Package 'productivity'

July 20, 2017

Version 1.0.0

Date 2017-07-20

Title Indices of Productivity Using Data Envelopment Analysis (DEA)

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

Description Levels and changes of productivity and profitability are measured with various indices.
In addition to the classic Malmquist productivity index, the 'productivity' package contains also the multiplicatively complete Färe-Primont, Fisher, Laspeyres, Lowe, and Paasche indices.
Färe-Primont and Lowe indices verify the transitivity property and can therefore be used for multilateral or multitemporal comparison.
Fisher, Laspeyres, Malmquist, and Paasche indices are 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 can be 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. Deflated shadow prices, used in those indices' computations, are also returned.
The package also allows to prohibit negative technological change.
All computations are carried out with the nonparametric Data Envelopment Analysis (DEA), and several assumptions regarding returns to scale are available.
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, plm, Rglpk

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation no

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

Repository CRAN

Date/Publication 2017-07-20 10:48:32 UTC

R topics documented:

`Changes` *Productivity and profitability change indices*

Description

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

Usage

```
Changes(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

The function returns a data frame containing all the elements and observations included in the "Changes" constituent element of object.

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

See Also

For details and information on returned values, see fareprim, fisher, laspeyres, lowe, malm, or paasche.
See also:
- Levels for productivity and profitability levels; and
- Shadowp for shadow prices.

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 <- Changes(PAASCHE)
    summary(Paasche.change)

## End(Not run)
```

---

fareprim *Fare-Primont productivity and profitability index*

Description

Using Data Envelopment Analysis (DEA), this function measures productivity and profitability in levels and changes with Färe-Primont index.

Profitability measures are only provided when price information is specified.

Deflated 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(x, digits = NULL, ...)
```

Arguments

data	A dataframe containing the required information for measuring productivity and profitability.
id.var	Firms' ID variable. 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 (Optional). Can be a vector of text strings or integers. NULL by default, so only productivity is measured.
p.vars	Output price variables (Optional). Can be a vector of text strings or integers. NULL by default, so only productivity is measured.
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. 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. The default value is "out" (output-orientation). Other possible options are "in" (input-orientation), and "in-out" (both input- and output-orientations). For the latter, the geometric mean of input- and output-orientations' results 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. 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.
x	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 allows to prohibit technological change. When `tech.change` is set to `FALSE`, this overrides the effect of `tech.reg`. 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 (>1e5) and/or very small (<1e-4) 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 Färe-Primont index may be sensitive to the rescaling, especially the mix efficiency component.

By default `by.id = NULL` and `by.year = NULL`. This means that in the computation of change indices, each observation is by default compared to itself in the first period. `by.id` and `by.year` allow to specify a reference (e.g. a specific observation in a specific period). If `by.id` is specified and `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) measures in levels and changes is printed, as well as input (`x.vars`) and output (`y.vars`) deflated shadow prices.

This list contains the following items:

`Levels` Several elements are provided, depending on the `orientation` specified:

`REV`	Revenues (if `w.vars` and `p.vars` are specified)
`COST`	Costs (if `w.vars` and `p.vars` are specified)
`PROF`	Profitability (if `w.vars` and `p.vars` are specified)
`P`	Aggregated output prices (if `w.vars` and `p.vars` are specified)
`W`	Aggregated input prices (if `w.vars` and `p.vars` are specified)
`TT`	Terms of trade (i.e. `P/W`) (if `w.vars` and `p.vars` are specified)
`AO`	Aggregated outputs
`AI`	Aggregated inputs
`TFP`	Total Factor Productivity (TFP)
`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 (when orientation = "in-out")
OSE.ISE	Geometric mean of OSE and ISE (when orientation = "in-out")
OME.IME	Geometric mean of OME and IME (when orientation = "in-out")
ROSE.RISE	Geometric mean of ROSE and RISE (when orientation = "in-out")
OSME.ISME	Geometric mean of OSME and ISME (when orientation = "in-out")
RME	Residual mix efficiency score

`Changes` Change indices of the different elements of `Levels` are provided. Each change is prefixed by `"d"` (e.g. profitability change is denoted `dPROF`, output-oriented efficiency change is denoted `dOTE`, etc.).

`Shadowp` The deflated cost input shadow prices and the deflated revenue output shadow prices of the representative observation used to compute the Färe-Primont index. These prices are derived from dual input- and output-oriented DEA models.

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

- `Levels` function extracts productivity and profitability **levels**;
- `Changes` function extracts productivity and profitability **change indices**; and
- `Shadowp` function extracts input and output **deflated shadow prices**.

Warning

The `fareprim()` function might not properly work with unbalanced panel data.

Note

All output-oriented efficiency scores are computed *a la* Shephard, while all input-oriented efficiency scores are computed *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 and profitability in levels and components.
See `Changes` to retrieve a data frame with Färe-Primont productivity and profitability changes and components.
See `Shadowp` to retrieve deflated input and output shadow prices.
See also `lowe` for computations with an alternative transitive index.

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 index*

Description

Using Data Envelopment Analysis (DEA), this function measures productivity and profitability in levels and changes with Fisher index.

Deflated shadow prices of inputs and outputs are also computed.

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

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(x, digits = NULL, ...)
```

Arguments

data	A dataframe containing the required information for measuring productivity and profitability.
id.var	Firms' ID variable. 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. 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. The default value is "out" (output-orientation). Other possible options are "in" (input-orientation), and "in-out" (both input- and output-orientations). For the latter, the geometric mean of input- and output-orientations' results 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. See also the Details section.
x	An object of class 'Fisher'.
digits	The minimum number of significant digits to be printed in values. Default = max(3, getOption("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 allows to prohibit technological change. When `tech.change` is set to `FALSE`, this overrides the effect of `tech.reg`. 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 (>1e5) and/or very small (<1e-4) 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. Since there is no previous period for the first period, the results for this first period are replaced by `NA`.

Value

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

This list contains the following items:

`Levels` Several elements are provided, depending on the `orientation` specified:

`REV`	Revenues
`COST`	Costs
`PROF`	Profitability
`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 (TFP)
`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 (when orientation = "in-out")
OSE.ISE	Geometric mean of OSE and ISE (when orientation = "in-out")
RAE.CAE	Geometric mean of RAE and CAE (when orientation = "in-out")
ROSE.RISE	Geometric mean of ROSE and RISE (when orientation = "in-out")
OSME.ISME	Geometric mean of OSME and ISME (when orientation = "in-out")
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")

Changes — Change indices of the different elements of Levels are provided. Each change is prefixed by "d" (e.g. profitability change is denoted dPROF, output-oriented efficiency change is denoted dOTE, etc.). Each firm is compared to itself in the previous period. Since there is no previous period for the first period, the results for this first period are replaced by NA.

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

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

- `Levels` function extracts productivity and profitability **levels**;
- `Changes` function extracts productivity and profitability **change indices**; and
- `Shadowp` function extracts input and output **deflated shadow prices**.

Warning

The `fisher()` function will not work with unbalanced data.

Note

All output-oriented efficiency scores are computed *a la* Shephard, while all input-oriented efficiency scores are computed *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 and profitability in levels and components.
See `Changes` to retrieve a data frame with Fisher productivity and profitability changes and components.
See `Shadowp` to retrieve deflated input and output shadow prices.
See also `laspeyres` and `paasche` for computations with alternative indices.

Examples

```
## Fisher profitability and productivity levels and changes' 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)
```

`laspeyres` *Laspeyres productivity and profitability index*

Description

Using Data Envelopment Analysis (DEA), this function measures productivity and profitability in levels and changes with Laspeyres index.

Deflated shadow prices of inputs and outputs are also computed.

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

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(x, digits = NULL, ...)
```

Arguments

data	A dataframe containing the required information for measuring productivity and profitability.
id.var	Firms' ID variable. 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. 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. The default value is "out" (output-orientation). Other possible options are "in" (input-orientation), and "in-out" (both input- and output-orientations). For the latter, the geometric mean of input- and output-orientations' results 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. See also the Details section.
x	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 allows to prohibit technological change. When tech.change is set to FALSE, this overrides the effect of tech.reg. 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 (>1e5) and/or very small (<1e-4) 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. Since there is no previous period for the first period, the results for this first period are replaced by NA.

Value

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

This list contains the following items:

Levels Several elements are provided, depending on the orientation specified:

REV	Revenues
COST	Costs
PROF	Profitability
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 (TFP)
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 (when orientation = "in-out")
OSE.ISE	Geometric mean of OSE and ISE (when orientation = "in-out")
RAE.CAE	Geometric mean of RAE and CAE (when orientation = "in-out")
ROSE.RISE	Geometric mean of ROSE and RISE (when orientation = "in-out")
OSME.ISME	Geometric mean of OSME and ISME

(when orientation = "in-out")

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")

Changes Change indices of the different elements of Levels are provided. Each change is prefixed by "d" (e.g. profitability change is denoted dPROF, output-oriented efficiency change is denoted dOTE, etc.). Each firm is compared to itself in the previous period. Since there is no previous period for the first period, the results for this first period are replaced by NA.

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

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

- Levels function extracts productivity and profitability **levels**;
- Changes function extracts productivity and profitability **change indices**; and
- Shadowp function extracts input and output **deflated shadow prices**.

Warning

The laspeyres() function will not work with unbalanced data.

Note

All output-oriented efficiency scores are computed *a la* Shephard, while all input-oriented efficiency scores are computed *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 and profitability in levels and components.
See `Changes` to retrieve a data frame with Laspeyres productivity and profitability changes and components.
See `Shadowp` to retrieve deflated input and output shadow prices.
See also `fisher` and `paasche` for computation with alternative indices.

Examples

```
## Laspeyres profitability and productivity levels and changes' 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 levels*

Description

This function extracts 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

The function returns a data frame containing all the elements and observations included in the "Levels" constituent element of object.

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

See Also

For details and information on returned values, see fareprim, fisher, laspeyres, lowe, malm, or paasche.
See also:
- Changes for productivity and profitability change indices; and
- Shadowp for shadow prices.

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 index*

Description

Using Data Envelopment Analysis (DEA), this function measures productivity and profitability in levels and changes with Lowe index.

Deflated 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(x, digits = NULL, ...)
```

Arguments

data	A dataframe containing the required information for measuring productivity and profitability.
id.var	Firms' ID variable. 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. 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. The default value is "out" (output-orientation). Other possible options are "in" (input-orientation), and "in-out" (both input- and output-orientations). For the latter, the geometric mean of input- and output-orientations' results 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. 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.
x	An object of class 'Lowe'.
digits	The minimum number of significant digits to be printed in values. Default = max(3, getOption("digits") - 3).
...	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 allows to prohibit technological change. When `tech.change` is set to `FALSE`, this cancels the effect of `tech.reg` whatever the value of the 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 (>1e5) and/or very small (<1e-4) 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`. This means that in the computation of change indices, each observation is by default compared to itself in the first period. `by.id` and `by.year` allow to specify a reference (e.g. a specific observation in a specific period). If `by.id` is specified and `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 measures in levels and changes, as well as a summary of input (`x.vars`) and output (`y.vars`) deflated shadow prices, is printed.

This list contains the following items:

`Levels` Several elements are provided, depending on the `orientation` specified:

`REV`	Revenues
`COST`	Costs
`PROF`	Profitability
`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 (TFP)
`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 (when orientation = "in-out")
OSE.ISE	Geometric mean of OSE and ISE (when orientation = "in-out")
OME.IME	Geometric mean of OME and IME (when orientation = "in-out")
ROSE.RISE	Geometric mean of ROSE and RISE (when orientation = "in-out")
OSME.ISME	Geometric mean of OSME and ISME (when orientation = "in-out")
RME	Residual mix efficiency score

Changes — Change indices of the different elements of Levels are provided. Each change is prefixed by "d" (e.g. profitability change is denoted dPROF, output-oriented efficiency change is denoted dOTE, etc.).

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

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

- Levels function extracts productivity and profitability **levels**;
- Changes function extracts productivity and profitability **change indices**; and
- Shadowp function extracts input and output **deflated shadow prices**.

Warning

The lowe() function might not properly work with unbalanced panel data.

Note

All output-oriented efficiency scores are computed *a la* Shephard, while all input-oriented efficiency scores are computed *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. aAgriculture. *American Journal of Agricultural Economics*, **94**(4), 873–890.

See Also

See Levels to retrieve a data frame with Lowe productivity and profitability in levels and components.
See Changes to retrieve a data frame with Lowe productivity and profitability changes and components.
See Shadowp to retrieve deflated input and output shadow prices.
See also fareprim for computations with an alternative transitive index.

Examples

```
## Lowe profitability and productivity levels and changes' 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 index*

Description

Using Data Envelopment Analysis (DEA), this function measures productivity with Malmquist index.

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(x, digits = NULL, ...)
```

Arguments

data	A dataframe containing the required information for measuring productivity.
id.var	Firms' ID variable. 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. 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. 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. See also the Details section.
x	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 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 (>1e5) and/or very small (<1e-4) 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. Note that all the distance functions required for computing the Malmquist index 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 measures 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 (i.e. `"vrs"`, `"nirs"`, or `"ndrs"`). 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 `c010o` 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 `id.var` variable and periods `1` and `0`, the dataframe therefore contains, depending on the orientation: `c111o`, `c100o`, `c011o`, `c000o`, `c110o`, `c010o`, `c111i`, `c100i`, `c011i`, `c000i`, `c110i`, and `c010i`.

When the returns to scale option (`rts`) is different from `"crs"`, then `v111o`, `v000o`, `v111i` and `v000i` are also provided depending on the orientation.

`Changes` Malmquist productivity index and its components are provided, depending on the orientation.

`malmquist`	Malmquist productivity index
`effch`	Efficiency change
`tech`	Technological change
`obtech`	Output-biased technological change
`ibtech`	Input-biased technological change
`matech`	Magnitude component
`pure.out.effch`	Pure output efficiency change
`out.scalech`	Output scale efficiency change
`pure.inp.effch`	Pure input efficiency change
`inp.scalech`	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) and `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 extracts Shephard distance function estimates; and
- `Changes` function extracts Malmquist productivity index and its components.

Warning

The `malm()` function will not work with unbalanced data.

Note

The Malmquist productivity index 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 Shephard distance function estimates.
See `Changes` to retrieve a data frame with Malmquist productivity index and its components.

Examples

```
## Malmquist productivity index compares 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 index*

Description

Using Data Envelopment Analysis (DEA), this function measures productivity and profitability in levels and changes with Paasche index.

Deflated shadow prices of inputs and outputs are also computed.

The Paasche index uses current period prices as aggregators.

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(x, digits = NULL, ...)
```

Arguments

data	A dataframe containing the required information for measuring productivity and profitability.
id.var	Firms' ID variable. 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. 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. The default value is "out" (output-orientation). Other possible options are "in" (input-orientation), and "in-out" (both input- and output-orientations). For the latter, the geometric mean of input- and output-orientations' results 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. See also the Details section.
x	An object of class 'Paasche'.
digits	The minimum number of significant digits to be printed in values. Default = max(3, 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 allows to prohibit technological change. When tech.change is set to FALSE, this overrides the effect of tech.reg. 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 (>1e5) and/or very small (<1e-4) 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 Paasche index is not transitive and therefore each firm is compared to itself in the previous period. Since there is no previous period for the first period, the results for this first period are replaced by NA.

Value

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

This list contains the following items:

Levels — Several elements are provided, depending on the orientation specified:

REV	Revenues
COST	Costs
PROF	Profitability
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 (TFP)
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 (when orientation = "in-out")
OSE.ISE	Geometric mean of OSE and ISE (when orientation = "in-out")
RAE.CAE	Geometric mean of RAE and CAE (when orientation = "in-out")
ROSE.RISE	Geometric mean of ROSE and RISE (when orientation = "in-out")
OSME.ISME	Geometric mean of OSME and ISME (when orientation = "in-out")
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")

`Changes` Change indices of the different elements of `Levels` are provided. Each change is prefixed by `"d"` (e.g. profitability change is denoted `dPROF`, output-oriented efficiency change is denoted `dOTE`, etc.). Each firm is compared to itself in the previous period. Since there is no previous period for the first period, the results for this first period are replaced by `NA`.

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

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

- `Levels` function extracts productivity and profitability **levels**;
- `Changes` function extracts productivity and profitability **change indices**; and
- `Shadowp` function extracts input and output **deflated shadow prices**.

Warning

The paasche() function will not work with unbalanced data.

Note

All output-oriented efficiency scores are computed *a la* Shephard, while all input-oriented efficiency scores are computed *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 and profitability in levels and components.
See Changes to retrieve a data frame with Paasche productivity and profitability change indices and components.
See Shadowp to retrieve deflated input and output shadow prices.
See also fisher and laspeyres for computations with alternative indices.

Examples

```
## Paasche profitability and productivity levels and changes' 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)
```

Shadowp *Shadow prices used in productivity and profitability computations*

Description

From any object created by either fareprim, fisher, laspeyres, lowe, or paasche function, this function extracts the deflated cost input shadow prices along with the deflated revenue output shadow prices used in productivity and profitability computations.

Arguments

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

Details

For all the price-based indices (i.e. Fisher, Lowe, Laspeyres and Paasche), deflated input and output shadow prices **for each observation** are return.

For the Färe-Primont index, the deflated input and output shadow price of the representative observation is returned.

- 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

The function returns a data frame containing all the elements and observations included in the "Shadowp" constituent element of object.

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

See Also

For details and information on returned values, see fareprim, fisher, laspeyres, lowe, or paasche.
See also:
- Changes for productivity and profitability change indices; and
- Levels for productivity and profitability levels.

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 <- Shadowp(FISHER)
    head(Fisher.shadowprices)

## End(Not run)
```

usagri *Price indices and implicit quantities of USA farm outputs and inputs by State, 1995-2004*

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

Further details 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