

Package ‘frontier’

November 11, 2009

Version 0.996-0

Date 2009-11-11

Title Stochastic Frontier Analysis

Author Tim Coelli, Arne Henningsen

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

Depends R (>= 2.0.0), plm (>= 1.0-1), micEcon (>= 0.5-16), Formula (>= 0.2-0)

Imports moments (>= 0.11), lmtest (>= 0.9-24)

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).

License GPL (>= 2)

URL <http://www.uq.edu.au/economics/cepa/>,
<http://frontier.r-forge.r-project.org/>

Repository CRAN

Date/Publication 2009-11-11 15:03:07

R topics documented:

coef.front41Output	2
coef.frontier	3
coef.summary.frontier	4
efficiencies	5
efficiencies.frontier	5
elas.frontierQuad	7
front41Data	8
front41Est	9
front41ReadOutput	10

front41WriteInput	12
frontierQuad	15
frontierTranslogRay	16
logLik.frontier	18
lrtest.frontier	19
residuals.frontier	21
riceProdPhil	22
sfa	23
summary.front41Output	28
summary.frontier	30
telecom	31
vcov.frontier	32
Index	33

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 <code>front41Output</code> or <code>summary.front41Output</code> (read/created by <code>front41ReadOutput</code> or <code>summary.front41Output</code> , 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 `front41Output` returns a vector containing all coefficients estimated by Frontier 4.1.

The `coef` method applied to an object of class `summary.front41Output` 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](#)

<code>coef.frontier</code>	<i>coef method for class frontier</i>
----------------------------	---------------------------------------

Description

Extract the coefficients from stochastic frontier models returned by [frontier](#).

Usage

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

Arguments

<code>object</code>	an object of class <code>frontier</code> (returned by the function frontier).
<code>which</code>	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).
<code>...</code>	currently unused.

Value

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

Author(s)

Arne Henningsen

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

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

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

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 ) )
```

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, ... )
```

Arguments

<code>object</code>	a stochastic frontier model returned by <code>frontier</code> .
<code>asInData</code>	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).
<code>logDepVar</code>	logical. Is the dependent variable logged?
<code>...</code>	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.

Author(s)

Arne Henningsen <arne.henningsen@gmail.com>

See Also

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

Examples

```
# rice producers in the Philippines (panel data)  
data( riceProdPhil )  
riceProdPhil <- plm.data( 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 )

# 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 )

```

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 <code>frontierQuad</code> (returned by <code>frontierQuad</code>).
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 <code>yName</code> and <code>xNames</code> and available in argument <code>data</code>) already logged? (If argument <code>dataLogged</code> is <code>TRUE</code> , the frontier function is of the translog form; if argument <code>dataLogged</code> is <code>FALSE</code> , the frontier function is quadratic).
yObs	logical. Use observed values of the endogenous variable. If <code>FALSE</code> (default) predicted values calculated by <code>quadFuncCalc</code> are used (ignored if argument <code>dataLogged</code> is <code>TRUE</code>).
...	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 )$ela
```

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.

capital capital.

labour labour.

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.htm>, 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

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

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:

<code>input</code>	object returned by <code>front41WriteInput</code> .
<code>messages</code>	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.htm>, 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 available as (FORTRAN) source code and (executable) binaries for GNU/Linux and MS-Windows.

Value

a list of class `front41Output` containing following objects:

<code>version</code>	the version of Frontier 4.1 that produced the output.
<code>insFile</code>	name of the instruction file used by Frontier 4.1.
<code>dtaFile</code>	name of the data file used by Frontier 4.1.
<code>modelType</code>	model type: either 1 for 'Error Components Frontier' or 2 for 'Tech. Eff. Effects Frontier'.
<code>modelTypeName</code>	model type: 'Error Components Frontier' or 'Tech. Eff. Effects Frontier'.
<code>functionType</code>	function type: either 1 for 'production function' or 2 for 'cost function'.
<code>functionTypeName</code>	function type: 'production function' or 'cost function'.
<code>logDepVar</code>	logical. Is the dependent variable logged.
<code>olsResults</code>	results of the OLS estimation.
<code>nXvars</code>	number X variables (exogenous variables of the production or cost function.
<code>olsLogl</code>	log likelihood value of the OLS estimation.
<code>gridResults</code>	results of the grid search.
<code>mleResults</code>	results of the maximum likelihood estimation.
<code>mleLogl</code>	log likelihood value of the maximum likelihood estimation.
<code>mleCov</code>	coefficient covariance matrix of the maximum likelihood estimation.
<code>lrTest</code>	LR test of the one-sided error.
<code>lrTestRestrict</code>	number of restrictions of the LR test.
<code>nIter</code>	number of iterations.
<code>maxIter</code>	maximum number of iterations set.
<code>nCross</code>	number of cross-sections.
<code>nPeriods</code>	umber of time periods.
<code>nObs</code>	total number of observations.
<code>nObsMissing</code>	number of observations that are not in the panel.
<code>efficiency</code>	technical efficiency estimates.
<code>meanEfficiency</code>	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.htm>, 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" ), insFile = "coelli.ins" )

## Not run:
system( "front41.bin coelli.ins" )
sfa <- front41ReadOutput( "coelli.out" )
print( 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,
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, igrd2 = 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).
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.
igrd2	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 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:

<code>nCrossSection</code>	number of cross section units.
<code>nTimePeriods</code>	number of time periods.
<code>nTotalObs</code>	total number of observations.
<code>nXtotal</code>	total number of X variables (including quadratic and interaction terms).
<code>nZvars</code>	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.htm>, 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" ), insFile = "coelli.ins" )

## Not run:
system( "front41.bin coelli.ins" )
sfa <- front41ReadOutput( "coelli.out" )

## End(Not run)
```

frontierQuad	<i>Quadratic or Translog Frontiers</i>
--------------	--

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, quadHalf = TRUE, ... )
```

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 <code>plm.data</code>), it is assumed that these are panel data.
quadHalf	logical. Multiply the quadratic terms by one half?
...	further arguments passed to <code>frontier</code> .

Value

`frontierQuad` returns a list of class `frontierQuad` (and `frontier`) containing the same elements as returned by `frontier` and additionally the element `quadHalf` which is equal to the argument `quadHalf`.

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 )

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

frontierTranslogRay

Translog Ray Frontiers

Description

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

Usage

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

Arguments

<code>yNames</code>	a vector of two character strings containing the names of the two dependent variables.
<code>xNames</code>	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.
<code>shifterNames</code>	a vector of strings containing the names of the X variables that should be included as shifters only (not in quadratic or interaction terms).
<code>zNames</code>	a vector of strings containing the names of the Z variables (variables explaining the efficiency level).
<code>data</code>	a (panel) data frame that contains the data (see documentation of <code>frontier</code>) NOTE: the variables in <code>yNames</code> and <code>xNames</code> are logarithmized internally and hence must be in natural units, while the variables in <code>shifterNames</code> and <code>zNames</code> are NOT logarithmized internally and hence must be specified as they should be used in the model.
<code>...</code>	further arguments passed to <code>frontierQuad</code> and possibly further to <code>frontier</code> .

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:

<code>distance</code>	the “distance” from the origin (zero) to the point of the dependent variables.
<code>theta</code>	the “direction” from the origin (zero) to the point of the dependent variables.

Author(s)

Arne Henningsen and Geraldine Henningsen

References

- LÄŧhgren, M. (1997) Generalized stochastic frontier production models, *Economics Letters*, 57, 255-259.
- LÄŧhgren, M. (1997) *A Multiple Output Stochastic Ray Frontier Production Model*, Working Paper Series in Economics and Finance, No. 158, Stockholm School of Economics.
- LÄŧhgren, 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 <code>frontier</code> (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 <code>object</code> 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 $2 * (\text{logLik}(\text{mu}) - \text{logLik}(\text{mr}))$, 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 <arne.henningsen@gmail.com>

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 )
riceProdPhil <- plm.data( 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 )
```

residuals.frontier *Returning Efficiency Estimates*

Description

This method returns the residuals from stochastic frontier models estimated with [sfa](#) or [frontier](#).

Usage

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

Arguments

object	a stochastic frontier model returned by frontier .
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

[frontier](#), [residuals](#).

Examples

```
# rice producers in the Philippines (panel data)
data( riceProdPhil )
riceProdPhil <- plm.data( 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	<i>Rice Production in the Philippines</i>
--------------	---

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/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.

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, printIter = 0 )

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

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

Arguments

<code>formula</code>	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’).
<code>data</code>	a (panel) data frame that contains the data; if <code>data</code> is a usual <code>data.frame</code> , it is assumed that these are cross-section data; if <code>data</code> is a panel data frame (created with <code>plm.data</code>), it is assumed that these are panel data.
<code>ineffDecrease</code>	logical. If <code>TRUE</code> , inefficiency decreases the endogenous variable (e.g. for estimating a production function); if <code>FALSE</code> , inefficiency increases the endogenous variable (e.g. for estimating a cost function).
<code>truncNorm</code>	logical. If <code>TRUE</code> , the inefficiencies are assumed to have a truncated normal distribution (i.e. parameter μ is added); if <code>FALSE</code> , they are assumed to have a half-normal distribution (only relevant for the ‘Error Components Frontier’).
<code>timeEffect</code>	logical. If <code>FALSE</code> (default), the efficiency estimates of an ‘Error Components Frontier’ are time invariant; if <code>TRUE</code> , time is allowed to have an effect on efficiency (this argument is ignored in case of an ‘Efficiency Effects Frontier’).
<code>startVal</code>	numeric vector. Optional starting values for the ML estimation.
<code>tol</code>	numeric. Convergence tolerance (proportional).
<code>maxit</code>	numeric. Maximum number of iterations permitted.
<code>muBound</code>	numeric. Bounds on the parameter μ (see ‘details’ section).
<code>bignum</code>	numeric. Used to set bounds on densities and distributions.
<code>searchStep</code>	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).
<code>searchTol</code>	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).
<code>searchScale</code>	logical or <code>NA</code> . 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 <code>TRUE</code> , the step length is scaled to the length of the last step; if <code>FALSE</code> , the step length is not scaled; if <code>NA</code> , the step length is scaled (to the length of last step) only if the last step was smaller.
<code>gridSize</code>	numeric. The size of the increment in the first phase grid search on γ .
<code>gridDouble</code>	logical. If <code>TRUE</code> , a second phase grid search on γ is conducted around the “best” value obtained in the first phase with an increment of <code>gridSize/10</code> .
<code>printIter</code>	numeric. Print info every <code>printIter</code> iterations; if this argument is 0, do not print.
<code>yName</code>	string: name of the endogenous variable.

<code>xNames</code>	a vector of strings containing the names of the X variables (exogenous variables of the production or cost function).
<code>zNames</code>	a vector of strings containing the names of the Z variables (variables explaining the efficiency level).
<code>zIntercept</code>	logical. If TRUE, an intercept (with parameter δ_0) is added to the Z variables (only relevant for the ‘Efficiency Effects Frontier’).
<code>x</code>	an object of class <code>frontier</code> (returned by the function <code>frontier</code>).
<code>digits</code>	a non-null value for ‘digits’ specifies the minimum number of significant digits to be printed in values. The default, NULL, uses <code>max(3, getOption("digits") - 3)</code> . Non-integer values will be rounded down, and only values greater than or equal to 1 and no greater than 22 are accepted.
<code>...</code>	additional arguments of <code>frontier</code> are passed to <code>sfa</code> ; additional arguments of the <code>print</code> 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-xpl.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 be 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 $\pm\text{muBound} * \sigma_u$, 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:

<code>modelType</code>	integer. A '1' denotes an 'Error Components Frontier' (ECF); a '2' denotes an 'Efficiency Effects Frontier' (EFF).
<code>ineffDecrease</code>	logical. Argument <code>ineffDecrease</code> (see above).
<code>nn</code>	number of cross-sections.
<code>nt</code>	number of time periods.
<code>nob</code>	number of observations in total.
<code>nb</code>	number of regressor variables (Xs).
<code>truncNorm</code>	logical. Argument <code>truncNorm</code> .
<code>zIntercept</code>	logical. Argument <code>zIntercept</code> .
<code>timeEffect</code>	logical. Argument <code>timeEffect</code> .
<code>printIter</code>	numeric. Argument <code>printIter</code> (see above).
<code>searchScale</code>	numeric. Argument <code>searchScale</code> (see above).
<code>tol</code>	numeric. Argument <code>tol</code> (see above).
<code>searchTol</code>	numeric. Argument <code>searchTol</code> (see above).
<code>bignum</code>	numeric. Argument <code>bignum</code> (see above).
<code>searchStep</code>	numeric. Argument <code>searchStep</code> (see above).
<code>gridDouble</code>	logical. Argument <code>gridDouble</code> (see above).
<code>gridSize</code>	numeric. Argument <code>gridSize</code> (see above).
<code>maxit</code>	numeric. Argument <code>maxit</code> (see above).
<code>startVal</code>	numeric vector. Argument <code>startVal</code> (only if specified by user).
<code>call</code>	the matched call.
<code>dataTable</code>	matrix. Data matrix sent to Frontier 4.1.
<code>olsParam</code>	numeric vector. OLS estimates.
<code>olsStdEr</code>	numeric vector. Standard errors of OLS estimates.
<code>olsLogl</code>	numeric. Log likelihood value of OLS estimation.
<code>olsResid</code>	numeric vector. Residuals of the OLS estimation.
<code>olsSkewness</code>	numeric. Skewness of the residuals of the OLS estimation.
<code>olsSkewnessOkay</code>	logical. Indicating if the residuals of the OLS estimation have the expected skewness.
<code>gridParam</code>	numeric vector. Parameters obtained from the grid search (if no starting values were specified).
<code>gridLogl</code>	numeric. Log likelihood value of the parameters obtained from the grid search (only if no starting values were specified).
<code>startLogl</code>	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.
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 <arne.henningsen@gmail.com>

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.htm>, 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, and [lrtest.frontier](#) for comparing models by LR tests.

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 )

# rice producers in the Philippines (panel data)
data( riceProdPhil )
riceProdPhil <- plm.data( riceProdPhil, c( "FMERCODE", "YEAR" ) )

# Error Components Frontier (Battese & Coelli 1992)
rice <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ) ,
  data = riceProdPhil )
```

```

summary( rice )

# Technical Efficiency Effects Frontier (Battese & Coelli 1995)
rice2 <- sfa( log( PROD ) ~ log( AREA ) + log( LABOR ) + log( NPK ) |
  EDYRS + BANRAT - 1, data = riceProdPhil )
summary( rice2 )

# 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 )

#####
### using frontier() with the "old" user interface ###
#####

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

# Cobb-Douglas production frontier
cobbDouglasFront <- frontier( yName = "logOutput",
  xNames = c( "logCapital", "logLabour" ), data = front41Data )
all.equal( cobbDouglasFront, cobbDouglas ) # only the call (element 34)
# and the names of the parameters are slightly different

# rice producers in the Philippines (panel data)
riceProdPhil$lPROD <- log( riceProdPhil$PROD )
riceProdPhil$lAREA <- log( riceProdPhil$AREA )
riceProdPhil$lLABOR <- log( riceProdPhil$LABOR )
riceProdPhil$lNPK <- log( riceProdPhil$NPK )

# Error Components Frontier (Battese & Coelli 1992)
riceFront <- frontier( yName = "lPROD",
  xNames = c( "lAREA", "lLABOR", "lNPK" ), data = riceProdPhil )
all.equal( riceFront, rice ) # only the call (element 34)
# and the names of the parameters are slightly different

# Technical Efficiency Effects Frontier (Battese & Coelli 1995)
rice2Front <- frontier( yName = "lPROD",
  xNames = c( "lAREA", "lLABOR", "lNPK" ),
  zNames = c( "EDYRS", "BANRAT" ), data = riceProdPhil )
all.equal( rice2Front, rice2 ) # only the call (element 34)
# and the names of the parameters are slightly different

```

```
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 column 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, effic = FALSE,
         logDepVar = TRUE, ... )
## S3 method for class 'summary.frontier':
print( x, effic = x$printEffic, ... )
```

Arguments

object	an object of class <code>frontier</code> (returned by the function frontier).
x	an object of class <code>summary.frontier</code> (returned by the function <code>summary.frontier</code>).
effic	logical. Print the individual efficiency estimates?
logDepVar	logical. Is the dependent variable logged?
...	currently unused.

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 <code>logDepVar</code> (see above).
printEffic	argument <code>effic</code> .
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 <arne.henningsen@gmail.com>

See Also

[sfa](#), [efficiencies.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 )
riceProdPhil <- plm.data( 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, ... )
```

Arguments

`object` an object of class `frontier` (returned by the function `frontier`).
`...` currently unused.

Value

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

Author(s)

Arne Henningsen

Examples

```
# example included in FRONTIER 4.1  
data( front41Data )  
  
sfaResult <- sfa( log( output ) ~ log( capital ) + log( labour ),  
          data = front41Data )  
vcov( sfaResult )
```

Index

*Topic **datasets**

front41Data, 8
riceProdPhil, 22
telecom, 31

*Topic **methods**

coef.front41Output, 2
efficiencias, 5
efficiencias.frontier, 5
residuals.frontier, 21

*Topic **models**

coef.frontier, 3
coef.summary.frontier, 4
elas.frontierQuad, 7
front41Est, 9
front41ReadOutput, 10
front41WriteInput, 12
frontierQuad, 15
frontierTranslogRay, 16
logLik.frontier, 18
lrtest.frontier, 19
sfa, 23
summary.front41Output, 28
summary.frontier, 30
vcov.frontier, 32

coef.front41Output, 2
coef.frontier, 3
coef.summary.front41Output
(*coef.front41Output*), 2
coef.summary.frontier, 4

efficiencias, 5, 6
efficiencias.frontier, 5, 5, 27, 30
elas.frontierQuad, 7

front41Data, 8
front41Est, 9, 12, 15
front41ReadOutput, 2, 9, 10, 10, 15, 29
front41WriteInput, 9, 10, 12, 12, 29
frontier, 3, 5, 6, 15–19, 21, 30, 32

frontier(*sfa*), 23
frontierQuad, 7, 15, 17, 27
frontierTranslogRay, 16

logLik, 25
logLik.frontier, 18
lrtest, 20
lrtest.frontier, 19, 27, 30

plm.data, 16, 24
print.front41Output
(*front41ReadOutput*), 10
print.frontier(*sfa*), 23
print.summary.front41Output
(*summary.front41Output*), 28
print.summary.frontier
(*summary.frontier*), 30

quadFuncCalc, 7
quadFuncEla, 7

residuals, 21
residuals.frontier, 21
riceProdPhil, 22

sfa, 6, 20, 21, 23, 30
summary.front41Output, 2, 28
summary.frontier, 4, 6, 27, 30

telecom, 31
translogEla, 7

vcov.front41Output
(*coef.front41Output*), 2
vcov.frontier, 32