



HAL
open science

**productivity: Indices of Productivity and Profitability
Using Data Envelopment Analysis (DEA). R package
version 1.0.0**

K Hervé Dakpo, Yann Desjeux, Laure Latruffe

► **To cite this version:**

K Hervé Dakpo, Yann Desjeux, Laure Latruffe. productivity: Indices of Productivity and Profitability Using Data Envelopment Analysis (DEA). R package version 1.0.0. 2017. hal-01607286

HAL Id: hal-01607286

<https://hal.science/hal-01607286>

Submitted on 5 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

Package ‘productivity’

July 17, 2017

Version 1.0.0

Date 2017-07-17

Title Indices of Productivity Using Data Envelopment Analysis (DEA)

Maintainer Yann Desjeux <yann.desjeux@inra.fr>

Description

Productivity and profitability, in levels and changes, are measured with various indices. In addition to the classic Malmquist productivity index, the 'productivity' package contains also the multiplicatively complete Fisher, Laspeyres, Paasche, Färe-Primont and Lowe indices. Färe-Primont and Lowe indices verify the transitivity property and can therefore be used for multilateral or multitemporal comparison. Malmquist, Fisher, Laspeyres and Paasche indices not transitive and are only to be used for binary comparison. All indices can also be decomposed into different components providing insightful information on the sources of productivity and profitability improvements. In the use of Malmquist productivity index, the technological change index is further decomposed into bias technological change components. In the case of the Fisher, Laspeyres, Paasche and the transitive Färe-Primont and Lowe indices, it is furthermore possible to rule out technological change. The package also allows to prohibit negative technological change. All the estimations are based on the nonparametric Data Envelopment Analysis (DEA) and several assumptions regarding returns to scale are available (i.e. CRS, VRS, NIRS, NDRS). The package allows parallel computing as an option, depending on the user's computer configuration.

Depends R (>= 2.16)

BugReports https://r-forge.r-project.org/tracker/?group_id=2245

Imports doParallel, foreach, methods, Rglpk

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation no

Author K Hervé Dakpo [aut],
Yann Desjeux [aut, cre],
Laure Latruffe [aut]

R topics documented:

fareprim	2
fisher	6

Indices	9
laspeyres	10
Levels	13
lowe	14
malm	18
paasche	21
Shadowprices	24
usagri	25
Index	27

fareprim	<i>Fare-Primont productivity and profitability indices and components</i>
----------	---

Description

Using Data Envelopment Analysis (DEA), this function measures Fare-Primont productivity and profitability indices in levels and changes, as well as their components.

Profitability measures are only provided when price information is specified.

Shadow prices of inputs and outputs are also computed.

Usage

```
fareprim(data, id.var, time.var, x.vars, y.vars, w.vars = NULL, p.vars = NULL,
  tech.change = TRUE, tech.reg = TRUE, rts = c("vrs", "crs", "nirs", "ndrs"),
  orientation = c("out", "in", "in-out"), parallel = FALSE, cores = max(1,
  detectCores() - 1), scaled = FALSE, by.id = NULL, by.year = NULL)
```

```
## S3 method for class 'FarePrimont'
print(object, digits = NULL, ...)
```

Arguments

<code>data</code>	A dataframe containing the required information for productivity and profitability assessment.
<code>id.var</code>	ID variable of the evaluated firms. Can be an integer or a text string.
<code>time.var</code>	Time period variable. Can be an integer or a text string.
<code>x.vars</code>	Input quantity variables. Can be a vector of text strings or integers.
<code>y.vars</code>	Output quantity variables. Can be a vector of text strings or integers.
<code>w.vars</code>	Input price variables (Optional). Can be a vector of text strings or integers. Set to NULL by default, so only productivity is computed.
<code>p.vars</code>	Output price variables (Optional). Can be a vector of text strings or integers. Set to NULL by default, so only productivity is computed.
<code>tech.change</code>	Logical. If TRUE (default), the model allows for technological change. See also the Details section.
<code>tech.reg</code>	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). See also the Details section.

rts	Character string specifying the returns to scale assumption to be considered. The default value is "vrs" (variable returns to scale). Other possible options are "crs" (constant returns to scale), "nirs" (non-increasing returns to scale), or "ndrs" (non-decreasing returns to scale).
orientation	Character string specifying the orientation to be considered. The default value is "out" (output orientation). Other possible options are "in" (input orientation), and "in-out" (both input and output orientations). For this latter, the geometric mean of input and output orientations is returned.
parallel	Logical. Allows parallel computation. If FALSE (default) the estimation is conducted in sequential mode. If TRUE parallel mode is activated using the number of cores specified in cores.
cores	Integer. Used only if parallel = TRUE. It specifies the number of cores to be used for parallel computation. By default, cores = max(1, detectCores() - 1).
scaled	Logical. Default is FALSE. When set to TRUE, the input and output quantities are rescaled, so that they all have unit means. See also the Details section.
by.id	Integer specifying the reference observation used for computing the indices (Optional). by.id must range between one and the total number of firms per period. See also the Details section.
by.year	Integer specifying the reference year used for computing the indices (Optional). by.year must range between one and the total number of time periods. See also the Details section.
object	An object of class 'FarePrimont'.
digits	The minimum number of significant digits to be printed in values. Default = max(3, getOption("digits") - 3).
...	Currently not used.

Details

fareprim() allows for parallel computations (when parallel = TRUE, possibly by registering a parallel backend (**doParallel** and **foreach** packages)). The cores argument can be used to specify the number of cores to use. However, when the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).

All DEA linear programs are implemented using the package **Rglpk**.

The tech.change option can totally prohibit any technological change. When tech.change is set to FALSE, this cancels the effect of tech.reg whatever the value of this latter. The tech.reg option, when set to FALSE, rules out negative technological change (i.e. technological regress). In this case, technological change will increment between consecutive periods.

The scaled option is useful when working with very large and/or very small values. By default scaled = FALSE. In such case, fareprim() may issue a warning when very large (or very small) values are present in the input and output quantity variables. Note that the Fare-Primont index may be sensitive to the rescaling especially the mix efficiency component.

By default by.id = NULL and by.year = NULL. Hence, with regards with change indices each observation is by default compared to itself in the first period. by.id and by.year allow the user to specify a given reference (e.g. a particular observation in a particular period). When the user specifies an observation in by.id and when by.year = NULL, then the reference observation is by.id in the first period. If by.year is specified and by.id = NULL, then each observation is compared to itself in the specified period of time.

Value

fareprim() returns a list of class 'FarePrimont' for which a summary of productivity and profitability (if price information is specified) indices in levels and changes as well as input (x.vars) and output (y.vars) shadow prices is printed.

This list contains the following items:

Levels	<p>Several elements are provided, depending on the chosen orientation set through orientation argument:</p> <p>REV: Revenues (if w.vars and p.vars are specified)</p> <p>COST: Costs (if w.vars and p.vars are specified)</p> <p>PROF: Profitability ratio (if w.vars and p.vars are specified)</p> <p>P: Aggregated output prices (if w.vars and p.vars are specified)</p> <p>W: Aggregated input prices (if w.vars and p.vars are specified)</p> <p>TT: Terms of trade (i.e. P/W) (if w.vars and p.vars are specified)</p> <p>AO: Aggregated outputs</p> <p>AI: Aggregated inputs</p> <p>TFP: Total Factor Productivity (TFP)</p> <p>MP: Maximum productivity</p> <p>TFPE: TFP efficiency score</p> <p>OTE: Output-oriented technical efficiency score</p> <p>OSE: Output-oriented scale efficiency score</p> <p>OME: Output-oriented mix efficiency score</p> <p>ROSE: Residual output-oriented scale efficiency score</p> <p>OSME: Output-oriented scale-mix efficiency score</p> <p>ITE: Input-oriented technical efficiency score</p> <p>ISE: Input-oriented scale efficiency score</p> <p>IME: Input-oriented mix efficiency score</p> <p>RISE: Residual input-oriented scale efficiency score</p> <p>ISME: Input-oriented scale-mix efficiency score</p> <p>OTE.ITE: Geometric mean of OTE and ITE</p> <p>OSE.ISE: Geometric mean of OSE and ISE</p> <p>OME.IME: Geometric mean of OME and IME</p> <p>ROSE.RISE: Geometric mean of ROSE and RISE</p> <p>OSME.ISME: Geometric mean of OSME and ISME</p> <p>RME: Residual mix efficiency score</p>
Indices	<p>Change indices of the different elements of Levels are provided. Each change index is prefixed by "d" (e.g. profitability change is named dPROF, output-oriented efficiency change is named dOTE, etc.)</p>
Shadowp	<p>The deflated cost input shadow prices and the deflated revenue output shadow prices of the representative observation used to compute the Färe-Primont index. Those prices are derived from dual output and input oriented DEA models.</p>

From an object of class 'FarePrimont' obtained from fareprim(), the

- [Levels](#) function extract productivity and profitability **levels**;
- [Indices](#) function extract productivity and profitability **change indices**; and
- [Shadowprices](#) function extract input and output **Shadow prices**.

Warning

The fareprim() function might not properly work with unbalanced panel data, so the user should make sure to use a balanced panel.

Note

All output efficiency scores are estimated *a la* Shephard, while all input efficiency scores are estimated *a la* Farrell. Hence, all efficiency scores are greater than zero and are lower or equal to one.

Author(s)

K Hervé Dakpo, Yann Desjeux, Laure Latruffe

References

O'Donnell C.J. (2008), An aggregate quantity-price framework for measuring and decomposing productivity and profitability change. School of Economics, University of Queensland, Australia.

O'Donnell C.J. (2011), The sources of productivity change in the manufacturing sectors of the U.S. economy. School of Economics, University of Queensland, Australia. URL <http://www.uq.edu.au/economics/cepa/docs/WP/WP072011.pdf>

O'Donnell C.J. (2012), Nonparametric estimates of the components of productivity and profitability change in U.S. agriculture. *American Journal of Agricultural Economics*, **94**(4), 873–890.

See Also

See [Levels\(\)](#) to retrieve a data frame with Färe-Primont productivity/profitability scores and components.

See [Indices\(\)](#) to retrieve a data frame with Färe-Primont change indices and components.

See [Shadowprices\(\)](#) to retrieve input and output shadow prices.

See also [lowe\(\)](#)

Examples

```
## Not run:
## Färe-Primont productivity, without price information
FareP1 <- fareprim(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), rts = "crs", orientation = "in", scaled = TRUE, by.id = 1, by.year = 1)
FareP1

## Färe-Primont productivity and profitability, with price information
FareP2 <- fareprim(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), w.vars = c(14:17), p.vars = c(11:13), scaled = TRUE, by.id = 1, by.year = 1)
FareP2

## End(Not run)
```

fisher

*Fisher productivity and profitability indices and components***Description**

Using Data Envelopment Analysis (DEA), this function measures Fisher productivity and profitability indices in levels and changes, as well as their components.

The Fisher productivity index is the geometric mean of Laspeyres and Paasche Indices.

Shadow prices of inputs and outputs are also computed.

Usage

```
fisher(data, id.var, time.var, x.vars, y.vars, w.vars, p.vars, tech.change = TRUE,
       tech.reg = TRUE, rts = c("vrs", "crs", "nirs", "ndrs"), orientation = c("out",
       "in", "in-out"), parallel = FALSE, cores = max(1, detectCores() - 1), scaled = FALSE)
```

```
## S3 method for class 'Fisher'
print(object, digits = NULL, ...)
```

Arguments

data	A dataframe containing the required information for productivity and profitability assessment.
id.var	ID variable of the evaluated firms. Can be an integer or a text string.
time.var	Time period variable. Can be an integer or a text string.
x.vars	Input quantity variables. Can be a vector of text strings or integers.
y.vars	Output quantity variables. Can be a vector of text strings or integers.
w.vars	Input price variables. Can be a vector of text strings or integers.
p.vars	Output price variables. Can be a vector of text strings or integers.
tech.change	Logical. If TRUE (default), the model allows for technological change. See also the Details section.
tech.reg	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). See also the Details section.
rts	Character string specifying the returns to scale assumption to be considered. The default value is "vrs" (variable returns to scale). Other possible options are "crs" (constant returns to scale), "nirs" (non-increasing returns to scale), or "ndrs" (non-decreasing returns to scale).
orientation	Character string specifying the orientation to be considered. The default value is "out" (output orientation). Other possible options are "in" (input orientation), and "in-out" (both input and output orientations). For this latter, the geometric mean of input and output orientations is returned.
parallel	Logical. Allows parallel computation. If FALSE (default) the estimation is conducted in sequential mode. If TRUE parallel mode is activated using the number of cores specified in cores.
cores	Integer. Used only if parallel = TRUE. It specifies the number of cores to be used for parallel computation. By default, cores = max(1, detectCores() - 1).

scaled	Logical. Default is FALSE. When set to TRUE, the input and output quantities are rescaled, so that they all have unit means. See also the Details section.
object	An object of class 'Fisher'.
digits	The minimum number of significant digits to be printed in values. Default = $\max(3, \text{getOption}(\text{"digits"}) - 3)$.
...	Currently not used.

Details

fisher() allows for parallel computations (when parallel = TRUE, possibly by registering a parallel backend (**doParallel** and **foreach** packages). The cores argument can be used to specify the number of cores to use. However, when the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).

All DEA linear programs are implemented using the package **Rglpk**.

The tech.change option can totally prohibit any technological change. When tech.change is set to FALSE, this cancels the effect of tech.reg whatever the value of this latter. The tech.reg option, when set to FALSE, rules out negative technological change (i.e. technological regress).

The scaled option is useful when working with very large and/or very small values. By default scaled = FALSE. In such case, fisher() may issue a warning when very large (or very small) values are present in the input and output quantity variables. Note that the Fisher index may be sensitive to the rescaling especially the mix efficiency component.

The Fisher index is not transitive and therefore each firm is compared to itself in the previous period. For the first period, since there is not a previous period, the indices results for this specific period are replaced by NA.

Value

fisher() returns a list of class 'Fisher' for which a summary of productivity and profitability indices in levels and changes as well as a summary of input (x.vars) and output (y.vars) shadow prices is printed.

This list contains the following items:

Levels Several elements are provided, depending on the chosen orientation set through orientation argument:

REV	Revenues
COST	Costs
PROF	Profitability ratio
P	Aggregated output prices
W	Aggregated input prices
TT	Terms of trade (i.e. P/W)
AO	Aggregated outputs
AI	Aggregated inputs
TFP	Total Factor Productivity
MP	Maximum productivity
TFPE	TFP efficiency score
OTE	Output-oriented technical efficiency score
OSE	Output-oriented scale efficiency score
RAE	Revenue allocative efficiency (equivalent to output-oriented mix efficiency score)
ROSE	Residual output-oriented scale efficiency score
OSME	Output-oriented scale-mix efficiency score

ITE	Input-oriented technical efficiency score
ISE	Input-oriented scale efficiency score
CAE	Cost allocative efficiency (equivalent to input-oriented mix efficiency score)
RISE	Residual input-oriented scale efficiency score
ISME	Input-oriented scale-mix efficiency score
OTE . ITE	Geometric mean of OTE and ITE
OSE . ISE	Geometric mean of OSE and ISE
RAE . CAE	Geometric mean of RAE and CAE
ROSE . RISE	Geometric mean of ROSE and RISE
OSME . ISME	Geometric mean of OSME and ISME
RME	Residual mix efficiency score
RE	Revenue efficiency (when orientation = "out")
CE	Cost efficiency (when orientation = "in")
RE . CE	Geometric mean of RE and CE (when orientation = "in-out")
Indices	Change indices of the different elements of Levels are provided. Each change index is prefixed by "d" (e.g. profitability change is named dPROF, output-oriented efficiency change is named dOTE, etc.). Each firm is compared to itself in the previous period. For the first period since there is not a previous period the indices results are replaced by NA for this specific period.
Shadowp	The deflated cost input shadow prices and the deflated revenue output shadow prices. Those prices are derived from dual output and input oriented DEA models for each observation in the sample.

From an object of class 'Fisher' obtained from `fisher()`, the

- `Levels` function extract productivity and profitability **levels**;
- `Indices` function extract productivity and profitability **change indices**; and
- `Shadowprices` function extract input and output **Shadow prices**.

Warning

The `fisher()` function will not work with unbalanced panel data, so the user should make sure to use a balanced panel.

Note

All output efficiency scores are estimated *a la* Shephard, while all input efficiency scores are estimated *a la* Farrell. Hence, all efficiency scores are greater than zero and are lower or equal to one.

Author(s)

K Hervé Dakpo, Yann Desjeux, Laure Latruffe

References

- Diewert W.E. (1992), "Fisher ideal output, input, and productivity indexes revisited". *Journal of Productivity Analysis*, **3**(3), 211-248.
- Coelli T.J., D.S.P. Rao, C.J. O'Donnell, and G.E. Battese (2005), *An Introduction to Efficiency and Productivity Analysis*. Springer Eds.

O'Donnell C.J. (2011), The sources of productivity change in the manufacturing sectors of the U.S. economy. School of Economics, University of Queensland, Australia. URL <http://www.uq.edu.au/economics/cepa/docs/WP/WP072011.pdf>

See Also

See [Levels\(\)](#) to retrieve a data frame with Fisher productivity/profitability scores and components.
 See [Indices\(\)](#) to retrieve a data frame with Fisher change indices and components.
 See [Shadowprices\(\)](#) to retrieve input and output shadow prices.
 See also [laspeyres\(\)](#) and [paasche\(\)](#)

Examples

```
## Fisher profitability and productivity levels and indices' computations
## Not run:
Fisher.prod <- fisher(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), w.vars = c(14:17), p.vars = c(11:13), orientation = "out", scaled = TRUE)
  Fisher.prod

## End(Not run)
```

Indices

Productivity and profitability change indices and components

Description

This function extract productivity and profitability change indices from any object created by either [fareprim\(\)](#), [fisher\(\)](#), [laspeyres\(\)](#), [lowe\(\)](#), [malm\(\)](#), or [paasche\(\)](#) function.

Usage

```
Indices(object, ...)
```

Arguments

object	Object of class 'FarePrimont', 'Fisher', 'Laspeyres', 'Lowe', 'Malmquist', or 'Paasche'.
...	Currently not used.

Details

- An object of class 'FarePrimont' is usually a result of a call to [fareprim](#)
- An object of class 'Fisher' is usually a result of a call to [fisher](#)
- An object of class 'Laspeyres' is usually a result of a call to [laspeyres](#)
- An object of class 'Lowe' is usually a result of a call to [lowe](#)
- An object of class 'Malmquist' is usually a result of a call to [malm](#)
- An object of class 'Paasche' is usually a result of a call to [paasche](#)

Value

[Indices\(\)](#) returns a data frame containing all the elements and observations included in the "Indices" value of the related call.

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

See Also

For details and information on returned values of `Indices()`, see `fareprim()`, `fisher()`, `laspeyres()`, `lowe()`, `malm()`, or `paasche()`

Examples

```
## Not run:
PAASCHE <- paasche(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), w.vars = c(14:17), p.vars = c(11:13), scaled = TRUE)
Paasche.change <- Indices(PAASCHE)
summary(Paasche.change)

## End(Not run)
```

laspeyres

Laspeyres productivity and profitability indices and components

Description

Using Data Envelopment Analysis (DEA), this function measures Laspeyres productivity and profitability indices in levels and changes, as well as their components.

The Laspeyres productivity index uses the previous period prices as aggregators.

Shadow prices of inputs and outputs are also computed.

Usage

```
laspeyres(data, id.var, time.var, x.vars, y.vars, w.vars, p.vars, tech.change = TRUE,
  tech.reg = TRUE, rts = c("vrs", "crs", "nirs", "ndrs"), orientation = c("out",
  "in", "in-out"), parallel = FALSE, cores = max(1, detectCores() - 1), scaled = FALSE)

## S3 method for class 'Laspeyres'
print(object, digits = NULL, ...)
```

Arguments

<code>data</code>	A dataframe containing the required information for productivity and profitability assessment.
<code>id.var</code>	ID variable of the evaluated firms. Can be an integer or a text string.
<code>time.var</code>	Time period variable. Can be an integer or a text string.
<code>x.vars</code>	Input quantity variables. Can be a vector of text strings or integers.
<code>y.vars</code>	Output quantity variables. Can be a vector of text strings or integers.
<code>w.vars</code>	Input price variables. Can be a vector of text strings or integers.
<code>p.vars</code>	Output price variables. Can be a vector of text strings or integers.
<code>tech.change</code>	Logical. If TRUE (default), the model allows for technological change. See also the Details section.

tech.reg	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). See also the Details section.
rts	Character string specifying the returns to scale assumption to be considered. The default value is "vrs" (variable returns to scale). Other possible options are "crs" (constant returns to scale), "nirs" (non-increasing returns to scale), or "ndrs" (non-decreasing returns to scale).
orientation	Character string specifying the orientation to be considered. The default value is "out" (output orientation). Other possible options are "in" (input orientation), and "in-out" (both input and output orientations). For this latter, the geometric mean of input and output orientations is returned.
parallel	Logical. Allows parallel computation. If FALSE (default) the estimation is conducted in sequential mode. If TRUE parallel mode is activated using the number of cores specified in cores.
cores	Integer. Used only if parallel = TRUE. It specifies the number of cores to be used for parallel computation. By default, cores = max(1, detectCores() - 1).
scaled	Logical. Default is FALSE. When set to TRUE, the input and output quantities are rescaled, so that they all have unit means. See also the Details section.
object	An object of class 'Laspeyres'.
digits	The minimum number of significant digits to be printed in values. Default = max(3, getOption("digits") - 3).
...	Currently not used.

Details

laspeyres() allows for parallel computations (when parallel = TRUE, possibly by registering a parallel backend (**doParallel** and **foreach** packages)). The cores argument can be used to specify the number of cores to use. However, when the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).

All DEA linear programs are implemented using the package **Rglpk**.

The tech.change option can totally prohibit any technological change. When tech.change is set to FALSE, this cancels the effect of tech.reg whatever the value of this latter. The tech.reg option, when set to FALSE, rules out negative technological change (i.e. technological regress).

The scaled option is useful when working with very large and/or very small values. By default scaled = FALSE. In such case, laspeyres() may issue a warning when very large (or very small) values are present in the input and output quantity variables. Note that the Laspeyres index may be sensitive to the rescaling especially the mix efficiency component.

The Laspeyres index is not transitive and therefore each firm is compared to itself in the previous period. For the first period, since there is not a previous period, the indices results for this specific period are replaced by NA.

Value

laspeyres() returns a list of class 'Laspeyres' for which a summary of productivity and profitability indices in levels and changes as well as a summary of input (x.vars) and output (y.vars) shadow prices is printed.

This list contains the following items:

Levels	Several elements are provided, depending on the chosen orientation set through orientation argument:
--------	--

REV	Revenues
COST	Costs
PROF	Profitability ratio
P	Aggregated output prices
W	Aggregated input prices
TT	Terms of trade (i.e. P/W)
AO	Aggregated outputs
AI	Aggregated inputs
TFP	Total Factor Productivity
MP	Maximum productivity
TFPE	TFP efficiency score
OTE	Output-oriented technical efficiency score
OSE	Output-oriented scale efficiency score
RAE	Revenue allocative efficiency (equivalent to output-oriented mix efficiency score)
ROSE	Residual output-oriented scale efficiency score
OSME	Output-oriented scale-mix efficiency score
ITE	Input-oriented technical efficiency score
ISE	Input-oriented scale efficiency score
CAE	Cost allocative efficiency (equivalent to input-oriented mix efficiency score)
RISE	Residual input-oriented scale efficiency score
ISME	Input-oriented scale-mix efficiency score
OTE.ITE	Geometric mean of OTE and ITE
OSE.ISE	Geometric mean of OSE and ISE
RAE.CAE	Geometric mean of RAE and CAE
ROSE.RISE	Geometric mean of ROSE and RISE
OSME.ISME	Geometric mean of OSME and ISME
RME	Residual mix efficiency score
RE	Revenue efficiency (when orientation = "out")
CE	Cost efficiency (when orientation = "in")
RE.CE	Geometric mean of RE and CE (when orientation = "in-out")

Indices Change indices of the different elements of Levels are provided. Each change index is prefixed by "d" (e.g. profitability change is named dPROF, output-oriented efficiency change is named dOTE, etc.). Each firm is compared to itself in the previous period. For the first period since there is not a previous period the indices results are replaced by NA for this specific period.

Shadowp The deflated cost input shadow prices and the deflated revenue output shadow prices. Those prices are derived from dual output and input oriented DEA models for each observation in the sample.

From an object of class 'Laspeyres' obtained from `laspeyres()`, the

- `Levels` function extract productivity and profitability **levels**;
- `Indices` function extract productivity and profitability **change indices**; and
- `Shadowprices` function extract input and output **Shadow prices**.

Warning

The `laspeyres()` function will not work with unbalanced panel data, so the user should make sure to use a balanced panel.

Note

All output efficiency scores are estimated *a la* Shephard, while all input efficiency scores are estimated *a la* Farrell. Hence, all efficiency scores are greater than zero and are lower or equal to one.

Author(s)

K Hervé Dakpo, Yann Desjeux, Laure Latruffe

References

Coelli T.J., D.S.P. Rao, C.J. O'Donnell, and G.E. Battese (2005), *An Introduction to Efficiency and Productivity Analysis*. Springer Eds.

O'Donnell C.J. (2011), The sources of productivity change in the manufacturing sectors of the U.S. economy. School of Economics, University of Queensland, Australia. URL <http://www.uq.edu.au/economics/cepa/docs/WP/WP072011.pdf>

See Also

See [Levels\(\)](#) to retrieve a data frame with Laspeyres productivity/profitability scores and components.

See [Indices\(\)](#) to retrieve a data frame with Laspeyres change indices and components.

See [Shadowprices\(\)](#) to retrieve input and output shadow prices.

See also [fisher\(\)](#) and [paasche\(\)](#)

Examples

```
## Laspeyres profitability and productivity levels and indices' computations
## Not run:
Laspeyres.prod <- laspeyres(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), w.vars = c(14:17), p.vars = c(11:13), orientation = "out", scaled = TRUE)
Laspeyres.prod

## End(Not run)
```

Levels

Productivity and profitability indices, in levels, and components

Description

This function extract productivity and profitability (when available) levels from any object created by either `fareprim()`, `fisher()`, `laspeyres()`, `lowe()`, `malm()`, or `paasche()` function.

Usage

```
Levels(object, ...)
```

Arguments

object	Object of class 'FarePrimont', 'Fisher', 'Laspeyres', 'Lowe', 'Malmquist', or 'Paasche'.
...	Currently not used.

Details

- An object of class 'FarePrimont' is usually a result of a call to [fareprim](#)
- An object of class 'Fisher' is usually a result of a call to [fisher](#)
- An object of class 'Laspeyres' is usually a result of a call to [laspeyres](#)
- An object of class 'Lowe' is usually a result of a call to [lowe](#)
- An object of class 'Malmquist' is usually a result of a call to [malm](#)
- An object of class 'Paasche' is usually a result of a call to [paasche](#)

Value

Levels() returns a data frame containing all the elements and observations included in the "Levels" value of the related call.

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

See Also

For details and information on returned values of Levels(), see [fareprim\(\)](#), [fisher\(\)](#), [laspeyres\(\)](#), [lowe\(\)](#), [malm\(\)](#), or [paasche\(\)](#)

Examples

```
## Not run:
LOWE <- lowe(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), w.vars = c(14:17), p.vars = c(11:13), scaled = TRUE)
Lowe.levels <- Levels(LOWE)
dim(Lowe.levels)
head(Lowe.levels)

## End(Not run)
```

lowe

Lowe productivity and profitability indices and components

Description

Using Data Envelopment Analysis (DEA), this function measures Lowe productivity and profitability indices in levels and changes, as well as their components.

Shadow prices of inputs and outputs are also computed.

Usage

```
lowe(data, id.var, time.var, x.vars, y.vars, w.vars, p.vars, tech.change = TRUE,
  tech.reg = TRUE, rts = c("vrs", "crs", "nirs", "ndrs"), orientation = c("out",
  "in", "in-out"), parallel = FALSE, cores = max(1, detectCores() - 1), scaled = FALSE,
  by.id = NULL, by.year = NULL)
```

```
## S3 method for class 'Lowe'
print(object, digits = NULL, ...)
```

Arguments

<code>data</code>	A dataframe containing the required information for productivity and profitability assessment.
<code>id.var</code>	ID variable of the evaluated firms. Can be an integer or a text string.
<code>time.var</code>	Time period variable. Can be an integer or a text string.
<code>x.vars</code>	Input quantity variables. Can be a vector of text strings or integers.
<code>y.vars</code>	Output quantity variables. Can be a vector of text strings or integers.
<code>w.vars</code>	Input price variables. Can be a vector of text strings or integers.
<code>p.vars</code>	Output price variables. Can be a vector of text strings or integers.
<code>tech.change</code>	Logical. If TRUE (default), the model allows for technological change. See also the <code>Details</code> section.
<code>tech.reg</code>	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). See also the <code>Details</code> section.
<code>rts</code>	Character string specifying the returns to scale assumption to be considered. The default value is "vrs" (variable returns to scale). Other possible options are "crs" (constant returns to scale), "nirs" (non-increasing returns to scale), or "ndrs" (non-decreasing returns to scale).
<code>orientation</code>	Character string specifying the orientation to be considered. The default value is "out" (output orientation). Other possible options are "in" (input orientation), and "in-out" (both input and output orientations). For this latter, the geometric mean of input and output orientations is returned.
<code>parallel</code>	Logical. Allows parallel computation. If FALSE (default) the estimation is conducted in sequential mode. If TRUE parallel mode is activated using the number of cores specified in <code>cores</code> .
<code>cores</code>	Integer. Used only if <code>parallel = TRUE</code> . It specifies the number of cores to be used for parallel computation. By default, <code>cores = max(1, detectCores() - 1)</code> .
<code>scaled</code>	Logical. Default is FALSE. When set to TRUE, the input and output quantities are rescaled, so that they all have unit means. See also the <code>Details</code> section.
<code>by.id</code>	Integer specifying the reference observation used for computing the indices (Optional). <code>by.id</code> must range between one and the total number of firms per period. See also the <code>Details</code> section.
<code>by.year</code>	Integer specifying the reference year used for computing the indices (Optional). <code>by.year</code> must range between one and the total number of time periods. See also the <code>Details</code> section.
<code>object</code>	An object of class 'Lowe'.
<code>digits</code>	The minimum number of significant digits to be printed in values. Default = <code>max(3, getOption("digits") - 3)</code> .
<code>...</code>	Currently not used.

Details

`lowe()` allows for parallel computations (when `parallel = TRUE`, possibly by registering a parallel backend (**doParallel** and **foreach** packages)). The `cores` argument can be used to specify the number of cores to use. However, when the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).

All DEA linear programs are implemented using the package **Rglpk**.

The `tech.change` option can totally prohibit any technological change. When `tech.change` is set to `FALSE`, this cancels the effect of `tech.reg` whatever the value of this latter. The `tech.reg` option, when set to `FALSE`, rules out negative technological change (i.e. technological regress). In this case technological change will increment between consecutive periods.

The `scaled` option is useful when working with very large and/or very small values. By default `scaled = FALSE`. In such case, `lowe()` may issue a warning when very large (or very small) values are present in the input and output quantity variables. Note that the Lowe index may be sensitive to the rescaling especially the mix efficiency component.

By default `by.id = NULL` and `by.year = NULL`. Hence, with regards with change indices each observation is by default compared to itself in the first period. `by.id` and `by.year` allow the user to specify a given reference (e.g. a particular observation in a particular period). When the user specifies an observation in `by.id` and when `by.year = NULL`, then the reference observation is `by.id` in the first period. If `by.year` is specified and `by.id = NULL`, then each observation is compared to itself in the specified period of time.

Value

`lowe()` returns a list of class 'Lowe' for which a summary of productivity and profitability indices in levels and changes as well as a summary of input (`x.vars`) and output (`y.vars`) shadow prices is printed.

This list contains the following items:

Levels Several elements are provided, depending on the chosen orientation set through `orientation` argument:

REV	Revenues
COST	Costs
PROF	Profitability ratio
P	Aggregated output prices
W	Aggregated input prices
TT	Terms of trade (i.e. P/W)
AO	Aggregated outputs
AI	Aggregated inputs
TFP	Total Factor Productivity
MP	Maximum productivity
TFPE	TFP efficiency score
OTE	Output-oriented technical efficiency score
OSE	Output-oriented scale efficiency score
OME	Output-oriented mix efficiency score
ROSE	Residual output-oriented scale efficiency score
OSME	Output-oriented scale-mix efficiency score
ITE	Input-oriented technical efficiency score
ISE	Input-oriented scale efficiency score
IME	Input-oriented mix efficiency score
RISE	Residual input-oriented scale efficiency score
ISME	Input-oriented scale-mix efficiency score
OTE.ITE	Geometric mean of OTE and ITE
OSE.ISE	Geometric mean of OSE and ISE
OME.IME	Geometric mean of OME and IME
ROSE.RISE	Geometric mean of ROSE and RISE
OSME.ISME	Geometric mean of OSME and ISME
RME	Residual mix efficiency score

Indices	Change indices of the different elements of Levels are provided. Each change index is prefixed by "d" (e.g. profitability change is named dPROF, output-oriented efficiency change is named dOTE, etc.).
Shadowp	The deflated cost input shadow prices and the deflated revenue output shadow prices. Those prices are derived from dual output and input oriented DEA models for each observation in the sample.

From an object of class 'Lowe' obtained from `lowe()`, the

- `Levels` function extract productivity and profitability **levels**;
- `Indices` function extract productivity and profitability **change indices**; and
- `Shadowprices` function extract input and output **Shadow prices**.

Warning

The `lowe()` function might not properly work with unbalanced panel data, so the user should make sure to use a balanced panel.

Note

All output efficiency scores are estimated *a la* Shephard, while all input efficiency scores are estimated *a la* Farrell. Hence, all efficiency scores are greater than zero and are lower or equal to one.

Author(s)

K Hervé Dakpo, Yann Desjeux, Laure Latruffe

References

O'Donnell C.J.(2008), An aggregate quantity-price framework for measuring and decomposing productivity and profitability change. School of Economics, University of Queensland, Australia.

O'Donnell C.J. (2011), The sources of productivity change in the manufacturing sectors of the U.S. economy. School of Economics, University of Queensland, Australia. URL <http://www.uq.edu.au/economics/cepa/docs/WP/WP072011.pdf>

O'Donnell C.J. (2012), Nonparametric estimates of the components of productivity and profitability change in U.S. agriculture. *American Journal of Agricultural Economics*, **94**(4), 873–890.

See Also

See `Levels()` to retrieve a data frame with Lowe productivity/profitability scores and components.

See `Indices()` to retrieve a data frame with Lowe change indices and components.

See `Shadowprices()` to retrieve input and output shadow prices.

See also `fareprim()`

Examples

```
# Lowe profitability and productivity levels and indices' computations
## Not run:
Lowe.prod <- lowe(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
y.vars = c(4:6), w.vars = c(14:17), p.vars = c(11:13), orientation = "in-out", scaled = TRUE,
by.id = 1, by.year = 1)
```

```

Lowe.prod
## End(Not run)

```

malm

Malmquist productivity indices and components

Description

This function computes Malmquist productivity indices, in levels and changes, as well as their components.

Usage

```

malm(data, id.var, time.var, x.vars, y.vars, tech.reg = TRUE, rts = c("vrs", "crs",
  "nirs", "ndrs"), orientation = c("out", "in"), parallel = FALSE, cores = max(1,
  detectCores() - 1), scaled = FALSE)

## S3 method for class 'Malmquist'
print(object, digits = NULL, ...)

```

Arguments

data	A dataframe containing the required information for productivity assessment.
id.var	ID variable of the evaluated firms. Can be an integer or a text string.
time.var	Time period variable. Can be an integer or a text string.
x.vars	Input quantity variables. Can be a vector of text strings or integers.
y.vars	Output quantity variables. Can be a vector of text strings or integers.
tech.reg	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). See also the Details section.
rts	Character string specifying the returns to scale assumption to be considered. The default value is "vrs" (variable returns to scale). Other possible options are "crs" (constant returns to scale), "nirs" (non-increasing returns to scale), or "ndrs" (non-decreasing returns to scale).
orientation	Character string specifying the orientation to be considered. The default value is "out" (output orientation). The other possible option is "in" (input orientation).
parallel	Logical. Allows parallel computation. If FALSE (default) the estimation is conducted in sequential mode. If TRUE parallel mode is activated using the number of cores specified in cores.
cores	Integer. Used only if parallel = TRUE. It specifies the number of cores to be used for parallel computation. By default, cores = max(1, detectCores() - 1).
scaled	Logical. Default is FALSE. When set to TRUE, the input and output quantities are rescaled, so that they all have unit means. See also the Details section.
object	An object of class 'Malmquist'.
digits	The minimum number of significant digits to be printed in values. Default = max(3, getOption("digits") - 3).
...	Currently not used.

Details

`malm()` allows for parallel computations (when `parallel = TRUE`, possibly by registering a parallel backend (**doParallel** and **foreach** packages)). The `cores` argument can be used to specify the number of cores to use. However, when the sample size is small, it is recommended to keep the `parallel` option to its default value (`FALSE`).

All DEA linear programs are implemented using the package **Rglpk**.

The `tech.reg` option, when set to `FALSE`, rules out technological regression. Therefore, the technological change will increment between consecutive periods.

The `scaled` option is useful when working with very large and/or very small values. By default `scaled = FALSE`. In such case, `malm()` may issue a warning when very large (or very small) values are present in the input and output quantity variables. It is worth noting that all the distance functions required for the Malmquist index estimation are radial measures which verify the translation invariance property. Hence, unless very large or very small values are present, the Malmquist index is insensitive to the rescaling option.

Value

`malm()` returns a list of class 'Malmquist' for which a summary of productivity indices in levels and changes is printed.

This list contains the following items:

Levels	It contains the Shephard distance function estimates, useful to compute and decompose the Malmquist productivity index. These distance functions use input and output quantities for period 1 and period 0. The prefix "c" stands for constant returns to scale ("crs") and "v" for all the other types of returns to scale. The suffix "o" means output-oriented while "i" refers to input-oriented. The distance function names are displayed with three digits: (i) the first digit represents the period of the reference technology, (ii) the second digit represents the period of the inputs, and (iii) the third digit represents the period of the outputs. For instance <code>c010o</code> means output-oriented efficiency under constant returns to scale (CRS), with the reference technology of period 0, inputs of period 1 and outputs of period 0. In addition to the <code>id.var</code> variable and periods 1 and 0, the dataframe therefore contains, depending on the chosen orientation set through <code>orientation</code> : <code>c111o</code> , <code>c100o</code> , <code>c011o</code> , <code>c000o</code> , <code>c110o</code> , <code>c010o</code> , <code>c111i</code> , <code>c100i</code> , <code>c011i</code> , <code>c000i</code> , <code>c110i</code> , and <code>c010i</code> . When the returns to scale option (<code>rts</code>) is different from "crs", then <code>v111o</code> , <code>v000o</code> , <code>v111i</code> and <code>v000i</code> are also provided depending on the orientation.
Indices	Malmquist productivity change indices and their various components are provided, depending on the chosen orientation set through <code>orientation</code> argument.

<code>malmquist</code>	Malmquist productivity index
<code>effch</code>	Efficiency change
<code>tech</code>	Technological change
<code>obtech</code>	Output-biased technological change
<code>ibtech</code>	Input-biased technological change
<code>matech</code>	Magnitude component
<code>pure.out.effch</code>	Pure output efficiency change
<code>out.scalech</code>	Output scale efficiency change
<code>pure.inp.effch</code>	Pure input efficiency change
<code>inp.scalech</code>	Input scale efficiency change

Note that:

1. obtech (*Output-biased technological change*), ibtech (*Input-biased technological change*), and matech (*Magnitude component*) are components of technological change (tech).
2. pure.out.effch (*Pure output efficiency change*), out.scalech (*Output scale efficiency change*) are components of efficiency change (effch), when `rts != "crs"` and `orientation = "out"`.
3. pure.inp.effch (*Pure input efficiency change*), and inp.scalech (*Input scale efficiency change*) are components of efficiency change (effch), when `rts != "crs"` and `orientation = "in"`.

From an object of class 'Malmquist' obtained from `malm()`, the

- [Levels](#) function extract productivity and profitability **levels**; and
- [Indices](#) function extract productivity and profitability **change indices**.

Warning

The `malm()` function does not work with unbalanced panel data, so the user should make sure to provide a balanced panel.

Note

The Malmquist and its components are computed such that both orientation's results can be read in the same way (growth when greater than one and decline when lower than one). Moreover under `rts = "crs"`, both orientation options (i.e. "out" and "in") yield the same results.

Author(s)

K Hervé Dakpo, Yann Desjeux, Laure Latruffe

References

Färe R., and Grosskopf S. (1996), *Intertemporal Production Frontiers: With Dynamic DEA*. Springer Eds.

See Also

See [Levels\(\)](#) to retrieve a data frame with Malmquist productivity/profitability scores and components.

See [Indices\(\)](#) to retrieve a data frame with Malmquist change indices and components.

Examples

```
# Malmquist productivity indices compare each observation in period 1
# to the same observation in period 0
## Not run:
Malmquist <- malm(data = usagri, id.var = "States", time.var = "Years",
  x.vars = c("q.capital", "q.land", "q.labor", "q.materials"),
  y.vars = c("q.livestock", "q.crop", "q.other"), rts = "nirs", scaled = TRUE)
Malmquist

## End(Not run)
```

paasche

*Paasche productivity and profitability indices and components***Description**

Using Data Envelopment Analysis (DEA), this function measures Paasche productivity and profitability indices in levels and changes, as well as their components.

The Paasche index uses current period prices as aggregators.

Shadow prices of inputs and outputs are also computed.

Usage

```
paasche(data, id.var, time.var, x.vars, y.vars, w.vars, p.vars, tech.change = TRUE,
        tech.reg = TRUE, rts = c("vrs", "crs", "nirs", "ndrs"), orientation = c("out",
        "in", "in-out"), parallel = FALSE, cores = max(1, detectCores() - 1), scaled = FALSE)
```

```
## S3 method for class 'Paasche'
print(object, digits = NULL, ...)
```

Arguments

<code>data</code>	A dataframe containing the required information for productivity and profitability assessment.
<code>id.var</code>	ID variable of the evaluated firms. Can be an integer or a text string.
<code>time.var</code>	Time period variable. Can be an integer or a text string.
<code>x.vars</code>	Input quantity variables. Can be a vector of text strings or integers.
<code>y.vars</code>	Output quantity variables. Can be a vector of text strings or integers.
<code>w.vars</code>	Input price variables. Can be a vector of text strings or integers.
<code>p.vars</code>	Output price variables. Can be a vector of text strings or integers.
<code>tech.change</code>	Logical. If TRUE (default), the model allows for technological change. See also the Details section.
<code>tech.reg</code>	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). See also the Details section.
<code>rts</code>	Character string specifying the returns to scale assumption to be considered. The default value is "vrs" (variable returns to scale). Other possible options are "crs" (constant returns to scale), "nirs" (non-increasing returns to scale), or "ndrs" (non-decreasing returns to scale).
<code>orientation</code>	Character string specifying the orientation to be considered. The default value is "out" (output orientation). Other possible options are "in" (input orientation), and "in-out" (both input and output orientations). For this latter, the geometric mean of input and output orientations is returned.
<code>parallel</code>	Logical. Allows parallel computation. If FALSE (default) the estimation is conducted in sequential mode. If TRUE parallel mode is activated using the number of cores specified in <code>cores</code> .
<code>cores</code>	Integer. Used only if <code>parallel = TRUE</code> . It specifies the number of cores to be used for parallel computation. By default, <code>cores = max(1, detectCores() - 1)</code> .

scaled	Logical. Default is FALSE. When set to TRUE, the input and output quantities are rescaled, so that they all have unit means. See also the Details section.
object	An object of class 'Paasche'.
digits	The minimum number of significant digits to be printed in values. Default = $\max(3, \text{getOption("digits")} - 3)$.
...	Currently not used.

Details

paasche() allows for parallel computations (when `parallel = TRUE`, possibly by registering a parallel backend (**doParallel** and **foreach** packages). The `cores` argument can be used to specify the number of cores to use. However, when the sample size is small, it is recommended to keep the `parallel` option to its default value (FALSE).

All DEA linear programs are implemented using the package **Rglpk**.

The `tech.change` option can totally prohibit any technological change. When `tech.change` is set to FALSE, this cancels the effect of `tech.reg` whatever the value of this latter. The `tech.reg` option, when set to FALSE, rules out negative technological change (i.e. technological regress).

The `scaled` option is useful when working with very large and/or very small values. By default `scaled = FALSE`. In such case, paasche() may issue a warning when very large (or very small) values are present in the input and output quantity variables. Note that the Paasche index may be sensitive to the rescaling especially the mix efficiency component.

The Laspeyres index is not transitive and therefore each firm is compared to itself in the previous period. For the first period, since there is not a previous period, the indices results for this specific period are replaced by NA.

Value

lpaasche() returns a list of class 'Paasche' for which a summary of productivity and profitability indices in levels and changes as well as a summary of input (`x.vars`) and output (`y.vars`) shadow prices is printed.

This list contains the following items:

Levels Several elements are provided, depending on the chosen orientation set through orientation argument:

REV	Revenues
COST	Costs
PROF	Profitability ratio
P	Aggregated output prices
W	Aggregated input prices
TT	Terms of trade (i.e. P/W)
AO	Aggregated outputs
AI	Aggregated inputs
TFP	Total Factor Productivity
MP	Maximum productivity
TFPE	TFP efficiency score
OTE	Output-oriented technical efficiency score
OSE	Output-oriented scale efficiency score
RAE	Revenue allocative efficiency (equivalent to output-oriented mix efficiency score)
ROSE	Residual output-oriented scale efficiency score
OSME	Output-oriented scale-mix efficiency score

ITE	Input-oriented technical efficiency score
ISE	Input-oriented scale efficiency score
CAE	Cost allocative efficiency (equivalent to input-oriented mix efficiency score)
RISE	Residual input-oriented scale efficiency score
ISME	Input-oriented scale-mix efficiency score
OTE . ITE	Geometric mean of OTE and ITE
OSE . ISE	Geometric mean of OSE and ISE
RAE . CAE	Geometric mean of RAE and CAE
ROSE . RISE	Geometric mean of ROSE and RISE
OSME . ISME	Geometric mean of OSME and ISME
RME	Residual mix efficiency score
RE	Revenue efficiency (when orientation = "out")
CE	Cost efficiency (when orientation = "in")
RE . CE	Geometric mean of RE and CE (when orientation = "in-out")
Indices	Change indices of the different elements of Levels are provided. Each change index is prefixed by "d" (e.g. profitability change is named dPROF, output-oriented efficiency change is named dOTE, etc.). Each firm is compared to itself in the previous period. For the first period since there is not a previous period the indices results are replaced by NA for this specific period.
Shadowp	The deflated cost input shadow prices and the deflated revenue output shadow prices. Those prices are derived from dual output and input oriented DEA models for each observation in the sample.

From an object of class 'Paasche' obtained from `paasche()`, the

- `Levels` function extract productivity and profitability **levels**;
- `Indices` function extract productivity and profitability **change indices**; and
- `Shadowprices` function extract input and output **Shadow prices**.

Warning

The `paasche()` function will not work with unbalanced panel data, so the user should make sure to use a balanced panel.

Note

All output efficiency scores are estimated *a la* Shephard, while all input efficiency scores are estimated *a la* Farrell. Hence, all efficiency scores are greater than zero and are lower or equal to one.

Author(s)

K Hervé Dakpo, Yann Desjeux, Laure Latruffe

References

Coelli T.J., D.S.P. Rao, C.J. O'Donnell, and G.E. Battese (2005), *An Introduction to Efficiency and Productivity Analysis*. Springer Eds.

O'Donnell C.J. (2011), The sources of productivity change in the manufacturing sectors of the U.S. economy. School of Economics, University of Queensland, Australia. URL <http://www.uq.edu.au/economics/cepa/docs/WP/WP072011.pdf>

See Also

See [Levels\(\)](#) to retrieve a data frame with Paasche productivity/profitability scores and components.

See [Indices\(\)](#) to retrieve a data frame with Paasche change indices and components.

See [Shadowprices\(\)](#) to retrieve input and output shadow prices.

See also [fisher\(\)](#) and [laspeyres\(\)](#)

Examples

```
# Paasche profitability and productivity levels and indices' computations
## Not run:
Paasche.prod <- paasche(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), w.vars = c(14:17), p.vars = c(11:13), orientation = "out", scaled = TRUE)
Paasche.prod

## End(Not run)
```

Shadowprices

Shadow prices used to compute productivity and profitability indices

Description

From any object created by either [fareprim\(\)](#), [fisher\(\)](#), [laspeyres\(\)](#), [lowe\(\)](#), or [paasche\(\)](#) function, this function extract the deflated cost input shadow prices along with the deflated revenue shadow prices used to compute productivity and profitability indices.

Usage

```
Shadowprices(object, ...)
```

Arguments

object	Object of class 'FarePrimont', 'Fisher', 'Laspeyres', 'Lowe', or 'Paasche'.
...	Currently not used.

Details

- An object of class 'FarePrimont' is usually a result of a call to [fareprim](#)
- An object of class 'Fisher' is usually a result of a call to [fisher](#)
- An object of class 'Laspeyres' is usually a result of a call to [laspeyres](#)
- An object of class 'Lowe' is usually a result of a call to [lowe](#)
- An object of class 'Paasche' is usually a result of a call to [paasche](#)

Value

[Shadowprices\(\)](#) returns a data frame containing all the elements and observations included in the "Shadowp" value of the related call.

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

See Also

For details and information on returned values of `Shadowprices()`, see `fareprim()`, `fisher()`, `laspeyres()`, `lowe()`, or `paasche()`

Examples

```
## Not run:
FISHER <- fisher(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), w.vars = c(14:17), p.vars = c(11:13), orientation = "out", scaled = TRUE)
Fisher.shadowprices <- Shadowprices(FISHER)
head(Fisher.shadowprices)

## End(Not run)
```

usagri	<i>Price indices and implicit quantities of USA farm outputs and inputs by State, 1995-2004</i>
--------	---

Description

This data set from the United States Department of Agriculture (USDA) and its Economic Research Service department contains USA agriculture's input and output quantities along with their respective price indices for 48 States.

All quantities are expressed in thousand US\$1996 and prices are relative to Alabama 1996 = 1.

Usage

```
usagri
```

Format

A data frame with 480 observations on the following 17 variables.

States 48 States of the USA identified with two capital letters.

States.num State number.

Years Year.

q.livestock Livestock and animal products' quantity, in thousand US\$1996.

q.crop Crops' quantity, in thousand US\$1996.

q.other Other farm-related productions' quantity, in thousand US\$1996.

q.capital Capital services' quantity, in thousand US\$1996.

q.land Land services' quantity, in thousand US\$1996.

q.labor Labor services' quantity, in thousand US\$1996.

q.materials Total intermediate input quantity, in thousand US\$1996.

p.livestock Livestock and animal products' relative price (1 = Alabama 1996).

p.crop Crops' relative price (1 = Alabama 1996).

p.other Other farm-related productions' relative price (1 = Alabama 1996).

p.capital Capital services' relative price (1 = Alabama 1996).

p.land Land service flows' relative price (1 = Alabama 1996).

p.labor Labor services' relative price (1 = Alabama 1996).

p.materials Total intermediate inputs' relative price (1 = Alabama 1996).

Details

More description on the data and the different variables can be found in the references.

Source

<http://www.ers.usda.gov/data-products/agricultural-productivity-in-the-us.aspx>

References

Ball V.E., Gollop F.M., Kelly-Hawke A., and Swinand G.P. (1999), Patterns of state productivity growth in the US farm sector: Linking state and aggregate models. *American Journal of Agricultural Economics*, **81**, 164–179.

Ball V.E., Hallahan C., and Nehring R. (2004), Convergence of productivity: An analysis of the catch-up hypothesis within a panel of states. *American Journal of Agricultural Economics*, **86**(5), 1315–1321.

Examples

```
head(usagri)
str(usagri)
summary(usagri)
```

Index

*Topic **datasets**

usagri, 25

*Topic **manip**

Indices, 9

Levels, 13

Shadowprices, 24

*Topic **models**

fareprim, 2

fisher, 6

laspeyres, 10

lowe, 14

malm, 18

paasche, 21

fareprim, 2, 9, 14, 24

fareprim(), 10, 14, 17, 25

fisher, 6, 9, 14, 24

fisher(), 10, 13, 14, 24, 25

Indices, 4, 8, 9, 12, 17, 20, 23

Indices(), 5, 9, 13, 17, 20, 24

laspeyres, 9, 10, 14, 24

laspeyres(), 9, 10, 14, 24, 25

Levels, 4, 8, 12, 13, 17, 20, 23

Levels(), 5, 9, 13, 17, 20, 24

lowe, 9, 14, 14, 24

lowe(), 5, 10, 14, 25

malm, 9, 14, 18

malm(), 10, 14

paasche, 9, 14, 21, 24

paasche(), 9, 10, 13, 14, 25

print.FarePrimont (fareprim), 2

print.Fisher (fisher), 6

print.Laspeyres (laspeyres), 10

print.Lowe (lowe), 14

print.Malmquist (malm), 18

print.Paasche (paasche), 21

Shadowprices, 4, 8, 12, 17, 23, 24

Shadowprices(), 5, 9, 13, 17, 24

usagri, 25