**Description**

`sfa` is used to fit stochastic frontier analysis models.

**Details**

```
Package: sfa
Type: Package
Version: 0.1-0
Date: 2010-08-09
License: GPL-2
```

The package implements stochastic frontier analysis models as introduced by Aigner et al. (1977) and Battese and Coelli (1992, 1995).

**Author(s)**

Ariane Straub
under the supervision of Torsten Hothorn <Ariane.Straub@gmail.com> (package maintainer).

**References**


---

**dgp**

*Sample data generating process*

**Description**

Sample data generating process

**Usage**

```
dgp(n, b, intercept = TRUE, sc = -1)
```
**Arguments**

- `n`: sample size
- `b`: parameter vector
- `intercept`: logical, TRUE includes intercept
- `sc`: form of the frontier model, -1 for cost frontier model, 1 for production frontier model

**Value**

list

**See Also**

- `sfa`, `rnorm`, `runif`, `abs`

---

**Description**

generic function to create efficiencies

**Usage**

eff(object, ...)

**Arguments**

- `object`: a sfa model
- `...`: ignored

**Value**

The form of the value returned by efficiencies depends on the class of its argument. See the documentation of the particular methods for details of what is produced by that method.

**See Also**

eff.sfa
**LogLik**

*The negative log likelihood function of the SFA*

**Description**

The negative log likelihood function is used for estimating the parameters. It varies depending on the distribution of the inefficiency term $u$. $L_{hN}$ is used by halfnormal distribution of $u$. $L_{exp}$ is used by exponential distribution of $u$. $L_{trunc}$ is used by truncated normal distribution of $u$. $L_{trunc\_mufest}$ is used by truncated normal distribution of $u$ and constant $mu$.

**Usage**

- $L_{hN}(p, y = y, X = X, sc = sc)$
- $L_{exp}(p, y = y, X = X, sc = sc)$
- $L_{trunc}(p, y = y, X = X, sc = sc)$
- $L_{trunc\_mufest}(p, mu = mu, y = y, X = X, sc = sc)$

**Arguments**

- $p$: vector with the parameters to estimate
- $y$: response
- $X$: design matrix of the covariables
- $sc$: specifies the form of the frontier model (-1 = cost, 1 = production)
- $mu$: if known, the parameter $mu$

**Value**

returns the value of the log likelihood function

**See Also**

- *sfa*

---

**Description**

`coef.sfa` is used to display the fitted coefficients. `print.sfa` is used to display some information about the fitted SFA. `predict.sfa` is used to predict (new) data with the fitted SFA model. `fitted.sfa` is used to predict the original data with the fitted SFA model. `logLik.sfa` is used to display the value of the log likelihood function. `residuals.sfa` is used to return the residuals of the fitted SFA model. `summary.sfa` is used to calculate the summary result of the SFA. `print.summary.sfa` is used to display the summary result of the SFA. `eff.sfa` is used to return the efficiencies of the SFA.
Usage

```r
## S3 method for class 'sfa'
coef(object, ...)  # S3 method for class 'sfa'
print(x, ...)      # S3 method for class 'sfa'
predict(object, newdata = NULL, intercept = NULL, ...)  # S3 method for class 'sfa'
fitted(object, ...)  # S3 method for class 'sfa'
logLik(object, ...)  # S3 method for class 'sfa'
residuals(object, ...)  # S3 method for class 'sfa'
summary(object, ...)  # S3 method for class 'sfa'
print.summary(x, ...)  # S3 method for class 'sfa'
eff(object, ...)  # S3 method for class 'sfa'
```

Arguments

- `x` an object of class `sfa`
- `object` an object of class `sfa`
- `newdata` a data frame. If `newdata = NULL` then original data will be used.
- `intercept` boolean or `NULL`. If `intercept = NULL` then the function uses the same intercept options as specified in `sfa`.
- `...` ignored.

Examples

```r
set.seed(225)
daten <- dgp(n = 100, b = c(1, 2), sc = -1)
test <- sfa(y ~ x, data = daten)
coef(test)
print(test)
predict(test)
fitted(test)
logLik(test)
residuals(test)
summary(test)
eff(test)
```
Description

`sfa` is used to fit stochastic frontier analysis models.

Usage

```r
sfa(formula, data = NULL, intercept = TRUE, fun = "hnormal", pars = NULL, par_mu = NULL, form = "cost", method = "BFGS", ...)
```

Arguments

- `formula`: an object of class `formula` (or one that can be coerced to that class): a symbolic description of the model to be fitted.
- `data`: a data frame.
- `intercept`: logical. If true the model includes intercept.
- `fun`: specifies the distribution for the inefficiency term u as half-normal ("hnormal"), exponential ("exp"), or truncated-normal ("tnormal").
- `pars`: initial values for the parameters to be estimated.
- `par_mu`: value for mu in the normal-/truncated-normal case. If mu is known.
- `form`: specifies the form of the frontier model as "cost" or "production".
- `method`: the method to be used. See `optim` for more details.
- `...`: ignored.

Value

`sfa` returns an object of class `sfa`:

- `y`: response
- `x`: covariables
- `X`: design matrix
- `coef`: coefficients
- `sigmav2`: sigmav2
- `sigmav2`: sigmav2
- `mu`: mu of the truncated-normal distribution (Only if fun = tnormal)
- `par_mu`: NULL if mu is not estimated
- `logLik`: value of the log likelihood function
- `maxLik`: log likelihood function
- `fun`: distribution of the inefficiency term u
- `sc`: specifies the form of the frontier model (-1 = cost, 1 = production)
- `hess`: a symmetric matrix giving an estimate of the Hessian at the solution found (See `optim`)
- `ols`: the linear model for the LR-test
Examples

```r
set.seed(225)
daten <- dgp(n = 100L, b = c(1, 2), sc = -1)
test <- sfa(y ~ x, data = daten)
```

---

`te.eff.sfa`  
*technical efficiencies of sfa objects*

Description

returns the technical efficiencies of sfa objects

Usage

```r
te.eff.sfa(object, ...)
```

Arguments

- `object`: object of class sfa
- `...`: ignored

Value

returns the technical efficiencies of each observation

See Also

`eff.sfa, te.eff.sfa`

---

`u.sfa`  
*Inefficiencies of a sfa-object*

Description

returns the absolute inefficiencies of a sfa-object.

Usage

```r
u.sfa(object, ...)
```

Arguments

- `object`: an object of class sfa
- `...`: ignored
Value
returns the absolute inefficiencies of each observation

See Also

eff.sfa, te.eff.sfa
Index

*Topic data
  dgp, 2
*Topic function
  LogLik, 4
*Topic methods
  methods.sfa, 4
*Topic models
  sfa, 6
*Topic package
  sfa-package, 2

abs, 3
coef.sfa (methods.sfa), 4
dgp, 2
eff, 3
eff.sfa, 3, 7, 8
eff.sfa (methods.sfa), 4
fitted.sfa (methods.sfa), 4
formula, 6
L_exp (LogLik), 4
L_hNV (LogLik), 4
L_trunc (LogLik), 4
L_trunc_mufest (LogLik), 4
LogLik, 4
logLik.sfa (methods.sfa), 4
methods.sfa, 4
optim, 6
predict.sfa (methods.sfa), 4
print.sfa (methods.sfa), 4
print.summary.sfa (methods.sfa), 4
residuals.sfa (methods.sfa), 4
rnorm, 3
runif, 3
sfa, 3, 4, 6
sfa-package, 2
summary.sfa (methods.sfa), 4
teff.sfa, 7, 7, 8
u.sfa, 7