

# Package ‘sfa’

May 9, 2026

**Version** 1.0.4

**Date** 2026-01-15

**Title** Stochastic Frontier Analysis

**Type** Package

**Maintainer** David Bernstein <davebernstein1@gmail.com>

**Description** Provides a user-friendly framework for estimating a wide variety of cross-sectional and panel stochastic frontier models. Suitable for a broad range of applications, the implementation offers extensive flexibility in specification and estimation techniques.

**Suggests** knitr, MASS, rmarkdown, pracma, testthat

**Imports** devtools, pso, cubature, moments, readxl, haven, fdrtool, numDeriv, gsl, Hmisc, plm, minqa, randtoolbox, matrixStats, frontier, Jmisc, mnormt, truncnorm, tmvtnorm, Formula, methods

**Depends** R (>= 4.4.0)

**License** GPL (>= 2)

**Language** en-US

**URL** <https://www.davidharrybernstein.com/software>

**LazyLoad** yes

**NeedsCompilation** yes

**Archs** i386, x64

**VignetteBuilder** knitr

**Author** David Bernstein [aut, cre] (ORCID: <<https://orcid.org/0000-0002-2267-5741>>),  
Christopher Parmeter [aut],  
Alexander Stead [aut]

**Repository** CRAN

**Date/Publication** 2026-01-21 19:00:02 UTC

## Contents

sfa-package	2
data_gen_cs	3
data_gen_p	6
FinnishElec	8
Indian	9
panel89	10
print.sfareg	12
psfm	13
sfm	15
summary.sfareg	18
USUtilities	19
zsfm	20
<b>Index</b>	<b>23</b>

---

sfa-package	<i>Stochastic Frontier Analysis</i>
-------------	-------------------------------------

---

## Description

Provides a user-friendly framework for estimating a wide variety of cross-sectional and panel stochastic frontier models. Suitable for a broad range of applications, the implementation offers extensive flexibility in specification and estimation techniques.

## Details

The DESCRIPTION file:

```

Package:          sfa
Version:          1.0.4
Date:             2026-01-15
Title:            Stochastic Frontier Analysis
Type:             Package
Authors@R:        c(person("David", "Bernstein", email = "davebernstein1@gmail.com", role = c("aut", "cre"), comment =
Maintainer:       David Bernstein <davebernstein1@gmail.com>
Description:      Provides a user-friendly framework for estimating a wide variety of cross-sectional and panel stochastic
Suggests:         knitr, MASS, rmarkdown, pracma, testthat
Imports:          devtools, pso, cubature, moments, readxl, haven, fdrtool, numDeriv, gsl, Hmisc, plm, minqa, randtoolbox
Depends:          R (>= 4.4.0)
License:          GPL (>= 2)
Language:         en-US
URL:              https://www.davidharrybernstein.com/software
LazyLoad:         yes
NeedsCompilation: yes
Archs:            i386, x64
VignetteBuilder: knitr

```

Author: David Bernstein [aut, cre] (ORCID: <<https://orcid.org/0000-0002-2267-5741>>), Christopher Parmeter [

#### Index of help topics:

FinnishElec	FinnishElec
Indian	Indian
USUtilities	USUtilities
data_gen_cs	Generate Cross-Sectional Data for Stochastic Frontier Analysis
data_gen_p	Generate Panel Data for Stochastic Frontier Analysis
panel89	Panel89
print.sfareg	sfa Object Summaries
psfm	psfm
sfa-package	Stochastic Frontier Analysis
sfm	sfm
summary.sfareg	sfa Object Summaries
zsfm	Zero-Inflated Stochastic Frontier Model

#### See Also

<http://www.davidharrybernstein.com/software>

#### Examples

```
## Simple application of the generalized true random effects estimator.
library(sfa)

data_trial <- data_gen_p(t=10,N=100, rand = 100,
                        sig_u = 1, sig_v = 0.3,
                        sig_r = .2, sig_h = .4,
                        cons = 0.5, beta1 = 0.5,
                        beta2 = 0.5)

psfm(formula = y_gtre ~ x1 + x2,
      model_name = "GTRE",
      data = data_trial,
      individual = "name",
      PSopt = FALSE)
```

## Description

data\_gen\_cs generates simulated cross-sectional data based on the stochastic frontier model, allowing for different distributional assumptions for the one-sided technical inefficiency error term ( $u$ ) and the two-sided idiosyncratic error term ( $v$ ). The model has the general form:  $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + v - u$  where  $u \geq 0$  and represents inefficiency. All variants are produced so that the user can select those that they want.

## Usage

```
data_gen_cs(N, rand, sig_u, sig_v, cons, beta1, beta2, a, mu)
```

## Arguments

N	A single integer specifying the number of observations (cross-sectional units).
rand	A single integer to set the seed for the random number generator, ensuring reproducibility.
sig_u	The standard deviation parameter ( $\sigma_u$ ) for the base distribution of the one-sided error term $u$ .
sig_v	The standard deviation parameter ( $\sigma_v$ ) for the base distribution of the two-sided error term $v$ .
cons	The value of the constant term (intercept) in the model.
beta1	The coefficient for the $x_1$ variable.
beta2	The coefficient for the $x_2$ variable.
a	The degrees of freedom parameter for the t half-t distribution ( $u\_t$ and $v\_t$ , respectively). Requires the <code>rt</code> function.
mu	The mean parameter ( $\mu$ ) for the normal truncated normal distribution ( $u\_tn$ ). Requires the <code>rtruncnorm</code> function.

## Details

The function simulates two explanatory variables,  $x_1$  and  $x_2$ , as transformations of uniform random variables.

The function generates several different frontier models by combining various distributions for  $u$  and  $v$ :

- ***u* Distributions (Inefficiency):** Half-Normal (HN), Truncated Normal (TN), Half-T (HT), Half-Cauchy (HC), Exponential (E), Half-Uniform (HU).
- ***v* Distributions (Idiosyncratic):** Normal (N), t, Cauchy (C).

**Specific Model Outputs ( $y\_pcs$  variants):**

- $y\_pcs$ : Normal-Half Normal (N-HN):  $v \sim N(0, \sigma_v^2)$ ,  $u \sim |N(0, \sigma_u^2)|$ .
- $y\_pcs\_z$ : N-HN with Heteroskedastic  $\sigma_u$ :  $\sigma_{u,i} = \exp(0.9 + 0.6Z_i)$ , where  $Z$  is a uniform variable.
- $y\_pcs\_t$ : T-Half T (T-HT):  $v \sim T(df = a) \cdot \sigma_v$ ,  $u \sim |T(df = a)| \cdot \sigma_u$ .
- $y\_pcs\_tn$ : Normal-Truncated Normal (N-TN):  $v \sim N(0, \sigma_v^2)$ ,  $u \sim TN(\mu, \sigma_u^2)$  on  $[0, \infty)$ .

- `y_pcs_e`: Normal-Exponential (N-E):  $v \sim N(0, \sigma_v^2)$ ,  $u \sim \text{Exp}(\phi)$ , where  $\phi = 1/\sigma_u$ .
- `y_pcs_c`: Cauchy-Half Cauchy (C-HC):  $v \sim \text{Cauchy}(0, \sigma_v)$ ,  $u \sim |\text{Cauchy}(0, \sigma_u)|$ .
- `y_pcs_u`: Normal-Half Uniform (N-HU):  $v \sim N(0, \sigma_v^2)$ ,  $u \sim U(0, \sigma_u)$ .
- `y_pcs_w`: Normal + Cauchy - Half Normal:  $v \sim N(0, \sigma_v^2) + \text{Cauchy}(0, \sigma_v)$ ,  $u \sim |N(0, \sigma_u^2)|$ . This introduces a composite  $v$  term.

**\*\*Note:\*\*** The `rtruncnorm` function is required for `y_pcs_tn` and loads with the package. In isolation it could be loaded by using `library(truncnorm)`.

### Value

A data frame containing  $N$  observations with the following columns:

<code>name</code>	Individual identifier (simply 1 to $N$ ).
<code>cons</code>	The constant term value.
<code>x1</code>	Simulated explanatory variable $x_1$ .
<code>x2</code>	Simulated explanatory variable $x_2$ .
<code>u, uz, u_t, u_c, u_e, u_u, u_tn</code>	The simulated one-sided error terms under different distributions.
<code>v, v_t, v_c</code>	The simulated two-sided error terms under different distributions.
<code>y_pcs, y_pcs_t, y_pcs_e, y_pcs_c, y_pcs_u, y_pcs_z, y_pcs_w, y_pcs_tn</code>	The dependent variable $Y$ under the corresponding SFA model distributions.
<code>z</code>	The auxiliary variable used for heteroskedasticity in <code>y_pcs_z</code> .
<code>con</code>	A constant column set to 1, potentially for use in estimation.

### Author(s)

David Bernstein

### See Also

[rnorm](#), [runif](#), [rt](#), [rexp](#), [rcauchy](#), [rtruncnorm](#) (if available).

### Examples

```
# Generate 100 observations of SFA data
data_sfa <- data_gen_cs(
  N      = 100,
  rand   = 123,
  sig_u  = 0.5,
  sig_v  = 0.2,
  cons   = 5,
  beta1  = 1.5,
  beta2  = 2.0,
  a      = 5, # degrees of freedom for T/Half-T
  mu     = 0.1 # mean for Truncated Normal
)
```

```
# Display the first few rows of the generated data
head(data_sfa)

# Example of a Normal-Half Normal SFA model data
summary(data_sfa$y_pcs)
plot(density(data_sfa$y_pcs))
```

---

data\_gen\_p

*Generate Panel Data for Stochastic Frontier Analysis*


---

## Description

data\_gen\_p generates simulated panel data for estimating various panel stochastic frontier models, including the Generalized True Random Effects (GTRE), True Random Effects (TRE), Pooled Cross-Section (PCS), and True Fixed Effects (TFE) models. The function returns the data as a `pdata.frame`. All variants are produced so that the user can select those that they want.

## Usage

```
data_gen_p(t, N, rand, sig_u, sig_v, sig_r, sig_h, cons, tau = 0.5, mu = 0, beta1, beta2)
```

## Arguments

t	The number of time periods.
N	The number of individuals.
rand	A seed for the random number generator to ensure reproducibility.
sig_u	The standard deviation ( $\sigma_u$ ) for the one-sided error component ( $u_{it}$ ).
sig_v	The standard deviation ( $\sigma_v$ ) for the two-sided error component ( $v_{it}$ ).
sig_r	The standard deviation ( $\sigma_r$ ) for the two-sided individual effect ( $r_i$ ).
sig_h	The standard deviation ( $\sigma_h$ ) for the one-sided individual effect ( $h_i$ ).
cons	The constant term ( $\beta_0$ ) for the frontier models.
tau	The dependence parameter ( $\tau$ ) used for the <code>y_tfe</code> (TFE) model formulation, default is 0.5. See Chen, Schmidt, and Wang (2014, Journal of Econometrics).
mu	The mean parameter ( $\mu$ ) used for the Truncated-Normal (TN) component of the <code>y_fd</code> model with default set to 0. See Wang and Ho (2010, Journal of Econometrics).
beta1	The coefficient for the <code>x1</code> variable ( $\beta_1$ ).
beta2	The coefficient for the <code>x2</code> variable ( $\beta_2$ ).

## Details

A `pdata.frame` object with  $N \times t$  observations, containing the following columns:

- `name` Individual identifier.
- `year` Time period identifier.
- `cons` The constant term used in the data generation.
- `x1`, `x2` Explanatory variables generated from a log-uniform distribution.
- `x1_w`, `x2_w` Explanatory variables with dependence parameter  $\tau$  and linkage with  $r_i$ , used for the TFE model.
- `u`, `v`, `r`, `h` The generated error and individual effect components.
- `y_gtre`, `y_tre`, `y_pcs`, `y_tfe` Output variables for the Production Frontier models, including the constant.
- `y_gtre_nc`, `y_tre_nc`, `y_pcs_nc` Output variables for the Production Frontier models, **excluding** the constant.
- `c_gtre`, `c_tre`, `c_pcs`, `c_tfe` Output variables for the Cost Frontier models, including the constant.
- `c_gtre_nc`, `c_tre_nc`, `c_pcs_nc` Output variables for the Cost Frontier models, **excluding** the constant.
- `y_fd` Output variable for the first difference model (see Wang and Ho, 2010).
- `x_fd` Explanatory variable for the `y_fd` model.
- `u_fd_star`, `z_fd`, `r_fd`, `u_fd` Components used to generate `y_fd`.
- `u_gtre`, `z_gtre`, `y_gtre_z`, `y_tre_z` Variables for models with heteroskedastic inefficiency ( $\sigma_{u,i} = \exp(0.9 + 0.6Z_i)$ ).

The data is generated based on standard Stochastic Frontier Analysis (SFA) formulations, primarily for a **Production Frontier** where the one-sided error component  $u_{it}$  is subtracted:

- `y_gtre`: GTRE model:  $y_{it} = \beta_0 + \beta_1 x_{1,it} + \beta_2 x_{2,it} + r_i - h_i + v_{it} - u_{it}$
- `y_tre`: TRE model:  $y_{it} = \beta_0 + \beta_1 x_{1,it} + \beta_2 x_{2,it} + r_i + v_{it} - u_{it}$
- `y_pcs`: PCS model:  $y_{it} = \beta_0 + \beta_1 x_{1,it} + \beta_2 x_{2,it} + v_{it} - u_{it}$
- `y_tfe`: TFE model:  $y_{it} = \beta_1 x_{1,it}^w + \beta_2 x_{2,it}^w + r_i + v_{it} - u_{it}$
- `y_gtre_z`: GTRE with Heteroskedastic  $u_{it}$ :  $\sigma_{u,i} = \exp(0.9 + 0.6Z_i)$ .

For **Cost Frontier** models, the one-sided error component  $u_{it}$  is added (e.g., `c_gtre`).

The error terms are generated as:

- $r_i \sim N(0, \sigma_r^2)$  (individual two-sided effect)
- $h_i \sim |N(0, \sigma_h^2)|$  (individual one-sided effect)
- $v_{it} \sim N(0, \sigma_v^2)$  (two-sided noise)
- $u_{it} \sim |N(0, \sigma_u^2)|$  (one-sided inefficiency)

The First-Difference estimation model (`y_fd`) uses a variation where  $r_{i,fd} \sim U(0, 1)$  and  $u_{it,fd}$  is generated using a heteroskedastic truncated-normal structure, reflecting an alternative model type.

**Value**

A `pdata.frame` object containing  $N \times t$  observations suitable for Stochastic Frontier Analysis (SFA).

**Author(s)**

David Bernstein

**References**

Chen, Y., Schmidt, P., & Wang, H. (2014). Consistent estimation of the fixed effects stochastic frontier model. *Journal of Econometrics*, 181(2), 65-76.

Filippini, M., & Greene, W. H. (2016). Persistent and transient productive inefficiency: a maximum simulated likelihood approach. *Journal of Productivity Analysis*, 45, 187-196.

Wang, H., & Ho, C. M. (2010). Estimating fixed-effect panel stochastic frontier models by model transformation. *Journal of Econometrics*, 157(2), 286-296.

**See Also**

`data_gen_p`, to see all the data generating processes

**Examples**

```
library(sfa)
# Generate a dataset
data_trial <- data_gen_p(t=10, N=100, rand = 100,
                        sig_u = 1, sig_v = 0.3,
                        sig_r = .2, sig_h = .4,
                        cons = 0.5, tau = 0.5,
                        mu = 0.5, beta1 = 0.5,
                        beta2 = 0.5)
# See the first few rows
head(data_trial)
```

---

FinnishElec

*FinnishElec*

---

**Description**

Cross-sectional data on Finnish electricity distribution firms, including annual averages of expenditure and output measures over a four-year regulatory period.

**Usage**

```
data("FinnishElec")
```

**Format**

A data frame with 89 observations on the following 6 variables.

id a character vector containing a unique identifier for each distribution firm

x a numeric vector containing total expenditure (TOTEX\*) (1000 Euros)

y1 a numeric vector containing weighted energy transmitted through the network (GWh of 0.4 kV equivalents)

y2 a numeric vector containing total length of the network (km)

y3 a numeric vector containing total number of customers connected to the network

z a numeric vector containing the proportion of underground cables in the total network length.

**Details**

\*TOTEX includes capital expenditure (CAPEX), controllable operational expenditure (OPEX), and estimated external cost of interruptions.

**Source**

Kuosmanen, T. (2012). 'Stochastic semi-nonparametric frontier estimation of electricity distribution networks: Application of the StoNED method in the Finnish regulatory model.' *Energy Economics*, 34(6), pp. 2189-2199. doi:10.1016/j.eneco.2012.03.005

**Examples**

```
data(FinnishElec)
plot(FinnishElec)
```

---

Indian

*Indian*

---

**Description**

Panel data on 14 paddy farmers from Aurepalle, India, collected over ten years (1975-76 to 1984-85). Includes farmer characteristics (age, schooling) and production variables (output, land, labor, bullocks, input costs).

**Usage**

```
data("Indian")
```

**Format**

A data frame with 273 observations (an unbalanced panel of 34 farmers over 10 years) on the following 10 variables.

`id` a numeric vector containing a unique identifier for each farmer

`yr` a numeric vector containing the year of the observation

`age` a numeric vector containing the age of the primary decision maker

`school` a numeric vector containing the number of years of schooling of the primary decision maker

`yvar` a numeric vector containing the natural logarithm of the total value of output (rupees)

`Lland` a numeric vector containing the natural logarithm of the total area of land operated (ha)

`PIland` a numeric vector containing the proportion of land that is irrigated

`Llabor` a numeric vector containing the natural logarithm of the total number of hours of hired and family labour used

`Lbull` a numeric vector containing the natural logarithm of the number of hours of bullock labour used

`Lcost` a numeric vector containing the natural logarithm of the value of inputs including fertilizer, manure, pesticides, machinery, etc.

**Source**

Battese, G.E. and Coelli, T.J. (1995) 'A model for technical inefficiency effects in a stochastic frontier production function for panel data', *Empirical Economics*, 20(2), pp. 325-332. doi:10.1007/BF01205442.

**References**

Battese, G.E. and Coelli, T.J. (1992) 'Frontier production functions, technical efficiency and panel data: With application to paddy farmers in India', *Journal of Productivity Analysis*, 3(1-2), pp. 153-169. doi:10.1007/BF00158774.

**Examples**

```
data(Indian)
```

---

panel89

*Panel89*

---

**Description**

The dataset is a cross-section of U.S. commercial banks for 1989, extracted from the panel dataset used by Kumbhakar, Parmeter and Tsionas (2013) and based on the Federal Reserve Bank of Chicago's Reports of Condition and Income. It contains detailed cost data with inputs and outputs defined under the intermediation approach, and input prices constructed as expense-quantity ratios.

**Usage**

```
data("panel89")
```

**Format**

A data frame with 4,985 observations on the following 11 variables.

y a numeric vector containing the natural logarithm of total cost\*  
q1 a numeric vector containing the natural logarithm of installment loans  
q2 a numeric vector containing the natural logarithm of real estate loans  
q3 a numeric vector containing the natural logarithm of business loans  
q4 a numeric vector containing the natural logarithm of federal funds sold and securities purchased  
q5 a numeric vector containing the natural logarithm of other assets  
w1 a numeric vector containing the natural logarithm of the price of labour\*  
w2 a numeric vector containing the natural logarithm of the price of capital\*  
w3 a numeric vector containing the natural logarithm of the price of purchased funds\*  
w4 a numeric vector containing the natural logarithm of the price of interest-bearing deposits in total transaction accounts\*  
z a numeric vector containing the natural logarithm of total assets

**Details**

\*The cost and input price variables are normalised by that of a fifth input: the price of interest-bearing deposits in total non-transaction accounts. Total cost is defined as the sum of total expenses for each input. Input prices are derived by dividing the total expense for each input by the corresponding input quantity.

**Source**

Kumbhakar, S.C., Parmeter, C.F. and Tsionas, E.G. (2013) 'A zero inefficiency stochastic frontier model', *Journal of Econometrics*, 172(1), pp. 66-76. doi:10.1016/j.jeconom.2012.08.021.

**References**

Kumbhakar, S.C. and Tsionas, E.G. (2005) 'Measuring technical and allocative inefficiency in the translog cost system: a Bayesian approach', *Journal of Econometrics*, 126(2), pp. 355-384. doi:10.1016/j.jeconom.2004.05.006.

**Examples**

```
data(panel89)  
plot(panel89)
```

---

`print.sfareg`*sfa Object Summaries*

---

**Description**

print function for stochastic frontier models of `sfm()`, `zsfm()`, and `psfm()` calls.

**Usage**

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

**Arguments**

`x` sfa regression objects of the `sfm()`, `zsfm()`, and `psfm()` calls.  
`...` Additional arguments passed to other methods

**Details**

Allows for the usage of `print()`

**Value**

No return value, called for side effects

**Author(s)**

David H. Bernstein

**Examples**

```
library(sfa)  
  
cs_data_trial <- data_gen_cs(N= 1000, rand = 1, sig_u = 0.3, sig_v = 0.3,  
  cons = 0.5, beta1 = 0.5, beta2 = 0.5, a = 4, mu = 1)  
  
cs.nhnz <- sfm(formula = y_pcs_z ~ x1 +x2| z, model_name = "NHN",  
  data = cs_data_trial, PSopt = TRUE)  
print(cs.nhnz)
```

---

 psfm

 psfm
 

---

## Description

Function to implement various panel data stochastic frontier estimators

## Usage

```
psfm(formula, model_name = c("TRE_Z", "GTRE_Z", "TRE",
                             "GTRE", "TFE", "FD", "GTRE_SEQ1", "GTRE_SEQ2"), data,
      maxit.bobyqa = 100, maxit.psoptim = 10, maxit.optim =
      10, REPORT = 1, trace = 3, pgtol = 0, individual,
      halton_num = NULL, start_val = FALSE, gamma = FALSE,
      PSopt = FALSE, optHessian, inefdec= TRUE, Method = "L-BFGS-B",
      verbose = FALSE, rand.gtre = NULL, rand.psoptim = NULL)
```

## Arguments

formula	a symbolic description for the model to be estimated
model_name	model name for the estimation
data	a pdata.frame
maxit.bobyqa	Maximum number of iterations for the bobyqa optimization routine
maxit.psoptim	Maximum number of iterations for the psoptim optimization routine
maxit.optim	Maximum number of iterations for the optim optimization routine
REPORT	reporting parameter
trace	trace
pgtol	pgtol
individual	individual unit in the regression model
halton_num	number of Halton draws to use in SML models
start_val	starting value (optional)
gamma	gamma
PSopt	use psoptim optimization routine (T or F)
optHessian	Logical. Should a numerically differentiated Hessian matrix be returned while using the optim routine? (for optim routine)
inefdec	Production or cost function
Method	The method to be used for optim. See 'Details' within optim.
verbose	Logical. Print optimization progress messages? Default is FALSE.
rand.psoptim	Integer. Seed for replication of psoptim. Default to NULL.
rand.gtre	Integer. Seed for replication of the gtre model. Default to NULL.

## Details

The generalized true random effects model (GTRE, 4-component model) and true random effects models (TRE) are both estimated by simulated maximum likelihood based on the paper by the Fillipini and Greene (2016, JPA). The TRE\_Z and GTRE\_Z allow for modeling the u-component of the GTRE and TRE with determinants of inefficiency. The first-difference estimator (FD) of Wang and Ho (2010, JoE) as well as the True Fixed Effect model estimated by within-maximum likelihood of Chen, Schmidt and Wang (2014, JoE) are also available.

## Value

An object of class "sfareg" containing components that vary by model. All models return:

out	A matrix with parameter estimates, standard errors, and t-values.
opt	A list containing the optimization results from the final optimization procedure (not returned for GTRE_SEQ1 and GTRE_SEQ2).
total_time	The total computation time for model estimation.
start_v	The starting values used in the optimization (not returned for GTRE_SEQ1 and GTRE_SEQ2).
model_name	The name of the panel stochastic frontier model estimated.
formula	The formula used in the model specification.
coefficients	A vector of estimated parameters.
std.errors	A vector of standard errors for the estimated parameters (NA if optHessian = FALSE).
t.values	A vector of t-values for the estimated parameters (NA if optHessian = FALSE).
call	The matched call.
data	The data used in estimation.

### Additional model-specific components:

#### For GTRE and GTRE\_Z models:

H Predicted time-invariant technical efficiency for each individual.

#### For GTRE, GTRE\_Z, TRE and TRE\_Z models:

U Predicted time-varying technical efficiency for each observation.

#### For TFE model:

r\_hat\_m Estimated individual-specific random effects.

exp\_u\_hat Predicted technical efficiency.

#### For FD model:

u\_hat Predicted technical efficiency in levels.

h\_hat Estimated z heterogeneity function values.

exp\_u\_hat Predicted technical efficiency.

#### For GTRE\_SEQ1 and GTRE\_SEQ2 models:

other\_parms A matrix of additional parameters (lambda, sigma, beta\_0 for SEQ1; sigma\_u, sigma\_v, sigma\_h, sigma\_r, lambda, sigma for SEQ2).

**Note**

Standard errors require `optHessian` set to `TRUE`

**Note**

The `GTRE_SEQ1` and `GTRE_SEQ2` models use sequential estimation methods and do not return optimization objects or starting values. All panel models require the `individual` argument to identify panel units.

**Author(s)**

David Bernstein

**References**

Fillipini and Greene (2016, JPA); Wang and Ho (2010, JoE); Chen, Schmidt and Wang (2014, JoE)

**See Also**

see also

**Examples**

```
library(sfa)

data_trial <- data_gen_p(t=10,N=100, rand = 100,
                        sig_u = 1, sig_v = 0.3,
                        sig_r = .2, sig_h = .4,
                        cons = 0.5, beta1 = 0.5,
                        beta2 = 0.5)

max_tre_z <- psfm(formula = y_tre_z ~ x1 +x2| z_gtre,
                  model_name = "TRE",                ## "TRE_Z" also works
                  data = data_trial,
                  individual = "name",
                  PSopt = TRUE)
```

---

sfm

*sfm*

---

**Description**

Implementation of the cross-sectional stochastic frontier model across an array of distributional assumptions for both  $v$  and  $u$  (user specified). For panel models, see the `psfm()` call.

**Usage**

```
sfm(formula, model_name, data, maxit.bobyqa, maxit.psoptim, maxit.optim, REPORT,
    trace, pgtol, start_val, PSopt, optHessian, inefdec, upper, Method, eta, alpha, verbose=FALSE,
    rand.psoptim=NULL)
```

**Arguments**

formula	a symbolic description for the model to be estimated
model_name	model name for the estimation includes the: normal-half normal (NHN), normal-exponential (NE), student's t-half t (THT), Normal-Rayleigh (NR), and the normal-truncated normal (NTN).
data	A data set
maxit.bobyqa	Maximum number of iterations for the bobyqa optimization routine
maxit.psoptim	Maximum number of iterations for the psoptim optimization routine
maxit.optim	Maximum number of iterations for the optim optimization routine
REPORT	reporting parameter
trace	trace
pgtol	pgtol
start_val	starting value (optional)
PSopt	use psoptim optimization routine (T or F)
optHessian	Logical. Should a numerically differentiated Hessian matrix be returned while using the optim routine? (for optim routine)
inefdec	Production or cost function
upper	Vector of upper values for the optim package.
Method	The method to be used for optim. See 'Details' within optim.
eta	Parameter used for psi-divergence.
alpha	Parameter used for MDPD.
verbose	Logical. Print optimization progress messages? Default is FALSE.
rand.psoptim	Integer. seed for replication of psoptim. Default to NULL.

**Details**

The options include the Normal-Half Normal (NHN), Normal-exponential (NE), Student's t-Half t (THT), and the Normal-Truncated Normal (NTN). NHN\_Z and NE\_Z are extensions for the NHN and NE models that allow for modeling the u-component of those models with determinants of inefficiency.

Outputs include  $E[\exp(-u)|e]$  given by  $\exp\_u\_hat$ , following Battese and Coelli (1988, JoE), where appropriate.

**Value**

An object of class "sfareg" containing the following components:

out	A matrix with parameter estimates, standard errors, and t-values.
opt	A list containing the optimization results from the final optimization procedure.
total_time	The total computation time for model estimation.
start_v	The starting values used in the optimization.
model_name	The name of the stochastic frontier model estimated.
formula	The formula used in the model specification.
exp_u_hat	Predicted technical efficiency (expected values). Available for models: NHN, NHN_Z, NR, NG, and NNAK.
med_u_hat	Predicted technical efficiency (median values). Available only for the NHN model.
coefficients	A vector of estimated parameters.
std.errors	A vector of standard errors for the estimated parameters (NA if optHessian = FALSE).
t.values	A vector of t-values for the estimated parameters (NA if optHessian = FALSE).
call	The matched call.

**Note**

Standard errors require optHessian set to TRUE

**Author(s)**

David H. Bernstein and Alexander Stead

**See Also**

see also

**Examples**

```
library(sfa)

cs_data_trial <- data_gen_cs(N= 1000, rand = 1, sig_u = 0.3, sig_v = 0.3,
  cons = 0.5, beta1 = 0.5, beta2 = 0.5, a = 4, mu = 1)

cs.nhnz <- sfm(formula = y_pcs_z ~ x1 +x2| z, model_name = "NHN",
  data = cs_data_trial, PSopt = TRUE)
```

---

summary.sfareg      *sfa Object Summaries*

---

## Description

Summary function for stochastic frontier models of sfm(), zsfm(), and psfm() calls.

## Usage

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

## Arguments

object            sfa regression objects of the sfm(), zsfm(), and psfm() calls.  
...                Additional arguments passed to other methods

## Details

Allows for the usage of summary()

## Value

prints while returning the sfareg object

## Author(s)

David Bernstein

## Examples

```
library(sfa)  
  
cs_data_trial <- data_gen_cs(N= 1000, rand = 1, sig_u = 0.3, sig_v = 0.3,  
  cons = 0.5, beta1 = 0.5, beta2 = 0.5, a = 4, mu = 1)  
  
cs.nhnz <- sfm(formula = y_pcs_z ~ x1 +x2| z, model_name = "NHN",  
  data = cs_data_trial, PSopt = TRUE)  
summary(cs.nhnz)
```

---

USUtilities

*USUtilities*

---

**Description**

Panel data on U.S. investor-owned fossil fuel-fired steam electric utilities for the period 1986-1999. These data include measures of output, capital, labour and maintenance, and fuel.

**Usage**

```
data("USUtilities")
```

**Format**

A data frame with 972 observations (a balanced panel of observations on 81 utilities over 12 years) on the following 7 variables.

`firmID` a numeric vector containing a unique firm identifier

`year` a numeric vector containing the year of the observation

`q` a numeric vector containing net steam electric power generation (MWh)

`K` a numeric vector containing capital stock, calculated using a method described by Christensen and Jorgenson (1970)

`L` a numeric vector containing quantity of labor and maintenance, calculated as cost divided by price index

`F` a numeric vector containing quantity of fuel used, calculated as fuel costs divided by fuel price index

`trend` a numeric vector containing an annual time trend (1992=100)

**Details**

The dataset covers 72 investor-owned utilities after aggregating subsidiaries and excluding plants in states with partial deregulation plans. Data sources include the Energy Information Administration (EIA), Federal Energy Regulatory Commission (FERC), and Bureau of Labor Statistics (BLS). Output is net steam electric generation from fossil fuel-fired boilers.

**Source**

Rungsuriyawiboon, S. and Stefanou, S.E. (2007). 'Dynamic Efficiency Estimation: An Application to U.S. Electric Utilities.' *Journal of Business & Economic Statistics*, 25(2), pp. 226-238. doi:10.1198/073500106000000288

**References**

Christensen, L.R. and Jorgenson, D.W. (1970). 'U.S. Real Product and Real Factor Input, 1928-1967.' *Review of Income and Wealth*, 16(1), pp. 19-50. doi: 10.1111/j.1475-4991.1970.tb00695.x

**Examples**

```
data(USUtilities)
```

---

zsfm

*Zero-Inflated Stochastic Frontier Model*


---

**Description**

Code to use the Zero-Inflated Stochastic Frontier Model

**Usage**

```
zsfm(formula, model_name = c("ZISF", "ZISF_Z"),
      data, maxit.bobyqa = 10000, maxit.psoptim = 1000, maxit.optim = 1000,
      REPORT = 1, trace = 0, pgtol = 0, start_val = FALSE, PSopt = FALSE,
      optHessian, inefdec = TRUE, upper = NA,
      Method = "L-BFGS-B", logit = TRUE, verbose=FALSE, rand.psoptim = NULL)
```

**Arguments**

formula	a symbolic description for the model to be estimated
model_name	model name for the estimation
data	A data set
maxit.bobyqa	Maximum number of iterations for the bobyqa optimization routine
maxit.psoptim	Maximum number of iterations for the psoptim optimization routine
maxit.optim	Maximum number of iterations for the optim optimization routine
REPORT	reporting parameter
trace	trace
pgtol	pgtol
start_val	starting value (optional)
PSopt	use psoptim optimization routine (T or F)
optHessian	Logical. Should a numerically differentiated Hessian matrix be returned while using the optim routine? (for optim routine)
inefdec	Production or cost function
upper	Vector of upper values for the optim package.
Method	The method to be used for optim. See 'Details' within optim.
logit	Choice of using logit function
verbose	Logical. Print optimization progress messages? Default is FALSE.
rand.psoptim	Integer. seed for replication of psoptim. Default to NULL.

**Details**

Example based on: A zero inefficiency stochastic frontier model, Journal of Econometrics, S. C. Kumbhakar, C. F. Parmeter and E. G. Tsionas, 2013

**Value**

An object of class "sfareg" containing the following components:

out	A matrix with parameter estimates, standard errors, and t-values.
opt	A list containing the optimization results from the final optimization procedure.
total_time	The total computation time for model estimation.
start_v	The starting values used in the optimization.
model_name	The name of the zero-inflated stochastic frontier model estimated (ZISF or ZISF_Z).
formula	The formula used in the model specification.
jlms	Predicted technical efficiency using the Jondrow et al. (1982) conditional mean estimator (JLMS).
post.prob	Posterior probabilities of being fully efficient.
coefficients	A vector of estimated parameters.
std.errors	A vector of standard errors for the estimated parameters (NA if optHessian = FALSE).
t.values	A vector of t-values for the estimated parameters (NA if optHessian = FALSE).
call	The matched call.

**Note**

Standard errors require optHessian set to TRUE

**Author(s)**

Chris F. Parmeter and David H. Bernstein

**References**

S. C. Kumbhakar, C. F. Parmeter and E. G. Tsionas (2013)

**See Also**

panel89

**Examples**

```
library(sfa)

eqz <- y ~ q1 + q2 + q3 + q4 + q5 + w1 + w2 + w3 + w4 | z

data(panel89)

zsfm(formula = eqz,
      model_name = "ZISF_Z",
      data = panel89,
      logit = TRUE)
```

# Index

## \* datasets

- FinnishElec, 8
- Indian, 9
- panel89, 10
- USUtilities, 19

## \* package

- sfa-package, 2

data\_gen\_cs, 3

data\_gen\_p, 6

FinnishElec, 8

Indian, 9

panel89, 10

pdata.frame, 7, 8

print.sfareg, 12

psfm, 13

rcauchy, 5

rexp, 5

rnorm, 5

rt, 5

rtruncnorm, 5

runif, 5

sfa-package, 2

sfm, 15

summary.sfareg, 18

USUtilities, 19

zsfm, 20