$capital )
front41Data$logLabour  <- log( front41Data$labour )

# estimate the translog function
translog <- frontierQuad( yName = "logOutput",
   xNames = c( "logCapital", "logLabour" ),
   data = front41Data )
translog

# estimate the same model using sfa()
translog2 <- sfa( logOutput ~ logCapital + logLabour
   + I( 0.5 * logCapital^2 ) + I( logCapital * logLabour )
```

```
   + I( 0.5 * logLabour^2 ), data = front41Data )
translog2
all.equal( coef( translog ), coef( translog2 ),
   check.attributes = FALSE )
```

`frontierTranslogRay` *Translog Ray Frontiers*

Description

This is a convenient interface for estimating translog stochastic ray frontier models using `frontier`.

Usage

```
frontierTranslogRay( yNames, xNames, shifterNames = NULL,
   zNames = NULL, data, ... )
```

Arguments

`yNames`	a vector of two or more character strings containing the names of the output variables.
`xNames`	a vector of strings containing the names of the input variables that should be included as linear, quadratic, and interaction terms.
`shifterNames`	a vector of strings containing the names of the explanatory variables that should be included as shifters only (not in quadratic or interaction terms).
`zNames`	a vector of strings containing the names of the Z variables (variables explaining the efficiency level).
`data`	a (panel) data frame that contains the data (see documentation of `frontier`) NOTE: the variables defined by arguments `yNames` and `xNames` must be in natural units; the variables defined by argument `xNames` are logarithmized internally; the variables defined by arguments `shifterNames` and `zNames` are NOT logarithmized internally and hence must be specified as they should be used in the model.
`...`	further arguments passed to `frontierQuad` and possibly further to `frontier`.

Value

`frontierTranslogRay` returns a list of class `frontierTranslogRay` (as well as `frontierQuad` and `frontier`) containing almost the same elements as returned by `frontier`. Additionally, it includes following objects:

`distance`	the "distance" from the origin (zero) to the point of the dependent variables.
`theta_i`	the "direction" from the origin (zero) to the point of the dependent variables (with `i` = $1, ..., N-1$ and N is the number of outputs).

Author(s)

Arne Henningsen and Geraldine Henningsen

References

LÃ¶thgren, M. (1997) Generalized stochastic frontier production models, *Economics Letters*, 57, 255-259.

LÃ¶thgren, M. (1997) *A Multiple Output Stochastic Ray Frontier Production Model*, Working Paper Series in Economics and Finance, No. 158, Stockholm School of Economics.

LÃ¶thgren, M. (2000) Specification and estimation of stochastic multiple-output production and technical inefficiency *Applied Economics*, 32, 1533-1540.

See Also

frontier, frontierQuad.

Examples

```
## preparing data
data( germanFarms )
# quantity of crop outputs
germanFarms$qCrop <- germanFarms$vCrop / germanFarms$pOutput
# quantity of animal outputs
germanFarms$qAnimal <- germanFarms$vAnimal / germanFarms$pOutput
# quantity of variable inputs
germanFarms$qVarInput <- germanFarms$vVarInput / germanFarms$pVarInput

# estimate a translog ray production function
estResultRay <- frontierTranslogRay( yNames = c( "qCrop", "qAnimal" ),
   xNames = c( "qLabor", "land", "qVarInput" ),
   data = germanFarms )
summary( estResultRay )
```

---

logLik.frontier *Extract Log-Likelihood Value*

Description

Extract the log-likelihood value(s) from stochastic frontier models returned by frontier.

Usage

```
## S3 method for class 'frontier'
logLik( object, which = "mle", newParam = NULL, ... )
```

Arguments

object	an object of class frontier (returned by the function frontier).
which	character string. Which log-likelihood value should be returned? 'ols' for the log-likelihood value of the parameters estimated by OLS, 'grid' for the log-likelihood value of the parameters obtained by the grid search (only if no starting values were provided), 'start' for the log-likelihood value of the starting values of the parameters specified by the user (only if starting values were provided), or 'mle' for the log-likelihood values of the parameters estimated by Maximum Likelihood.
newParam	optional vector of parameters. If this argument is provided by the user, the log-likelihood value of the model object is calculated with these (new) parameters.
...	currently unused.

Value

logLik.frontier returns an object of class logLik, which is a numeric scalar (the log-likelihood value) with 2 attributes: nobs (total number of observations in all equations) and df (number of free parameters, i.e. length of the coefficient vector).

Author(s)

Arne Henningsen

See Also

frontier.

Examples

```
# example included in FRONTIER 4.1
data( front41Data )

# SFA estimation with starting values obtained from a grid search
sfaResult <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data )
logLik( sfaResult, which = "ols" )
logLik( sfaResult, which = "grid" )
logLik( sfaResult )

# SFA estimation with starting values provided by the user
sfaResult2 <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data, startVal = 0.9 * coef( sfaResult ) )
logLik( sfaResult2, which = "ols" )
logLik( sfaResult2, which = "start" )
logLik( sfaResult2 )

# evaluate log likelihood function for a user-provided parameter vector
logLik( sfaResult, newParam = 0.9 * coef( sfaResult ) )
   # equal to  logLik( sfaResult2, which = "start" )
```

```
# log likelihood function for different values of gamma
plot( t( sapply( seq( 0.05, 0.95, 0.05 ), function(x) c( x,
   logLik( sfaResult, newParam = c( coef( sfaResult )[1:4], x ) ) ) ) ) )
```

`lrtest.frontier` *Likelihood Ratio test for Stochastic Frontier Models*

Description

Testing parameter restrictions in stochastic frontier models by a Likelihood Ratio test.

Usage

```
   ## S3 method for class 'frontier'
lrtest( object, ... )
```

Arguments

`object`	a fitted model object of class `frontier`.
`...`	further fitted model objects of class `frontier`.

Details

If `lrtest.frontier` is called with only one argument/object (i.e. argument `...` is not used), it compares the fitted model to a corresponding model without inefficiency (i.e. estimated by OLS).

If `lrtest.frontier` is called with more than one argument/object (i.e. argument `...` is used), it consecutively compares the fitted model object `object` with the models passed in `...`.

The test statistic is `2 * ( logLik( mu ) - logLik( mr ) )`, where `mu` is the unrestricted model and `mr` is the restricted model.

If a Frontier model (estimated by ML) is compared to a model without inefficiency (estimated by OLS), the test statistic asymptotically has a mixed χ^2 distribution under the null hypothesis (see Coelli, 1995).

If two Frontier models (estimated by ML) are compared, the test statistic asymptotically has a χ^2 distribution with j degrees of freedom under the null hypothesis, where j is the number of restrictions.

Value

An object of class `anova`, which contains the log-likelihood value, degrees of freedom, the difference in degrees of freedom, likelihood ratio Chi-squared statistic and corresponding p value. See documentation of `lrtest` in package "lmtest".

Author(s)

Arne Henningsen

References

Coelli, T.J. (1995), Estimators and Hypothesis Tests for a Stochastic: A Monte Carlo Analysis, *Journal of Productivity Analysis*, 6, 247-268.

See Also

sfa, lrtest

Examples

```
# rice producers in the Philippines (panel data)
data( "riceProdPhil" )
library( "plm" )
riceProdPhil <- pdata.frame( riceProdPhil, c( "FMERCODE", "YEARDUM" ) )

# Error Components Frontier with truncated normal distribution
# and time effects (unrestricted model)
mu <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   truncNorm = TRUE, timeEffect = TRUE, data = riceProdPhil )

# Error Components Frontier with half-normal distribution
# without time effects (restricted model)
mr <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil )

## compare the two models by an LR-test
lrtest( mu, mr )

## compare each of the models to a corresponding model without inefficiency
lrtest( mu )
lrtest( mr )
```

---

resettestFrontier *RESET test for Stochastic Frontier Models*

Description

Generalized Ramsey's RESET test (REgression Specification Error Test) for misspecification of the functional form based on a Likelihood Ratio test.

Usage

```
resettestFrontier( object, power = 2:3 )
```

Arguments

object	a fitted model object of class frontier.
power	a vector indicating the powers of the fitted variables that should be included as additional explanatory variables. By default, the test is for quadratic or cubic influence of the fitted response.

Value

An object of class anova as returned by `lrtest.frontier`.

Author(s)

Arne Henningsen

References

Ramsey, J.B. (1969), Tests for Specification Error in Classical Linear Least Squares Regression Analysis. *Journal of the Royal Statistical Society, Series B* 31, 350-371.

See Also

`sfa`, `resettest`, and `lrtest.frontier`

Examples

```
# load data set
data( front41Data )

# estimate a Cobb-Douglas production frontier
cobbDouglas <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data )

# conduct the RESET test
resettestFrontier( cobbDouglas )
```

---

residuals.frontier *Returning Residuals*

Description

This method returns the residuals from stochastic frontier models estimated with the **frontier** package (e.g. function `sfa`).

Usage

```
## S3 method for class 'frontier'
residuals( object, asInData = FALSE, ... )
```

Arguments

`object`	a stochastic frontier model estimated with the **frontier** package (e.g. function `sfa`).
`asInData`	logical. If `TRUE`, the residuals are returned in the same order as the corresponding observations in the data set used for the estimation (see section 'value' below).
`...`	currently ignored.

Value

If argument `asInData` is `FALSE` (default), a matrix of the residuals is returned, where each row corresponds to a firm (cross-section unit) and each column corresponds to a time period.

If argument `asInData` is `TRUE`, a vector of residuals is returned, where the residuals are in the same order as the corresponding observations in the data set used for the estimation.

Author(s)

Arne Henningsen

See Also

`sfa`, `residuals`.

Examples

```
# rice producers in the Philippines (panel data)
data( "riceProdPhil" )
library( "plm" )
riceProdPhil <- pdata.frame( riceProdPhil, c( "FMERCODE", "YEARDUM" ) )

# Error Components Frontier (Battese & Coelli 1992), no time effect
rice <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil )
residuals( rice )
riceProdPhil$residuals <- residuals( rice, asInData = TRUE )

# Error Components Frontier (Battese & Coelli 1992), with time effect
riceTime <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil, timeEffect = TRUE )
residuals( riceTime )
riceProdPhil$residualsTime <- residuals( riceTime, asInData = TRUE )
```

`riceProdPhil` *Rice Production in the Philippines*

Description

The `riceProdPhil` data frame contains annual data collected from 43 smallholder rice producers in the Tarlac region of the Philippines between 1990 and 1997.

Usage

```
data( riceProdPhil )
```

Format

This data frame contains the following variables (columns):

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

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

PROD Output (tonnes of freshly threshed rice).

AREA Area planted (hectares).

LABOR Labour used (man-days of family and hired labour).

NPK Fertiliser used (kg of active ingredients).

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

PRICE Output price (pesos per kg).

AREAP Rental price of land (pesos per hectare).

LABORP Labour 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 data set is published as supplement to Coelli et al. (2005). While most variables of this data set 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.

sfa *Stochastic Frontier Analysis*

Description

Maximum Likelihood Estimation of Stochastic Frontier Production and Cost Functions. Two specifications are available: the error components specification with time-varying efficiencies (Battese and Coelli 1992) and a model specification in which the firm effects are directly influenced by a number of variables (Battese and Coelli 1995). This R package uses the Fortran source code of Frontier 4.1 (Coelli 1996).

Usage

```
sfa( formula, data = sys.frame( sys.parent() ),
   ineffDecrease = TRUE, truncNorm = FALSE,
   timeEffect = FALSE, startVal = NULL,
   tol = 0.00001, maxit = 1000, muBound = 2, bignum = 1.0E+16,
   searchStep = 0.00001, searchTol = 0.001, searchScale = NA,
   gridSize = 0.1, gridDouble = TRUE,
   restartMax = 10, restartFactor = 0.999, printIter = 0 )

frontier( yName, xNames = NULL, zNames = NULL, data,
   zIntercept = FALSE, ... )

## S3 method for class 'frontier'
print( x, digits = NULL, ... )
```

Arguments

formula	a symbolic description of the model to be estimated; it can be either a (usual) one-part or a two-part formula (see section 'Details').
data	a (panel) data frame that contains the data; if data is a usual data.frame, it is assumed that these are cross-section data; if data is a panel data frame (created with pdata.frame), it is assumed that these are panel data.
ineffDecrease	logical. If TRUE, inefficiency decreases the endogenous variable (e.g. for estimating a production function); if FALSE, inefficiency increases the endogenous variable (e.g. for estimating a cost function).
truncNorm	logical. If TRUE, the inefficiencies are assumed to have a truncated normal distribution (i.e. parameter μ is added); if FALSE, they are assumed to have a half-normal distribution (only relevant for the 'Error Components Frontier').
timeEffect	logical. If FALSE (default), the efficiency estimates of an 'Error Components Frontier' are time invariant; if TRUE, time is allowed to have an effect on efficiency (this argument is ignored in case of an 'Efficiency Effects Frontier').
startVal	numeric vector. Optional starting values for the ML estimation.
tol	numeric. Convergence tolerance (proportional).

maxit	numeric. Maximum number of iterations permitted.
muBound	numeric. Bounds on the parameter μ (see 'details' section).
bignum	numeric. Used to set bounds on densities and distributions.
searchStep	numeric. Size of the first step in the Coggin uni-dimensional search procedure done each iteration to determine the optimal step length for the next iteration (see Himmelblau 1972).
searchTol	numeric. Tolerance used in the Coggin uni-dimensional search procedure done each iteration to determine the optimal step length for the next iteration (see Himmelblau 1972).
searchScale	logical or NA. Scaling in the Coggin uni-dimensional search procedure done each iteration to determine the optimal step length for the next iteration (see Himmelblau 1972): if TRUE, the step length is scaled to the length of the last step; if FALSE, the step length is not scaled; if NA, the step length is scaled (to the length of last step) only if the last step was smaller.
gridSize	numeric. The size of the increment in the first phase grid search on γ.
gridDouble	logical. If TRUE, a second phase grid search on γ is conducted around the "best" value obtained in the first phase with an increment of gridSize/10.
restartMax	integer: maximum number of restarts of the search procedure when it cannot find a parameter vector that results in a log-likelihood value larger than the log-likelihood value of the initial parameters.
restartFactor	numeric scalar: if the search procedure cannot find a parameter vector that results in a log-likelihood value larger than the log-likelihood value of the initial parameters, the initial values (provided by argument startVal or obtained by the grid serach) are multiplied by this number before the search procedure is restarted.
printIter	numeric. Print info every printIter iterations; if this argument is 0, do not print.
yName	string: name of the endogenous variable.
xNames	a vector of strings containing the names of the X variables (exogenous variables of the production or cost function).
zNames	a vector of strings containing the names of the Z variables (variables explaining the efficiency level).
zIntercept	logical. If TRUE, an intercept (with parameter δ_0) is added to the Z variables (only relevant for the 'Efficiency Effects Frontier').
x	an object of class frontier (returned by the function frontier).
digits	a non-null value for 'digits' specifies the minimum number of significant digits to be printed in values. The default, NULL, uses max(3,getOption("digits")-3). Non-integer values will be rounded down, and only values greater than or equal to 1 and no greater than 22 are accepted.
...	additional arguments of frontier are passed to sfa; additional arguments of the print method are currently ignored.

Details

Function frontier is a wrapper function that calls sfa for the estimation. The two functions differ only in the user interface; function frontier has the "old" user interface and is kept to maintain compatibility with older versions of the frontier package.

One can use functions sfa and frontier to calculate the log likelihood value for a given model, a given data set, and given parameters by using the argument startVal to specify the parameters and using the other arguments to specify the model and the data. The log likelihood value can then be retrieved by the logLik method with argument which set to "start". Setting argument maxit to 0 avoids the (eventually time-consuming) ML estimation and allows to retrieve the log likelihood value with the logLik method without further arguments.

The frontier function uses the Fortran source code of Tim Coelli's software FRONTIER 4.1 (http://www.uq.edu.au/economics/cepa/frontier.htm) and hence, provides the same features as FRONTIER 4.1. A comprehensive documentation of FRONTIER 4.1 is available in the file Front41.pdf that is included in the archive FRONT41-xp1.zip, which is available at http://www.uq.edu.au/economics/cepa/frontier.htm. It is recommended to read this documentation, because the frontier function is based on the FRONTIER 4.1 software.

If argument formula of sfa is a (usual) one-part formula (or argument zNames of frontier is NULL), an 'Error Components Frontier' (ECF, see Battese and Coelli 1992) is estimated. If argument formula is a two-part formula (or zNames is not NULL), an 'Efficiency Effects Frontier' (EEF, see Battese and Coelli 1995) is estimated. In this case, the first part of the formula (i.e. the part before the "|" symbol) is used to explain the endogenous variable directly (X variables), while the second part of the formula (i.e. the part after the "|" symbol) is used to explain the efficiency levels (Z variables). Generally, there should be no reason for estimating an EEF without Z variables, but this can done by setting the second part of argument formula to 1 (with Z intercept) or - 1 (without Z intercept) (or by setting argument zNames) to NA).

In case of an Error Components Frontier (ECF) with the inefficiency terms u following a truncated normal distribution with mean μ, argument muBound can be used to restrict μ to be in the interval $\pm$muBound * σ_u, where σ_u is the standard deviation of u. If muBound is infinity, zero, or negative, no bounds on μ are imposed.

Value

sfa and frontier return a list of class frontier containing following elements:

modelType	integer. A '1' denotes an 'Error Components Frontier' (ECF); a '2' denotes an 'Efficiency Effects Frontier' (EFF).
ineffDecrease	logical. Argument ineffDecrease (see above).
nn	number of cross-sections.
nt	number of time periods.
nob	number of observations in total.
nb	number of regressor variables (Xs).
truncNorm	logical. Argument truncNorm.
zIntercept	logical. Argument zIntercept.
timeEffect	logical. Argument timeEffect.
printIter	numeric. Argument printIter (see above).

searchScale	numeric. Argument searchScale (see above).
tol	numeric. Argument tol (see above).
searchTol	numeric. Argument searchTol (see above).
bignum	numeric. Argument bignum (see above).
searchStep	numeric. Argument searchStep (see above).
gridDouble	logical. Argument gridDouble (see above).
gridSize	numeric. Argument gridSize (see above).
maxit	numeric. Argument maxit (see above).
muBound	numeric. Argument muBound (see above).
restartMax	numeric. Argument restartMax (see above).
restartFactor	numeric. Argument restartFactor (see above).
nRestart	numeric. Number of restarts of the search procedure when it cannot find a parameter vector that results in a log-likelihood value larger than the log-likelihood value of the initial parameters.
startVal	numeric vector. Argument startVal (only if specified by user).
call	the matched call.
dataTable	matrix. Data matrix sent to Frontier 4.1.
olsParam	numeric vector. OLS estimates.
olsStdEr	numeric vector. Standard errors of OLS estimates.
olsLogl	numeric. Log likelihood value of OLS estimation.
olsResid	numeric vector. Residuals of the OLS estimation.
olsSkewness	numeric. Skewness of the residuals of the OLS estimation.
olsSkewnessOkay	logical. Indicating if the residuals of the OLS estimation have the expected skewness.
gridParam	numeric vector. Parameters obtained from the grid search (if no starting values were specified).
gridLogl	numeric. Log likelihood value of the parameters obtained from the grid search (only if no starting values were specified).
startLogl	numeric. Log likelihood value of the starting values for the parameters (only if starting values were specified).
mleParam	numeric vector. Parameters obtained from ML estimation.
mleCov	matrix. Covariance matrix of the parameters obtained from the OLS estimation.
mleLogl	numeric. Log likelihood value of the ML estimation.
nIter	numeric. Number of iterations of the ML estimation.
code	integer indication the reason for determination: 1 = log likelihood values and parameters of two successive iterations are within the tolerance limits; 5 = cannot find a parameter vector that results in a log-likelihood value larger than the log-likelihood value obtained in the previous step; 6 = search failed on gradient step; 10 = maximum number of iterations reached.

nFuncEval	Number of evaluations of the log likelihood function during the grid search and the iterative ML estimation.
fitted	matrix. Fitted "frontier" values of the dependent variable: each row corresponds to a cross-section; each column corresponds to a time period.
resid	matrix. Residuals: each row corresponds to a cross-section; each column corresponds to a time period.
validObs	vector of logical values indicating which observations of the provided data were used for the estimation, i.e. do not have values that are not available (NA, NaN) or infinite (Inf).

Author(s)

Tim Coelli and Arne Henningsen

References

Battese, G.E. and T. Coelli (1992), Frontier production functions, technical efficiency and panel data: with application to paddy farmers in India. *Journal of Productivity Analysis*, 3, 153-169.

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

Coelli, T. (1996) A Guide to FRONTIER Version 4.1: A Computer Program for Stochastic Frontier Production and Cost Function Estimation, CEPA Working Paper 96/08, http://www.uq.edu.au/economics/cepa/frontier.php, University of New England.

Himmelblau, D.M. (1972), *Applied Non-Linear Programming*, McGraw-Hill, New York.

See Also

frontierQuad for quadratic/translog frontiers, summary.frontier for creating and printing summary results, efficiencies.frontier for calculating efficiency estimates, lrtest.frontier for comparing models by LR tests, fitted.frontier for obtaining the fitted "frontier" values, ang residuals.frontier for obtaining the residuals.

Examples

```
# example included in FRONTIER 4.1 (cross-section data)
data( front41Data )

# Cobb-Douglas production frontier
cobbDouglas <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data )
summary( cobbDouglas )

# load data about rice producers in the Philippines (panel data)
data( riceProdPhil )

# Error Components Frontier (Battese & Coelli 1992)
# with observation-specific efficiencies (ignoring the panel structure)
rice <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil )
```

```
summary( rice )

# Error Components Frontier (Battese & Coelli 1992)
# with "true" fixed individual effects and observation-specific efficiencies
riceTrue <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ) +
   factor( FMERCODE ),  data = riceProdPhil )
summary( riceTrue )

# add data set with information about its panel structure
library( "plm" )
ricePanel <- pdata.frame( riceProdPhil, c( "FMERCODE", "YEARDUM" ) )

# Error Components Frontier (Battese & Coelli 1992)
# with time-invariant efficiencies
riceTimeInv <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = ricePanel )
summary( riceTimeInv )

# Error Components Frontier (Battese & Coelli 1992)
# with time-variant efficiencies
riceTimeVar <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = ricePanel, timeEffect = TRUE )
summary( riceTimeVar )

# Technical Efficiency Effects Frontier (Battese & Coelli 1995)
# (efficiency effects model with intercept)
riceZ <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ) |
   EDYRS + BANRAT, data = riceProdPhil )
summary( riceZ )

# Technical Efficiency Effects Frontier (Battese & Coelli 1995)
# (efficiency effects model without intercept)
riceZ2 <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ) |
   EDYRS + BANRAT - 1, data = riceProdPhil )
summary( riceZ2 )

# Cost Frontier (with land as quasi-fixed input)
riceProdPhil$cost <- riceProdPhil$LABOR * riceProdPhil$LABORP +
   riceProdPhil$NPK * riceProdPhil$NPKP
riceCost <- sfa( log( cost ) ~ log( PROD ) + log( AREA ) + log( LABORP )
   + log( NPKP ), data = riceProdPhil, ineffDecrease = FALSE )
summary( riceCost )
```

`summary.front41Output` *Summarizing the Estimation of Frontier 4.1*

Description

`summary.front41Output` summarizes the estimation results of a model estimated by Frontier 4.1..

Usage

```
## S3 method for class 'front41Output'
summary( object, ... )

## S3 method for class 'summary.front41Output'
print( x, efficiencies = FALSE, ... )
```

Arguments

object	an object of class front41Output (read/created by front41ReadOutput.
x	object of class summary.front41Output (returned by the summary method for objects of class front41ReadOutput).
efficiencies	logical. Print all efficiency estimates? (If FALSE, only the mean efficiency is printed.)
...	currently ignored.

Value

The summary method returns a list of class summary.front41Output with the same elements as object returned by front41ReadOutput. However, the elements olsResults, gridResults, and mleResults have an additional culumn with marginal significance levels (P values). The P values of the OLS estimates are calculated using the t distribution, while the (asymptotic) P values of the ML estimates are calculated based on the assumption that their t values follow an (asymptotic) standard normal distribution.

Author(s)

Arne Henningsen

See Also

front41ReadOutput, front41WriteInput.

Examples

```
# read the output file that is provided with Frontier 4.1
outFile <- system.file( "front41/EG1.OUT", package = "frontier" )
sfa <- front41ReadOutput( outFile )
summary( sfa )
```

summary.frontier *summary method for class frontier*

Description

Create and print summary results of a stochastic frontier analysis returned by frontier.

Usage

```
## S3 method for class 'frontier'
summary( object, extraPar = FALSE, effic = FALSE,
   logDepVar = TRUE, effMinusU = farrell, farrell = TRUE, ... )
## S3 method for class 'summary.frontier'
print( x, effic = x$printEffic, ... )
```

Arguments

object	an object of class frontier (returned by the function frontier).
x	an object of class summary.frontier (returned by the function summary.frontier).
extraPar	logical. If TRUE, some additional parameters, their standard errors, z-values, and P values are returned: sigmaSqU = sigmaSq * gamma (with $u \sim N^{+}$(mu, sigmaSqU)), sigmaSqV = sigmaSq * (1 - gamma) (with $v \sim$ N(0, sigmaSqV)), sigma = sigmaSq^0.5, sigmaU = sigmaSqU^0.5, sigmaV = sigmaSqV^0.5, lambdaSq = sigmaSqU / sigmaSqV, and lambda = sigmaU / sigmaV. Please note that sigmaSqU and sigmaU are not the variance and standard error, respectively, of u. If the model is an error components frontier, also the following additional parameters are returned: varU = the variance of u, sdU = varU^0.5, and gammaVar = varU / (varU + sigmaSqV). Please note that the variance of u usually differs between observations if the model is an error component frontier with 'time effect' or an efficiency effects frontier.
effic	logical. Print the individual efficiency estimates?
logDepVar	logical. Is the dependent variable logged?
effMinusU	logical. If TRUE (the default), the efficiencies are calculated by E[exp(-u)]. If FALSE, the efficiencies are calculated by E[exp(u)]. For details, see documentation of argument minusU of efficiencies.frontier.
farrell	logical. This argument is only kept for backward compatibility and will be removed in the future.
...	further arguments to the summary method are currently ignored; further arguments to the print method are forwarded to printCoefmat.

Details

The standard errors of the estimated parameters are taken from the direction matrix that is used in the final iteration of the Davidon-Fletcher-Powell procedure that is used for maximising the (log) likelihood function.

If argument `extraPar` is `TRUE`, the standard errors of the additional parameters are obtained by the delta method. Please note that the delta method might provide poor approximations of the 'true' standard errors, because parameter σ^2 is left-censored and parameter γ is both left-censored and right-censored so that these parameters cannot be normally distributed.

Please note further that the t statistic and the z statistic are not reliable for testing the statistical signicance of σ^2, γ, and the 'additional parameters', because these parameters are censored and cannot follow a normal distribution or a t distribution.

Value

`summary.frontier` returns a list of class `summary.frontier` that is identical to an object returned by `frontier` with two modifications and (up to) four additional elements:

`olsParam`	matrix of OLS estimates, their standard errors, t-values, and P-values.
`mleParam`	matrix of ML estimates, their standard errors, z-values, and asymptotic P-values.
`logDepVar`	logical. Argument `logDepVar` (see above).
`printEffic`	argument `effic`.
`effic`	matrix. Efficiency estimates: each row corresponds to a cross-section; each column corresponds to a time period.
`efficMean`	numeric scalar. Mean efficiency.
`efficYearMeans`	numeric vector. Mean efficiency for each year in the sample (only for panel data but not for the Error Components Frontier without time effects).

Author(s)

Arne Henningsen

See Also

`sfa`, `efficiencies.frontier`, `vcov.frontier`, and `lrtest.frontier`.

Examples

```
# example included in FRONTIER 4.1 (cross-section data)
data( front41Data )

sfaResult <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data )
summary( sfaResult )

# rice producers in the Phillipines (panel data)
data( "riceProdPhil" )
library( "plm" )
riceProdPhil <- pdata.frame( riceProdPhil, c( "FMERCODE", "YEARDUM" ) )

# Error Components Frontier
rice <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ),
   data = riceProdPhil )
summary( rice )
```

```
# Efficiency Effects Frontier
rice2 <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ) |
   EDYRS + BANRAT, data = riceProdPhil )
summary( rice2 )
```

`telecom` *Telecommunications Providers*

Description

The `telecom` data frame contains data on telecommunications providers in 21 countries in 1990.

Usage

```
data( telecom )
```

Format

This data frame contains the following variables (columns):

country The name of the country.

output Output (index).

mainlines Mainlines (M km).

employees Number of employees (10,000 persons).

Source

Supplementary files for Coelli et al. (1998), p. 193.

References

Coelli, T. J., Rao, D. S. P., and Battese, G. E. (1998) *An Introduction to Efficiency and Productivity Analysis*, Springer, New York.

`vcov.frontier` *vcov method for class frontier*

Description

Extract the covariance matrix of the maximum likelihood coefficients of a stochastic frontier model returned by `frontier`.

Usage

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

Arguments

`object`	an object of class `frontier` (returned by the function `frontier`).
`extraPar`	logical. If `TRUE`, the variances and covariances of additional parameters are returned: `sigmaSqU` = `sigmaSq` * `gamma` (with $u \sim N^{+}($ `mu`, `sigmaSqU` $)$), `sigmaSqV` = `sigmaSq` * (1 - `gamma`) (with $v \sim$ N(0, `sigmaSqV`)), `sigma` = `sigmaSq`^0.5, `sigmaU` = `sigmaSqU`^0.5, `sigmaV` = `sigmaSqV`^0.5, `lambdaSq` = `sigmaSqU` / `sigmaSqV`, and `lambda` = `sigmaU` / `sigmaV`. Please note that `sigmaSqU` and `sigmaU` are not the variance and standard error, respectively, of u.
`...`	currently unused.

Details

The variance-covariance matrix of the estimated parameters is taken from the direction matrix that is used in the final iteration of the Davidon-Fletcher-Powell procedure that is used for maximising the (log) likelihood function.

If argument `extraPar` is `TRUE`, the variances and covariances of the additional parameters are obtained by the delta method. Please note that the delta method might provide poor approximations of the 'true' variances and covariances, because parameter σ^2 is left-censored and parameter γ is both left-censored and right-censored so that these parameters cannot be normally distributed.

Please note further that it might not be appropriate to use standard statistical tests (e.g. t-tests or other Wald tests) that are based on the variances and covariances of σ^2, γ, and the 'additional parameters', because these parameters are censored and cannot follow normal distributions.

Value

`vcov.frontier` returns the covariance matrix of the maximum likelihood coefficients.

Author(s)

Arne Henningsen

See Also

coef.frontier, coef.summary.frontier, summary.frontier, and sfa.

Examples

```
# example included in FRONTIER 4.1
data( front41Data )

sfaResult <- sfa( log( output ) ~ log( capital ) + log( labour ),
   data = front41Data )
vcov( sfaResult )
```

Index