Package ‘npsf’

September 23, 2015

Type Package

Title Nonparametric and Stochastic Efficiency and Productivity Analysis

Version 0.1.6

Date 2015-09-22

Author Oleg Badunenko [aut, cre],
Yaryna Kolomiytseva [aut],
Pavlo Mozharovskyi [aut]

Maintainer Oleg Badunenko <obadunen@uni-koeln.de>

Description Provides a variety of tools for nonparametric and parametric efficiency measurement.

Depends Formula

Suggests snowFT, Rmpi

Encoding UTF-8

License GPL-2 | file LICENSE

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-09-23 02:52:00

R topics documented:

`npsf-package` *Introduction to Nonparametric and Stochastic Frontier Analysis*

Description

This package provides a variety of tools for nonparametric and parametric efficiency measurement.

Details

The nonparametric models in `npsf` comprise nonradial efficiency measurement (`tenonradial`), where non-proportional reductions (expansions) in each positive input (output) are allowed, as well as popular radial efficiency measurement (`teradial`), where movements to the frontier are proportional.

Using bootstrapping techniques, `teradialbc`, `nptestrts`, `nptestind` deal with statistical inference about the radial efficiency measurement. `nptestind` helps in deciding which type of the bootstrap to employ. Global return to scale and individual scale efficiency is tested by `nptestrts`. Finally, `teradialbc` performs bias correction of the radial Debrue-Farrell input- or output-based measure of technical efficiency, computes bias and constructs confidence intervals.

Computer intensive functions `teradialbc` and `nptestrts` allow making use of parallel computing, even on a single machine with multiple cores. Help files contain examples that are intended to introduce the usage.

Solving linear programming problems makes use of quickhull (http://www.qhull.org/, Barber et al. (1996)) algorithm and GLPK Version 4.55 (GNU Linear Programming Kit 2012, available at http://www.gnu.org/software/glpk/) coded in C.

The parametric stochastic frontier models in `npsf` can be estimated by `sf`, which performs maximum likelihood estimation of the frontier parameters and technical or cost efficiencies. Inefficiency error component can be assumed to be have either half-normal or truncated normal distribution. `sf` allows modelling multiplicative heteroskedasticity of either inefficiency or random noise component, or both. Additionally, marginal effects of exogenous variable(s) on the expected value of inefficiency term can be computed.

For details of the respective method please see the reference at the end of this introduction and of the respective help file.

All function in `npsf` accept formula with either names of variables in the data set, or names of the matrices. Except for `nptestind`, all function return `esample`, a logical vector length of which is determined by `data` and `subset` (if specified) or number of rows in matrix `outputs`. `esample` equals `TRUE` if this data point parted in estimation procedure, and `FALSE` otherwise.

Package:	npsf
Type:	Package
Version:	0.1.6
Date:	2015-09-22
License:	GPL-2 | file LICENSE

Author(s)

Oleg Badunenko, <obadunen@uni-koeln.de>

Yaryna Kolomiytseva, <yaryna.kolomiytseva@uni-koeln.de>

Pavlo Mozharovskyi, <mozharovskyi@statistik.uni-koeln.de>

Maintainer: Oleg Badunenko <obadunen@uni-koeln.de>

References

Barber, C. B., D. P. Dobkin, and H. Huhdanpaa. 1996. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software* **22**(4): 469–483.

Bartelsman, E.J. and Gray, W. (1996), The NBER Manufacturing Productivity Database, *National Bureau of Economic Research*, Technical Working Paper Series, http://www.nber.org/papers/t0205.

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

Charnes, A., W. W. Cooper, and E. Rhodes. 1981. Evaluating Program and Managerial Efficiency: An Application of Data Envelopment Analysis to Program Follow Through. *Management Science* **27**: 668–697.

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

Debreu, G. 1951. The Coefficient of Resource Utilization. *Econometrica* **19**: 273–292.

Färe, R. and Lovell, C. A. K. (1994), Measuring the technical efficiency of production, *Journal of Economic Theory*, **19**, 150–162.

Färe, R., Grosskopf, S. and Lovell, C. A. K. (1994), *Production Frontiers*, Cambridge U.K.: Cambridge University Press.

Farrell, M. J. 1957. The Measurement of Productive Efficiency. *Journal of the Royal Statistical Society. Series A (General)* **120**(3): 253–290.

GNU Linear Programming Kit. 2012. (http://www.gnu.org/software/glpk/glpk.html).

Heston, A., and R. Summers. 1991. The Penn World Table (Mark 5): An Expanded Set of International Comparisons, 1950-1988. *The Quarterly Journal of Economics* **106**: 327–368.

Horrace, W. and Schmidt, P. (1996), On ranking and selection from independent truncated normal distributions. *Journal of Productivity Analysis*, **7**, 257–282.

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

Kneip, A., Simar L., and P.W. Wilson (2008), Asymptotics and Consistent Bootstraps for DEA Estimators in Nonparametric Frontier Models, *Econometric Theory*, **24**, 1663–1697

Koetter, M., Kolari, J., and Spierdijk, L. (2012), Enjoying the quiet life under deregulation? Evidence from adjusted Lerner indices for U.S. banks. *Review of Economics and Statistics*, **94**, 2, 462–480.

Kumbhakar, S. and Lovell, C. (2003), *Stochastic Frontier Analysis.* Cambridge: Cambridge University Press.

Restrepo-Tobon, D. and Kumbhakar, S. (2014), Enjoying the quiet life under deregulation? Not Quite. *Journal of Applied Econometrics*, **29**, 2, 333–343.

Simar, L. and P.W. Wilson (1998), Sensitivity Analysis of Efficiency Scores: How to Bootstrap in Nonparametric Frontier Models, *Management Science*, **44**, 49–61

Simar, L. and P.W. Wilson (2000), A General Methodology for Bootstrapping in Nonparametric Frontier Models, *Journal of Applied Statistics*, **27**, 779–802

Simar, L. and P.W. Wilson (2002), Nonparametric Tests of Return to Scale, *European Journal of Operational Research*, **139**, 115–132

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

Wilson P.W. (2003), Testing Independence in Models of Productive Efficiency, *Journal of Productivity Analysis*, **20**, 361–390

`banks05` *U.S. Commercial Banks Data*

Description

`banks05` data frame contains selected variables from the U.S. commercial banks data of Koetter et al. (2012) for year 2005 and 500 banks randomly sampled from around 5000. Dependent variable was randomly generated, as described under 'Details', to satisfy the assumptions of doubly heteroskedastic stochastic cost frontier model. This data is, therefore, not suitable for research purposes.

Usage

```
data(banks05)
```

Format

This data frame contains the following variables (columns):

`lnC` Randomly generated total operating costs.

`lnw1` Logarithm of the cost of fixed assets divided by the cost of borrowed funds.

`lnw2` Logarithm of the cost of labor (in thousands of US dollars) divided by the cost of borrowed funds.

`lny1` Logarithm of total securities (in thousands of US dollars).

`lny2` Logarithm of total loans and leases (in thousands of US dollars).

`ER` Gross total equity to gross total assets ratio.

`LA` Total loans and leases to gross total assets ratio.

Details

The variable representing total operating costs was generated as follows:

$$lnC = \beta0 + \beta1 * lnw1 + \beta2 * lnw2 + \gamma1 * lny1 + \gamma2 * lny2 + \nu + u,$$

where $\nu\tilde{}N(0, exp(\alpha0 + \alpha1 * LA))$ and $u\tilde{}N+(\delta0 + \delta1 * ER, exp(\omega0 + \omega1 * ER))$. More detailed description of input prices, outputs, and exogenous variables is provided in Koetter et al. (2012). See also related study of Restrepo-Tobon and Kumbhakar (2014).

Source

http://qed.econ.queensu.ca/jae/2014-v29.2/restrepo-tobon-kumbhakar/.

References

Koetter, M., Kolari, J., and Spierdijk, L. (2012), Enjoying the quiet life under deregulation? Evidence from adjusted Lerner indices for U.S. banks. *Review of Economics and Statistics*, **94**, 2, 462–480.

Restrepo-Tobon, D. and Kumbhakar, S. (2014), Enjoying the quiet life under deregulation? Not Quite. *Journal of Applied Econometrics*, **29**, 2, 333–343.

---

ccr81 *Program Follow Through at Primary Schools*

Description

The data set is from an US federally sponsored program for providing remedial assistance to disadvantaged primary school students. The data comprises 70 school sites.

Usage

```
data( ccr81 )
```

Format

This data frame contains the following variables (columns):

`nu` School Site Number

`y1` Total Reading Score

`y2` Total Math Score

`y3` Total Coopersmith Score

`x1` Education Level of Mother

`x2` Occupation Index

`x3` Parental Visit Index

`x4` Counseling Index

`x5` Number of Teachers

Details

The data were originally used to evaluate the efficiency of public programs and their management.

Source

A. Charnes, W. W. Cooper and E. Rhodes (1981), Evaluating Program and Managerial Efficiency: An Application of Data Envelopment Analysis to Program Follow Through, *Management Science*, **27**, 668–697.

References

Charnes, A., W. W. Cooper, and E. Rhodes. 1981. Evaluating Program and Managerial Efficiency: An Application of Data Envelopment Analysis to Program Follow Through. *Management Science* **27**: 668–697.

---

nptestind *Nonparametric Test of Independence*

Description

In output based efficiency measurement, routine `nptestind` perform test that radial (Debreu-Farrell) output-based measure of technical efficiency under chosen assumption about the technology and mix of outputs are independent. In input-based efficiency measurement, routine `nptestind` perform test that radial (Debreu-Farrell) input-based measure of technical efficiency under chosen assumption about the technology and mix of inputs are independent. Testing is performed using bootstrap technique.

Usage

```
nptestind(formula, data, subset,
 rts = c("C", "NI", "V"), base = c("output", "input"),
 reps = 999, alpha = 0.05,
 print.level = 1, dots = TRUE)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model. The details of model specification are given under 'Details'.
`data`	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment (`formula`), typically the environment from which `teradial` is called.
`subset`	an optional vector specifying a subset of observations for which technical efficiency is to be computed.
`rts`	character or numeric. string: first letter of the word "c" for constant, "n" for non-increasing, or "v" for variable returns to scale assumption. numeric: 3 for constant, 2 for non-increasing, or 3 for variable returns to scale assumption.

`base` character or numeric. string: first letter of the word "o" for computing output-based or "i" for computing input-based technical efficiency measure. string: 2 for computing output-based or 1 for computing input-based technical efficiency measure

`reps` specifies the number of bootstrap replications to be performed. The default is 999. The minimum is 100. Adequate estimates of confidence intervals using bias-corrected methods typically require 1,000 or more replications.

`alpha` sets significance level; default is `alpha=0.05`.

`dots` logical. Relevant if `print.level>=1`. If TRUE, one dot character is displayed for each successful replication; if FALSE, display of the replication dots is suppressed.

`print.level` numeric. 0 - nothing is printed; 1 - print summary of the model and data. 2 - print summary of technical efficiency measures. 3 - print estimation results observation by observation. Default is 1.

Details

In output based efficiency measurement, routine `nptestind` perform test that radial (Debreu-Farrell) output-based measure of technical efficiency under chosen assumption about the technology and mix of outputs are independent. In input-based efficiency measurement, routine `nptestind` perform test that radial (Debreu-Farrell) input-based measure of technical efficiency under chosen assumption about the technology and mix of inputs are independent.

Testing is performed using bootstrap technique.

Value

`nptestrts` returns a list of class `npsf` containing the following elements:

`K` numeric: number of data points.

`M` numeric: number of inputs.

`N` numeric: number of outputs.

`rts` string: RTS assumption.

`base` string: base for efficiency measurement.

`reps` numeric: number of bootstrap replications.

`alpha` numeric: significance level.

`t4n` numeric: value of the T4n statistic.

`pval` numeric: p-value of the test of independence.

Author(s)

Oleg Badunenko <obadunen@uni-koeln.de>, Pavlo Mozharovskyi <mozharovskyi@statistik.uni-koeln.de>

References

Färe, R. and Lovell, C. A. K. (1994), Measuring the technical efficiency of production, *Journal of Economic Theory*, **19**, 150–162

Färe, R., Grosskopf, S. and Lovell, C. A. K. (1994), *Production Frontiers*, Cambridge U.K.: Cambridge University Press

Wilson P.W. (2003), Testing Independence in Models of Productive Efficiency, *Journal of Productivity Analysis*, **20**, 361–390

See Also

`teradial`, `tenonradial`, `teradialbc`, `nptestrts`, `sf`

Examples

```
## Not run:

require( npsf )

# Prepare data and matrices

data( ccr81 )
head( ccr81 )

# Create some missing values

ccr81 [64, "x4"] <- NA # just to create missing
ccr81 [68, "y2"] <- NA # just to create missing

Y2 <- as.matrix( ccr81[ , c("y1", "y2", "y3"), drop = FALSE] )
X2 <- as.matrix( ccr81[ , c("x1", "x2", "x3", "x4", "x5"), drop = FALSE] )

# Perform nonparametric test that radial (Debreu-Farrell)
# output-based measure of technical efficiency under assumption of
# NIRS technology and mix of outputs are independent. Test is
# performed based on 999 replications at the 5

t1 <- nptestind ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
data = ccr81, base = "o", rts = "n",
reps = 999, dots = TRUE)


# Really large data-set

data(usmanuf)
head(usmanuf)

nrow(usmanuf)
table(usmanuf$year)

# This will take some time depending on computer power
```

```
data(usmanuf)
head(usmanuf)

t2 <- nptestind ( Y ~ K + L + M, data = usmanuf,
subset = year >= 1999 & year <= 2000,
reps = 999, dots = TRUE, base = "i", rts = "v")


## End(Not run)
```

---

nptestrts *Nonparametric Test of Returns to Scale*

Description

Routine `nptestrts` performs nonparametric tests the returns to scale of the underlying technology via bootstrapping techniques.

Usage

```
nptestrts(formula, data, subset,
 base = c("output", "input"),
 homogeneous = TRUE, test.two = TRUE,
 reps = 999, alpha = 0.05,
 core.count = 1, cl.type = c("SOCK", "MPI"),
 print.level = 1, dots = TRUE)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model. The details of model specification are given under 'Details'.
`data`	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment (`formula`), typically the environment from which `teradial` is called.
`subset`	an optional vector specifying a subset of observations for which technical efficiency is to be computed.
`base`	character or numeric. string: first letter of the word "o" for computing output-based or "i" for computing input-based technical efficiency measure. string: 2 for computing output-based or 1 for computing input-based technical efficiency measure
`homogeneous`	logical. If TRUE, the reference set is bootstrapped with homogeneous smoothing; if FALSE, the reference set is bootstrapped with heterogeneous smoothing.

`test.two` logical. If TRUE, test 2, where efficiency measures under assumption of non-increasing and variable returns to scale technology are compared; if FALSE, `nptestrts` stops after test 1 is completed.

`reps` specifies the number of bootstrap replications to be performed. The default is 999. The minimum is 100. Adequate estimates of confidence intervals using bias-corrected methods typically require 1,000 or more replications.

`alpha` sets significance level; default is `alpha=0.05`.

`core.count` positive integer. Number of cluster nodes. If `core.count=1`, the process runs sequentially. See `performParallel` for more details.

`cl.type` Character string that specifies cluster type (see `makeClusterFT`). Possible values are 'MPI' and 'SOCK' ('PVM' is currently not available). See `performParallel` for more details.

`dots` logical. Relevant if `print.level>=1`. If TRUE, one dot character is displayed for each successful replication; if FALSE, display of the replication dots is suppressed.

`print.level` numeric. 0 - nothing is printed; 1 - print summary of the model and data. 2 - print summary of technical efficiency measures. 3 - print estimation results observation by observation. Default is 1.

Details

Routine `nptestrts` `nptestrts` performs nonparametric tests the returns to scale of the underlying technology.

If `test.two` is not specified, `nptestrts` performs only Test #1, which consists of two parts. First, the null hypothesis that the technology is globally CRS (vs VRS) is tested. Second, the null hypothesis that the data point is scale efficient is tested.

If `test.two` is specified, `nptestrts` may perform Test #2. If the null hypothesis that the technology is CRS is rejected, `test.two` requests that `nptestrts` tests the null hypothesis that the technology is NIRS (vs VRS). If not all data points are scale efficient, `nptestrts` tests that the reason for scale inefficiency is DRS. If the null hypothesis that the technology is CRS is not rejected and all data points are scale efficient, `nptestrts` will not perform Test #2 even if `test.two` is specified.

Models for `nptestrts` are specified symbolically. A typical model has the form `outputs ~ inputs`, where `outputs` (`inputs`) is a series of (numeric) terms which specifies outputs (inputs). Refer to the examples.

If `core.count>=1`, `nptestrts` will perform bootstrap on multiple cores. Parallel computing requires package `snowFT`. By the default cluster type is defined by option `cl.type="SOCK"`. Specifying `cl.type="MPI"` requires package `Rmpi`.

On some systems, specifying option `cl.type="SOCK"` results in much quicker execution than specifying option `cl.type="MPI"`. Option `cl.type="SOCK"` might be problematic on Mac system.

Parallel computing make a difference for large data sets. Specifying option `dots=TRUE` will indicate at what speed the bootstrap actually proceeds. Specify `reps=100` and compare two runs with option `core.count=1` and `core.count>1` to see if parallel computing speeds up the bootstrap. For small samples, parallel computing may actually slow down the `nptestrts`.

Value

nptestrts returns a list of class npsf containing the following elements:

K	numeric: number of data points.
M	numeric: number of inputs.
N	numeric: number of outputs.
rts	string: RTS assumption.
base	string: base for efficiency measurement.
reps	numeric: number of bootstrap replications.
alpha	numeric: significance level.
teCrs	numeric: measures of technical efficiency under the assumption of CRS.
teNrs	numeric: measures of technical efficiency under the assumption of NiRS.
teVrs	numeric: measures of technical efficiency under the assumption of VRS.
sefficiency	numeric: scale efficiency.
sefficiencyMean	numeric: ratio of means of technical efficiency measures under CRS and VRS.
pGlobalCRS	numeric: p-value of the test that the technology is globally CRS.
psefficient	numeric: p-value of the test that data point is statistically scale efficient.
sefficient	logical: returns TRUE, if statistically scale efficient; FALSE otherwise.
nsefficient	numeric: number of statistically scale efficient.
nrsOVERvrsMean	numeric: ratio of means of technical efficiency measures under NIRS and VRS (if test.two=TRUE).
pGlobalNRS	numeric: p-value of the test the technology is globally NIRS (if test.two=TRUE).
sineffdrs	logical: returns TRUE if statistically scale inefficient due to DRS and FALSE if statistically scale inefficient due to IRS (if test.two=TRUE and not all data points are statistically scale efficient nsefficient<K).
pineffdrs	numeric: p-value of the test that data point is scale inefficient due to DRS (if test.two=TRUE and not all data points are statistically scale efficient nsefficient<K).
nrsOVERvrs	numeric: ratio of measures of technical efficiency under NiRS and VRS (if test.two=TRUE and not all data points are statistically scale efficient nsefficient<K).
esample	logical: returns TRUE if the observation in user supplied data is in the estimation subsample and FALSE otherwise.

Note

Before specifying option homogeneous it is advised to preform the test of independence (see nptestind).

Author(s)

Oleg Badunenko <obadunen@uni-koeln.de>, Pavlo Mozharovskyi <mozharovskyi@statistik.uni-koeln.de>

References

Färe, R. and Lovell, C. A. K. (1994), Measuring the technical efficiency of production, *Journal of Economic Theory*, **19**, 150–162

Färe, R., Grosskopf, S. and Lovell, C. A. K. (1994), *Production Frontiers*, Cambridge U.K.: Cambridge University Press

Simar L. and P.W. Wilson (2002), Nonparametric Tests of Return to Scale, *European Journal of Operational Research*, **139**, 115–132

See Also

teradial, tenonradial, teradialbc, nptestind, sf

Examples

```
## Not run:

require( npsf )

# Prepare data and matrices

data( ccr81 )
head( ccr81 )

# Create some missing values

ccr81 [64, "x4"] <- NA # just to create missing
ccr81 [68, "y2"] <- NA # just to create missing

Y2 <- as.matrix( ccr81[ , c("y1", "y2", "y3"), drop = FALSE] )
X2 <- as.matrix( ccr81[ , c("x1", "x2", "x3", "x4", "x5"), drop = FALSE] )

# Perform output-based test of returns to scale smoothed
# homogeneous bootstrap with 999 replications at the 5
# significance level.  Also perform Test #2

t1 <- nptestrts(y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
data = ccr81, homogeneous = TRUE,
reps = 999, dots = TRUE, base = "o")

# suppress printing replication dots
t2 <- nptestrts(Y2 ~ X2,
homogeneous = TRUE,
reps = 100, dots = FALSE, base = "o")


# heterogeneous
t3 <- nptestrts(Y2 ~ X2,
homogeneous = FALSE,
reps = 100, dots = TRUE, base = "o")
```

```
# ==========================
# ===  Parallel computing ===
# ==========================

# Perform previous test but use 4 cores and
# cluster type MPI

t3 <-  nptestrts(y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
data = ccr81, homogeneous = FALSE,
reps = 100, dots = TRUE, base = "o",
core.count = 4, cl.type = "MPI")


# Really large data-set

data(usmanuf)
head(usmanuf)

nrow(usmanuf)
table(usmanuf$year)

# Figure industries to include in the sample (first quarter)

summary(usmanuf[usmanuf$year >= 1999 & usmanuf$year < 2000, "naics"])

# This test is quite demanding and it will take some time
# depending on computer power

t4 <- nptestrts(Y ~ K + L + M, data = usmanuf,
subset = year >= 1999 & year < 2000 & naics < 321900,
homogeneous = FALSE, reps = 100, dots = TRUE, base = "o",
core.count = 4, cl.type = "MPI")

# This is very computer intensive task

t5 <- nptestrts(Y ~ K + L + M, data = usmanuf,
subset = year >= 1999 & year < 2000,
homogeneous = FALSE, reps = 100, dots = TRUE, base = "o",
core.count = 4, cl.type = "MPI")


## End(Not run)
```

---

pwt56 *Penn World Tables 5.6 (compiled in 1995)*

Description

The data set is from Penn World Tables (PWT) 5.6. This data set provides only selected variables for years 1965 and 1990.

Usage

```
data( pwt56 )
```

Format

This data frame contains the following variables (columns):

`Nu` Order Number

`Country` Country Name

`year` 1965 or 1990

`Y` Real GDP chain, international prices of 1985

`K` Capital stock, international prices of 1985

`L` Number of workers, in thousands

Details

The Penn World Table was developed by Robert Summers and Alan Heston (and others) to facilitate consistent national accounts comparisons across countries as well as over time. The data can be used to evaluate the efficiency of economies of various countries in years 1965 and 1990.

Source

`http://www.rug.nl/research/ggdc/data/pwt/pwt-5.6`. These data were originally hosted on the website of the Center for International Comparisons at the University of Pennsylvania.

References

Heston, A. and Summers, R. (1991), The Penn World Table (Mark 5): An Expanded Set of International Comparisons, 1950-1988, *The Quarterly Journal of Economics*, **106**, 327–368.

`sf` *Stochastic Frontier Models Using Cross-Sectional Data*

Description

`sf` performs maximum likelihood estimation of the parameters and technical or cost efficiencies in cross-sectional stochastic (production or cost) frontier models with half-normal or truncated normal distributional assumption imposed on inefficiency error component.

Usage

```
sf(formula, uhet = NULL, vhet = NULL,
 tmean = NULL, prod = TRUE, data, subset,
 distribution = c("h", "t"), start.val = NULL,
 alpha = 0.05, marg.eff = FALSE, digits = 4,
 print.level = 2)
```

Arguments

`formula`	an object of class “formula” (or one that can be coerced to that class): a symbolic description of the model. The details of model specification are given under ‘Details’.
`uhet`	one-sided formula; e.g. `uhet ~ z1 + z2`. Specifies exogenous variables entering the expression for variance of inefficiency error component. If `NULL`, inefficiency term is assumed to be homoskedastic, i.e. $\sigma_u^2 = exp(\gamma[0])$.
`vhet`	one-sided formula; e.g. `vhet ~ z1 + z2`. Specifies exogenous variables entering the expression for variance of random noise error component. If `NULL`, random noise component is assumed to be homoskedastic, i.e. $\sigma_v^2 = exp(\gamma[0])$.
`tmean`	one-sided formula; e.g. `tmean ~ z1 + z2`. Specifies whether the mean of pre-truncated normal distribution of inefficiency term is a linear function of exogenous variables. Used only when `distribution = "t"`. If `NULL`, mean is assumed to be constant.
`prod`	logical. If `TRUE`, the estimates of parameters of stochastic production frontier model and of technical efficiencies are returned; if `FALSE`, the estimates of parameters of stochastic cost frontier model and of cost efficiencies are returned.
`data`	an optional data frame containing variables in the model. If not found in data, the variables are taken from environment (`formula`), typically the environment from which `sf` is called.
`subset`	an optional vector specifying a subset of observations for which technical or cost efficiencies are to be computed.
`distribution`	either `"h"` (half-normal) or `"t"` (truncated normal), specifying the distribution of inefficiency term.
`start.val`	numeric. Starting values to be supplied to the optimization routine. If `NULL`, OLS and method of moments estimates are used (see Kumbhakar and Lovell 2000).
`alpha`	numeric. Defines (1-α)100% two-sided prediction interval for technical or cost efficiencies (see Horrace and Schmidt 1996). Default is 0.05.
`marg.eff`	logical. If `TRUE`, unit-specific marginal effects of exogenous variables on the mean of distribution of inefficiency term are returned.
`digits`	numeric. Number of digits to be displayed in estimation results and for efficiency estimates. Default is 4.
`print.level`	numeric. 1 - print estimation results. 2 - print optimization details. 3 - print summary of point estimates of technical or cost efficiencies. 4 - print unit-specific point and interval estimates of technical or cost efficiencies. Default is 2.

Details

Models for `sf` are specified symbolically. A typical model has the form `y ~ x1 + ...`, where `y` represents the logarithm of outputs or total costs and `{x1,...}` is a series of inputs or outputs and input prices (in logs).

Options `uhet` and `vhet` can be used if multiplicative heteroskedasticity of either inefficiency or random noise component (or both) is assumed; i.e. if their variances can be expressed as exponential functions of (e.g. size-related) exogenous variables (including intercept) (see Caudill et al. 1995).

If `marg.eff = TRUE` and `distribution = "h"`, the marginal effect of *k*th exogenous variable on the expected value of inefficiency term of unit *i* is computed as: $\gamma[k]\sigma[i]/\sqrt{2}\pi$, where $\sigma_u[i] = \sqrt{e}xp(z[i]'\gamma)$. If `distribution = "t"`, marginal effects are returned if either `tmean` or `uhet` are not `NULL`. If the same exogenous variables are specified under both options, (non-monotonic) marginal effects are computed as explained in Wang (2002).

Value

`sf` returns a list of class `npsf` containing the following elements:

`coef`	numeric. Named vector of ML parameter estimates.
`vcov`	matrix. Estimated covariance matrix of ML estimator.
`loglik`	numeric. Value of log-likelihood at ML estimates.
`efficiencies`	data frame. Contains point estimates of unit-specific technical or cost efficiencies: exp(-E(u|e)) of Jondrow et al. (1982), E(exp(-u)|e) of Battese and Coelli (1988), and exp(-M(u|e)), where M(u|e) is the mode of conditional distribution of inefficiency term. In addition, estimated lower and upper bounds of (1-α)100% two-sided prediction intervals are returned.
`marg.effects`	data frame. Contains unit-specific marginal effects of exogenous variables on the expected value of inefficiency term.
`sigmas_u`	matrix. Estimated unit-specific variances of inefficiency term. Returned if `uhet` is not `NULL`.
`sigmas_v`	matrix. Estimated unit-specific variances of random noise component. Returned if `vhet` is not `NULL`.
`mu`	matrix. Estimated unit-specific means of pre-truncated normal distribution of inefficiency term. Returned if `tmean` is not `NULL`.
`esample`	logical. Returns TRUE if the observation in user supplied data is in the estimation subsample and FALSE otherwise.

Author(s)

Oleg Badunenko <obadunen@uni-koeln.de>, Yaryna Kolomiytseva <yaryna.kolomiytseva@uni-koeln.de>

References

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

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

Horrace, W. and Schmidt, P. (1996), On ranking and selection from independent truncated normal distributions. *Journal of Productivity Analysis*, **7**, 257–282.

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

Kumbhakar, S. and Lovell, C. (2003), *Stochastic Frontier Analysis.* Cambridge: Cambridge University Press.

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

See Also

teradial, tenonradial, teradialbc, nptestrts, nptestind

Examples

```
require( npsf )

# Load Penn World Tables 5.6 dataset

data( pwt56 )
head( pwt56 )

# Create some missing values

pwt56 [4, "K"] <- NA

# Stochastic production frontier model with
# homoskedastic error components (half-normal)

# Use subset of observations - for year 1965

m1 <- sf(log(Y) ~ log(L) + log(K), data = pwt56,
 subset = year == 1965, distribution = "h")

# Write efficiencies to the data frame using 'esample':

pwt56$BC[ m1$esample ] <- m1$efficiencies$BC
## Not run: View(pwt56)

# Computation using matrices

Y1 <- as.matrix(log(pwt56[pwt56$year == 1965,
c("Y"), drop = FALSE]))
X1 <- as.matrix(log(pwt56[pwt56$year == 1965,
c("K", "L"), drop = FALSE]))

X1 [51, 2] <- NA # create missing
```

```
X1 [49, 1] <- NA # create missing

m2 <- sf(Y1 ~ X1, distribution = "h")

# Load U.S. commercial banks dataset

data(banks05)
head(banks05)

# Doubly heteroskedastic stochastic cost frontier
# model (truncated normal)

# Print summaries of cost efficiencies' estimates

m3 <- sf(lnC ~ lnw1 + lnw2 + lny1 + lny2, uhet = ~ ER,
 vhet = ~ LA, data = banks05, distribution = "t",
 prod = FALSE, print.level = 3)

# Non-monotonic marginal effects of equity ratio on
# the mean of distribution of inefficiency term

m4 <- sf(lnC ~ lnw1 + lnw2 + lny1 + lny2, uhet = ~ ER,
 tmean = ~ ER, data = banks05, distribution = "t",
 prod = FALSE, marg.eff = TRUE)

summary(m4$marg.effects)
```

`tenonradial` *Nonradial Measure of Technical Efficiency, the Russell Measure*

Description

Routine `tenonradial` uses reduced linear programming to compute the nonradial output- or input-based measure of technical efficiency, which is known as the Russell measure. In input-based nonradial efficiency measurement, this measure allows for non-proportional/different reductions in each positive input, and this is what permits it to shrink an input vector all the way back to the efficient subset. In output-based nonradial efficiency measurement, the Russell measure allows for non-proportional/different expansions of each positive output.

Usage

```
tenonradial(formula, data, subset,
 rts = c("C", "NI", "V"),
 base = c("output", "input"),
 ref = NULL, data.ref = NULL, subset.ref = NULL,
 print.level = 1)
```

Arguments

- `formula` — an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model. The details of model specification are given under 'Details'.
- `data` — an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment (`formula`), typically the environment from which `tenonradial` is called.
- `subset` — an optional vector specifying a subset of observations for which technical efficiency is to be computed.
- `rts` — character or numeric. string: first letter of the word "c" for constant, "n" for non-increasing, or "v" for variable returns to scale assumption. numeric: 3 for constant, 2 for non-increasing, or 3 for variable returns to scale assumption.
- `base` — character or numeric. string: first letter of the word "o" for computing output-based or "i" for computing input-based technical efficiency measure. string: 2 for computing output-based or 1 for computing input-based technical efficiency measure
- `ref` — an object of class "formula" (or one that can be coerced to that class): a symbolic description of inputs and outputs that are used to define the technology reference set. The details of technology reference set specification are given under 'Details'. If reference is not provided, the technical efficiency measures for data points are computed relative to technology based on data points themselves.
- `data.ref` — an optional data frame containing the variables in the technology reference set. If not found in `data.ref`, the variables are taken from environment(ref), typically the environment from which `tenonradial` is called.
- `subset.ref` — an optional vector specifying a subset of observations to define the technology reference set.
- `print.level` — numeric. 0 - nothing is printed; 1 - print summary of the model and data. 2 - print summary of technical efficiency measures. 3 - print estimation results observation by observation. Default is 1.

Details

Routine `tenonradial` computes the nonradial output- or input-based measure of technical efficiency under assumption of constant, non-increasing, or variable returns to scale technology.

Models for `tenonradial` are specified symbolically. A typical model has the form `outputs ~ inputs`, where `outputs` (`inputs`) is a series of (numeric) terms which specifies outputs (inputs). The same goes for reference set. Refer to the examples.

Value

`tenonradial` returns a list of class `npsf` containing the following elements:

- `K` — numeric: number of data points.
- `M` — numeric: number of inputs.
- `N` — numeric: number of outputs.

rts	string: RTS assumption.
base	string: base for efficiency measurement.
te	numeric: nonradial measure (Russell) of technical efficiency.
esample	logical: returns TRUE if the observation in user supplied data is in the estimation subsample and FALSE otherwise.
esample.ref	logical: returns TRUE if the observation in the user supplied reference is in the reference subsample and FALSE otherwise.

Note

In case of one input (output), the input (output)-based Russell measure is equal to Debrue-Farrell (teradial) measure of technical efficiency.

Author(s)

Oleg Badunenko <obadunen@uni-koeln.de>, Pavlo Mozharovskyi <mozharovskyi@statistik.uni-koeln.de>

References

Färe, R. and Lovell, C. A. K. (1994), Measuring the technical efficiency of production, *Journal of Economic Theory*, **19**, 150–162

Färe, R., Grosskopf, S. and Lovell, C. A. K. (1994), *Production Frontiers*, Cambridge U.K.: Cambridge University Press

See Also

teradial, teradialbc, nptestrts, nptestind, sf

Examples

```
require( npsf )

# Prepare data and matrices

data( pwt56 )
head( pwt56 )

# Create some missing values

pwt56 [49, "K"] <- NA # create missing

Y1 <- as.matrix ( pwt56[ pwt56$year == 1965, c("Y"), drop = FALSE] )
X1 <- as.matrix ( pwt56[ pwt56$year == 1965, c("K", "L"), drop = FALSE] )

X1 [51, 2] <- NA # create missing
X1 [49, 1] <- NA # create missing

data( ccr81 )
```

```
head( ccr81 )

# Create some missing values

ccr81 [64, "x4"] <- NA # create missing
ccr81 [68, "y2"] <- NA # create missing

Y2 <- as.matrix( ccr81[ , c("y1", "y2", "y3"), drop = FALSE] )
X2 <- as.matrix( ccr81[ , c("x1", "x2", "x3", "x4", "x5"), drop = FALSE] )

# Computing without reference set

# Using formula

# Country is a categorical variable, so nonradial gives error message

# t1 <- tenonradial ( Country ~ K + L, data = pwt56 )

# for computing the efficiencies of countries in 1965
# with technology reference set is defined by observations in 1965
# (that same sample of countries)

t2 <- tenonradial ( Y ~ K + L, data = pwt56, rts = "v",
base = "in", print.level = 2)

# Using a subset

t3 <- tenonradial ( Y ~ K + L, data = pwt56, subset = year == 1965,
rts = "VRS", base = "in", print.level = 3 )

t4 <- tenonradial ( Y ~ K + L, data = pwt56, subset = Nu < 10,
rts = "vrs", base = "I" )

t5 <- tenonradial ( Y ~ L, data = pwt56, subset = Nu < 10, rts = "v" )

# Multiple outputs

t8 <- tenonradial ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5, data = ccr81,
rts = "v", base = "i" )

# Using a subset

t7 <- tenonradial ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5, data = ccr81,
subset = x5 != 22, rts = "n", base = "o" )

# Computation using matrices

t9 <- tenonradial ( Y1 ~ X1, rts = "v", base = "i" )

# Define subsets on a fly

t10 <- tenonradial ( Y1[-1,] ~ X1[-2,1] )
t11 <- tenonradial ( Y1[-3,] ~ X1[-1,], rts = "v", base = "o" )
```

```
# Multiple outputs

t12 <- tenonradial ( Y2 ~ X2 )
t13 <- tenonradial ( Y2[-66,] ~ X2[-1, -c(1,3)] )


# Computing with reference set

# Using formula

# For computing the efficiencies of countries with order number
# less than 10 with technology reference set defined by countries
# with order number larger than 10 and smaller than 11 (in effect
# no reference set, hence warning) type

t14 <- tenonradial ( Y ~ K + L, data = pwt56, subset = Nu < 10,
ref = Y ~ K + L, data.ref = pwt56,
subset.ref = Nu > 10 & Nu < 11 ) # warning

# For computing the efficiencies of countries with order number
# less than 10 with technology reference set defined by countries
# with order number larger than 10 and smaller than 15 type

t15 <- tenonradial ( Y ~ K + L, data = pwt56, subset = Nu < 10, ref = Y ~ K + L,
data.ref = pwt56, subset.ref = Nu > 10 & Nu < 15 )

# For computing the efficiencies of countries in 1965
# with technology reference set is defined by observations in both
# 1965 and 1990 (all) type

t16 <- tenonradial ( Y ~ K + L, data = pwt56, subset = year == 1965,
rts = "v", base = "i",
ref = Y ~ K + L, data.ref = pwt56 )

# For computing the efficiencies of countries in 1990
# with technology reference set is defined by observations in 1965
# type

t17 <- tenonradial ( Y ~ K + L, data = pwt56, subset = year == 1990,
ref = Y ~ K + L, data.ref = pwt56, subset.ref = year == 1965 )

# Using matrices

t18 <- tenonradial ( Y1[-1,] ~ X1[-2,], ref = Y1[-2,] ~ X1[-1,] )

# error: not equal number of observations in outputs and inputs

# t19 <- tenonradial ( Y1[-1,] ~ X1[-(1:2),],
# ref = Y1[-2,] ~ X1[-1,1] )

# Combined formula and matrix
```

```
# error: not equal number of inputs in data and reference set

# t20 <- tenonradial ( Y ~ K + L, data = pwt56, subset = Nu < 10,
# ref = Y1[-2,] ~ X1[-1,1] )

t21 <- tenonradial ( Y ~ K + L, data = pwt56, subset = Nu < 10,
ref = Y1[-2,] ~ X1[-1,] )

## Not run:

# Really large data-set

data(usmanuf)
head(usmanuf)

nrow(usmanuf)
table(usmanuf$year)

# This will take some time depending on computer power

t22 <- tenonradial ( Y ~ K + L + M, data = usmanuf,
subset = year >= 1995 & year <= 2000 )

# Summary

summary ( t22$te )

# Write efficiencies to the data frame:

usmanuf$te_nonrad_crs_out[ t22$esample ] <- t22$te

head(usmanuf, 17)


## End(Not run)
```

`teradial` *Radial Measure of Technical Efficiency, the Debrue-Farrell Measure*

Description

Routine `teradial` computes radial Debrue-Farrell input- or output-based measure of efficiency via reduced linear programing. In input-based radial efficiency measurement, this measure allows for proportional reductions in each positive input, and this is what permits it to shrink an input vector all the way back to the efficient subset. In output-based radial efficiency measurement, the Debrue-Farrell measure allows for proportional expansions of each positive output.

Usage

```
teradial(formula, data, subset,
 rts = c("C", "NI", "V"),
 base = c("output", "input"),
 ref = NULL, data.ref = NULL, subset.ref = NULL,
 print.level = 1)
```

Arguments

- `formula` an object of class “formula” (or one that can be coerced to that class): a symbolic description of the model. The details of model specification are given under ‘Details’.
- `data` an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment (`formula`), typically the environment from which `teradial` is called.
- `subset` an optional vector specifying a subset of observations for which technical efficiency is to be computed.
- `rts` character or numeric. string: first letter of the word “c” for constant, “n” for non-increasing, or “v” for variable returns to scale assumption. numeric: 3 for constant, 2 for non-increasing, or 3 for variable returns to scale assumption.
- `base` character or numeric. string: first letter of the word “o” for computing output-based or “i” for computing input-based technical efficiency measure. string: 2 for computing output-based or 1 for computing input-based technical efficiency measure
- `ref` an object of class “formula” (or one that can be coerced to that class): a symbolic description of inputs and outputs that are used to define the technology reference set. The details of technology reference set specification are given under ‘Details’. If reference is not provided, the technical efficiency measures for data points are computed relative to technology based on data points themselves.
- `data.ref` an optional data frame containing the variables in the technology reference set. If not found in `data.ref`, the variables are taken from environment(ref), typically the environment from which `teradial` is called.
- `subset.ref` an optional vector specifying a subset of observations to define the technology reference set.
- `print.level` numeric. 0 - nothing is printed; 1 - print summary of the model and data. 2 - print summary of technical efficiency measures. 3 - print estimation results observation by observation. Default is 1.

Details

Routine `teradial` computes the radial output- or input-based measure of technical efficiency under assumption of constant, non-increasing, or variable returns to scale technology.

Models for `teradial` are specified symbolically. A typical model has the form `outputs ~ inputs`, where `outputs` (`inputs`) is a series of (numeric) terms which specifies outputs (inputs). The same goes for reference set. Refer to the examples.

Value

`teradial` returns a list of class `npsf` containing the following elements:

`K`	numeric: number of data points.
`M`	numeric: number of inputs.
`N`	numeric: number of outputs.
`rts`	string: RTS assumption.
`base`	string: base for efficiency measurement.
`te`	numeric: radial measure (Russell) of technical efficiency.
`esample`	logical: returns TRUE if the observation in user supplied data is in the estimation subsample and FALSE otherwise.
`esample.ref`	logical: returns TRUE if the observation in the user supplied reference is in the reference subsample and FALSE otherwise.

Note

In case of one input (output), the input (output)-based Debrue-Farrell measure is equal to Russell measure of technical efficiency (see `tenonradial`).

Author(s)

Oleg Badunenko <obadunen@uni-koeln.de>, Pavlo Mozharovskyi <mozharovskyi@statistik.uni-koeln.de>

References

Färe, R. and Lovell, C. A. K. (1994), Measuring the technical efficiency of production, *Journal of Economic Theory*, **19**, 150–162

Färe, R., Grosskopf, S. and Lovell, C. A. K. (1994), *Production Frontiers*, Cambridge U.K.: Cambridge University Press

See Also

`tenonradial`, `teradialbc`, `nptestrts`, `nptestind`, `sf`

Examples

```
require( npsf )

# Prepare data and matrices

data( pwt56 )
head( pwt56 )

# Create some missing values

pwt56 [49, "K"] <- NA # create missing
```

```
Y1 <- as.matrix ( pwt56[ pwt56$year == 1965, c("Y"), drop = FALSE] )
X1 <- as.matrix ( pwt56[ pwt56$year == 1965, c("K", "L"), drop = FALSE] )

X1 [51, 2] <- NA # create missing
X1 [49, 1] <- NA # create missing

data( ccr81 )
head( ccr81 )

# Create some missing values

ccr81 [64, "x4"] <- NA # create missing
ccr81 [68, "y2"] <- NA # create missing

Y2 <- as.matrix( ccr81[ , c("y1", "y2", "y3"), drop = FALSE] )
X2 <- as.matrix( ccr81[ , c("x1", "x2", "x3", "x4", "x5"), drop = FALSE] )

# Computing without reference set

# Using formula

# Country is a categorical variable, so nonradial gives error message

# t1 <- teradial ( Country ~ K + L, data = pwt56 )

# for computing the efficiencies of countries in 1965
# with technology reference set is defined by observations in 1965
# (that same sample of countries)

t2 <- teradial ( Y ~ K + L, data = pwt56, rts = "v",
base = "in", print.level = 2)

# Using a subset

t3 <- teradial ( Y ~ K + L, data = pwt56, subset = year == 1965,
rts = "VRS", base = "in", print.level = 3 )

t4 <- teradial ( Y ~ K + L, data = pwt56, subset = Nu < 10,
rts = "vrs", base = "I" )

t5 <- teradial ( Y ~ L, data = pwt56, subset = Nu < 10, rts = "v" )

# Multiple outputs

t8 <- teradial ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5, data = ccr81,
rts = "v", base = "i" )

# Using a subset

t7 <- teradial ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5, data = ccr81,
subset = x5 != 22, rts = "n", base = "o" )
```

```
# Computation using matrices

t9 <- teradial ( Y1 ~ X1, rts = "v", base = "i" )

# Define subsets on a fly

t10 <- teradial ( Y1[-1,] ~ X1[-2,1] )
t11 <- teradial ( Y1[-3,] ~ X1[-1,], rts = "v", base = "o" )

# Multiple outputs

t12 <- teradial ( Y2 ~ X2 )
t13 <- teradial ( Y2[-66,] ~ X2[-1, -c(1,3)] )


# Computing with reference set

# Using formula

# For computing the efficiencies of countries with order number
# less than 10 with technology reference set defined by countries
# with order number larger than 10 and smaller than 11 (in effect
# no reference set, hence warning) type

t14 <- teradial ( Y ~ K + L, data = pwt56, subset = Nu < 10,
ref = Y ~ K + L, data.ref = pwt56,
subset.ref = Nu > 10 & Nu < 11 ) # warning

# For computing the efficiencies of countries with order number
# less than 10 with technology reference set defined by countries
# with order number larger than 10 and smaller than 15 type

t15 <- teradial ( Y ~ K + L, data = pwt56, subset = Nu < 10, ref = Y ~ K + L,
data.ref = pwt56, subset.ref = Nu > 10 & Nu < 15 )

# For computing the efficiencies of countries in 1965
# with technology reference set is defined by observations in both
# 1965 and 1990 (all) type

t16 <- teradial ( Y ~ K + L, data = pwt56, subset = year == 1965,
rts = "v", base = "i",
ref = Y ~ K + L, data.ref = pwt56 )

# For computing the efficiencies of countries in 1990
# with technology reference set is defined by observations in 1965
# type

t17 <- teradial ( Y ~ K + L, data = pwt56, subset = year == 1990,
ref = Y ~ K + L, data.ref = pwt56, subset.ref = year == 1965 )

# Using matrices

t18 <- teradial ( Y1[-1,] ~ X1[-2,], ref = Y1[-2,] ~ X1[-1,] )
```

```
# error: not equal number of observations in outputs and inputs

# t19 <- teradial ( Y1[-1,] ~ X1[-(1:2),],
# ref = Y1[-2,] ~ X1[-1,1] )

# Combined formula and matrix

# error: not equal number of inputs in data and reference set

# t20 <- teradial ( Y ~ K + L, data = pwt56, subset = Nu < 10,
# ref = Y1[-2,] ~ X1[-1,1] )

t21 <- teradial ( Y ~ K + L, data = pwt56, subset = Nu < 10,
ref = Y1[-2,] ~ X1[-1,] )

# Really large data-set

data(usmanuf)
head(usmanuf)

nrow(usmanuf)
table(usmanuf$year)

# This will take some time depending on computer power

t22 <- teradial ( Y ~ K + L + M, data = usmanuf,
subset = year >= 1995 & year <= 2000 )

# Summary

summary ( t22$te )

# Write efficiencies to the data frame:

usmanuf$te_nonrad_crs_out[ t22$esample ] <- t22$te

head(usmanuf, 17)
```

teradialbc — *Statistical Inference Regarding the Radial Measure of Technical Efficiency*

Description

Routine `teradialbc` performs bias correction of the radial Debrue-Farrell input- or output-based measure of technical efficiency, computes bias and constructs confidence intervals via bootstrapping techniques.

Usage

```
teradialbc(formula, data, subset,
 ref = NULL, data.ref = NULL, subset.ref = NULL,
 rts = c("C", "NI", "V"), base = c("output", "input"),
 homogeneous = TRUE, smoothed = TRUE, kappa = NULL,
 reps = 999, level = 95,
 core.count = 1, cl.type = c("SOCK", "MPI"),
 print.level = 1, dots = TRUE)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model. The details of model specification are given under 'Details'.
`data`	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment (`formula`), typically the environment from which `teradial` is called.
`subset`	an optional vector specifying a subset of observations for which technical efficiency is to be computed.
`rts`	character or numeric. string: first letter of the word "c" for constant, "n" for non-increasing, or "v" for variable returns to scale assumption. numeric: 3 for constant, 2 for non-increasing, or 3 for variable returns to scale assumption.
`base`	character or numeric. string: first letter of the word "o" for computing output-based or "i" for computing input-based technical efficiency measure. string: 2 for computing output-based or 1 for computing input-based technical efficiency measure
`ref`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of inputs and outputs that are used to define the technology reference set. The details of technology reference set specification are given under 'Details'. If reference is not provided, the technical efficiency measures for data points are computed relative to technology based on data points themselves.
`data.ref`	an optional data frame containing the variables in the technology reference set. If not found in `data.ref`, the variables are taken from environment(ref), typically the environment from which `teradial` is called.
`subset.ref`	an optional vector specifying a subset of observations to define the technology reference set.
`smoothed`	logical. If TRUE, the reference set is bootstrapped with smoothing; if FALSE, the reference set is bootstrapped with subsampling.
`homogeneous`	logical. Relevant if `smoothed=TRUE`. If TRUE, the reference set is bootstrapped with homogeneous smoothing; if FALSE, the reference set is bootstrapped with heterogeneous smoothing.
`kappa`	relevant if `smoothed=TRUE`. 'kappa' sets the size of the subsample as K^kappa, where K is the number of data points in the original reference set. The default value is 0.7. 'kappa' may be between 0.5 and 1.

reps	specifies the number of bootstrap replications to be performed. The default is 999. The minimum is 100. Adequate estimates of confidence intervals using bias-corrected methods typically require 1,000 or more replications.
level	sets confidence level for confidence intervals; default is level=95.
core.count	positive integer. Number of cluster nodes. If core.count=1, the process runs sequentially. See performParallel for more details.
cl.type	Character string that specifies cluster type (see makeClusterFT). Possible values are 'MPI' and 'SOCK' ('PVM' is currently not available). See performParallel for more details.
dots	logical. Relevant if print.level>=1. If TRUE, one dot character is displayed for each successful replication; if FALSE, display of the replication dots is suppressed.
print.level	numeric. 0 - nothing is printed; 1 - print summary of the model and data. 2 - print summary of technical efficiency measures. 3 - print estimation results observation by observation. Default is 1.

Details

Routine teradialbc performs bias correction of the radial Debrue-Farrell input- or output-based measure of technical efficiency, computes bias and constructs confidence intervals via bootstrapping techniques.

Models for teradialbc are specified symbolically. A typical model has the form outputs ~ inputs, where outputs (inputs) is a series of (numeric) terms which specifies outputs (inputs). The same goes for reference set. Refer to the examples.

If core.count>=1, teradialbc will perform bootstrap on multiple cores. Parallel computing requires package snowFT. By the default cluster type is defined by option cl.type="SOCK". Specifying cl.type="MPI" requires package Rmpi.

On some systems, specifying option cl.type="SOCK" results in much quicker execution than specifying option cl.type="MPI". Option cl.type="SOCK" might be problematic on Mac system.

Parallel computing make a difference for large data sets. Specifying option dots=TRUE will indicate at what speed the bootstrap actually proceeds. Specify reps=100 and compare two runs with option core.count=1 and core.count>1 to see if parallel computing speeds up the bootstrap. For small samples, parallel computing may actually slow down the teradialbc.

Value

teradialbc returns a list of class npsf containing the following elements:

K	numeric: number of data points.
M	numeric: number of inputs.
N	numeric: number of outputs.
rts	string: RTS assumption.
base	string: base for efficiency measurement.
reps	numeric: number of bootstrap replications.
level	numeric: confidence level for confidence intervals.

te	numeric: radial measure (Russell) of technical efficiency.
tebc	numeric: bias-corrected radial measures of technical efficiency.
biasboot	numeric: bootstrap bias estimate for the original radial measures of technical efficiency.
varboot	numeric: bootstrap variance estimate for the radial measures of technical efficiency.
biassqvar	numeric: one-third of the ratio of bias squared to variance for radial measures of technical efficiency.
realreps	numeric: actual number of replications used for statistical inference.
telow	numeric: lower bound estimate for radial measures of technical efficiency.
teupp	numeric: upper bound estimate for radial measures of technical efficiency.
teboot	numeric: `reps x K` matrix containing bootstrapped measures of technical efficiency from each of `reps` bootstrap replications.
esample	logical: returns TRUE if the observation in user supplied data is in the estimation subsample and FALSE otherwise.

Note

Before specifying option `homogeneous` it is advised to preform the test of independence (see `nptestind`). Routine `nptestrts` may help deciding regarding option `rts`.

Author(s)

Oleg Badunenko <obadunen@uni-koeln.de>, Pavlo Mozharovskyi <mozharovskyi@statistik.uni-koeln.de>

References

Färe, R. and Lovell, C. A. K. (1994), Measuring the technical efficiency of production, *Journal of Economic Theory*, **19**, 150–162

Färe, R., Grosskopf, S. and Lovell, C. A. K. (1994), *Production Frontiers*, Cambridge U.K.: Cambridge University Press

Kneip, A., Simar L., and P.W. Wilson (2008), Asymptotics and Consistent Bootstraps for DEA Estimators in Nonparametric Frontier Models, *Econometric Theory*, **24**, 1663–1697

Simar, L. and P.W. Wilson (1998), Sensitivity Analysis of Efficiency Scores: How to Bootstrap in Nonparametric Frontier Models, *Management Science*, **44**, 49–61

Simar, L. and P.W. Wilson (2000), A General Methodology for Bootstrapping in Nonparametric Frontier Models, *Journal of Applied Statistics*, **27**, 779–802

See Also

`teradial`, `tenonradial`, `nptestrts`, `nptestind`, `sf`

Examples

```
## Not run:

require( npsf )

# Prepare data and matrices

data( pwt56 )
head( pwt56 )

# Create some missing values

pwt56 [49, "K"] <- NA # just to create missing

Y1 <- as.matrix ( pwt56[ pwt56$year == 1965, c("Y"), drop = FALSE] )
X1 <- as.matrix ( pwt56[ pwt56$year == 1965, c("K", "L"), drop = FALSE] )

X1 [51, 2] <- NA # just to create missing
X1 [49, 1] <- NA # just to create missing

data( ccr81 )
head( ccr81 )

# Create some missing values

ccr81 [64, "x4"] <- NA # just to create missing
ccr81 [68, "y2"] <- NA # just to create missing

Y2 <- as.matrix( ccr81[ , c("y1", "y2", "y3"), drop = FALSE] )
X2 <- as.matrix( ccr81[ , c("x1", "x2", "x3", "x4", "x5"), drop = FALSE] )

# Compute output-based measures of technical efficiency under
# the assumption of CRS (the default) and perform bias-correctiion
# using smoothed homogeneous bootstrap (the default) with 999
# replications (the default).

t1 <- teradialbc ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
data = ccr81)

# or just

t2 <- teradialbc ( Y2 ~ X2)

# Combined formula and matrix

t3 <- teradialbc ( Y ~ K + L, data = pwt56, subset = Nu < 10,
ref = Y1[-2,] ~ X1[-1,] )

# Compute input-based measures of technical efficiency under
# the assumption of VRS and perform bias-correctiion using
# subsampling heterogenous bootstrap with 1999 replications.
```

```
# Choose to report 99
# formed by data points where x5 is not equal 10.
# Suppress printing dots.

t4 <- teradialbc ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
data = ccr81, ref = y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
subset.ref = x5 != 10, data.ref = ccr81, reps = 1999,
smoothed = FALSE, kappa = 0.7, dots = FALSE,
base = "i", rts = "v", level = 99)

# Compute input-based measures of technical efficiency under
# the assumption of NRS and perform bias-correctiion using
# smoothed heterogenous bootstrap with 499 replications for
# all data points. The reference set formed by data points
# where x5 is not equal 10.

t5 <- teradialbc ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
data = ccr81, ref = y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
subset.ref = x5 != 10, data.ref = ccr81, homogeneous = FALSE,
reps = 999, smoothed = TRUE, dots = TRUE, base = "i", rts = "n")


# ==========================
# ===  Parallel computing ===
# ==========================

# Perform previous bias-correction but use 4 cores and
# cluster type MPI

t51 <-  teradialbc ( y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
data = ccr81, ref = y1 + y2 + y3 ~ x1 + x2 + x3 + x4 + x5,
subset.ref = x5 != 10, data.ref = ccr81, homogeneous = FALSE,
reps = 999, smoothed = TRUE, dots = TRUE, base = "i", rts = "n",
core.count = 4, cl.type = "MPI")


# Really large data-set

data(usmanuf)
head(usmanuf)

nrow(usmanuf)
table(usmanuf$year)

# This will take some time depending on computer power

data(usmanuf)
head(usmanuf)

t6 <- teradialbc ( Y ~ K + L + M, data = usmanuf,
subset = year >= 1999 & year <= 2000, homogeneous = FALSE,
base = "o", reps = 100,
core.count = 4, cl.type = "MPI")
```

```
## End(Not run)
```

---

usmanuf *US Manufacturing Industry Data*

Description

This data come from the National Bureau of Economic Research Center for Economic Studies manufacturing industry database. This data set provides only selected variables.

Usage

```
data( usmanuf )
```

Format

This data frame contains the following variables (columns):

`naics` NAICS 6-digit Codes

`year` Year ranges from 1990 to 2009

`Y` Total value of shipments in $1m divided by the deflator for VSHIP 1997=1.000 (vship/piship)

`K` Total real capital stock in $1m (cap)

`L` Total employment in thousands (emp)

`M` Total cost of materials in $1m divided by the deflator for MATCOST 1997=1.000 plus oost of electric & fuels in $1m divided by the deflator for ENERGY 1997=1.000 (matcost/pimat + energy/pien)

Details

This data come from the National Bureau of Economic Research Center for Economic Studies manufacturing industry database, which was compiled by Randy A. Becker and Wayne B. Gray. This database is a joint effort between the National Bureau of Economic Research (NBER) and U.S. Census Bureau's Center for Economic Studies (CES), containing annual industry-level data from 1958-2009 on output, employment, payroll and other input costs, investment, capital stocks, TFP, and various industry-specific price indexes. Because of the change from SIC to NAICS industry definitions in 1997, the database is provided in two versions: one with 459 four-digit 1987 SIC industries and the other with 473 six-digit 1997 NAICS industries.

Source

http://www.nber.org/nberces/.

References

Bartelsman, E.J. and Gray, W. (1996), The NBER Manufacturing Productivity Database, *National Bureau of Economic Research*, Technical Working Paper Series, http://www.nber.org/papers/t0205.

Index