Package 'productivity'

September 8, 2026

Version 1.1.2

Date 2026-09-08

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.
The package contains the multiplicatively complete Färe-Primont, Fisher, Hicks-Moorsteen, Laspeyres, Lowe, and Paasche indices, as well as the classic Malmquist productivity index.
Färe-Primont and Lowe indices verify the transitivity property and can therefore be used for multilateral or multitemporal comparison.
Fisher, Hicks-Moorsteen, 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 changes.
In the use of Malmquist productivity index, the technological change index can be further decomposed into bias technological change components.
The package also allows to prohibit technological regression (negative technological change). In the case of the Fisher, Hicks-Moorsteen, Laspeyres, Paasche and the transitive Färe-Primont and Lowe indices, it is furthermore possible to rule out technological change.
Deflated shadow prices can also be obtained. Besides, the package allows parallel computing as an option, depending on the user's computer configuration.
All computations are carried out with the nonparametric Data Envelopment Analysis (DEA), and several assumptions regarding returns to scale are available.
All DEA linear programs are implemented using 'lp_solve'.

Depends R (>= 2.15.3)

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

Imports lpSolveAPI, doParallel, foreach, iterators, methods

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 2026-09-08 12:30:18 UTC

Contents

Changes . . . 2
fareprim . . . 4
fisher . . . 8
hicksmoorsteen . . . 12
laspeyres . . . 16
Levels . . . 20
lowe . . . 22
malm . . . 26
paasche . . . 29
Shadowp . . . 33
usagri . . . 34

Index **36**

`Changes` *Productivity and profitability change indices*

Description

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

Usage

```
Changes(object, ...)
```

Arguments

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

Details

- An object of class `'FarePrimont'` is a result of a call to `fareprim`.
- An object of class `'Fisher'` is a result of a call to `fisher`.
- An object of class `'HicksMoorsteen'` is a result of a call to `hicksmoorsteen`.
- An object of class `'Laspeyres'` is a result of a call to `laspeyres`.
- An object of class `'Lowe'` is a result of a call to `lowe`.
- An object of class `'Malmquist'` is a result of a call to `malm`.
- An object of class `'Paasche'` is a result of a call to `paasche`.

Value

- In the case of Färe-Primont, Fisher, Laspeyres, Lowe, Malmquist, and Paasche indices, the function returns a **data frame** containing all the elements and observations included in the `"Changes"` component of `object`.
- In the case of Hicks-Moorsteen index:
  - When `components = FALSE` (default) in the call to `hicksmoorsteen`, the function returns a **data frame** containing all the elements and observations included in the `"Changes"` component of the object of class `'HicksMoorsteen'`.
  - When `components = TRUE` in the call to `hicksmoorsteen`, the function returns a **list** of three data frames:
    - **HicksMoorsteen:** A data frame containing all the elements and observations related to `"Changes"` component of the Hicks-Moorsteen index.
    - **MalmquistHS:** A data frame containing all the elements and observations related to `"Changes"` component of the Malmquist-*hs* index.
    - **MalmquistIT:** A data frame containing all the elements and observations related to `"Changes"` component of the Malmquist-*it* index.

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

See Also

For details and information on returned values, see `fareprim`, `fisher`, `hicksmoorsteen`, `laspeyres`, `lowe`, `malm`, or `paasche`.

See also:
- `Levels` for productivity and profitability levels; and
- `Shadowp` for shadow prices.

Examples

```
## Not run:
  FAREPRIM <- fareprim(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"), w.vars = c("p.capital", "p.land", "p.labor", "p.materials"),
  p.vars = c("p.livestock", "p.crop", "p.other"))
  Fareprim.change <- Changes(FAREPRIM)
  head(Fareprim.change)

## End(Not run)
```

`fareprim` *Färe-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 can also be 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 = TRUE, by.id = NULL, by.year = NULL, shadow = FALSE)

## 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, and in this case productivity only is measured.
`p.vars`	Output price variables (Optional). Can be a vector of text strings or integers. `NULL` by default, and in this case productivity only is measured.
`tech.change`	Logical. If `TRUE` (default), the model allows for technological change. If `FALSE`, technological change is prohibited. See also the `Details` section.
`tech.reg`	Logical. If `TRUE` (default), the model allows for negative technological change (i.e. technological regress). If `FALSE`, only positive technological change (i.e. technological progress) is allowed. 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 "in-out", 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. When the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).
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. If TRUE (default), input and output quantities are rescaled. If FALSE, a warning message is displayed when very large (>1e5) and/or very small (<1e-4) values are present in the input and output quantity variables. 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.
shadow	Logical. Default is FALSE (no shadow prices are returned). When set to TRUE, the deflated input and output shadow prices of the 'representative observation' (i.e. the sample means of quantities and prices) used to compute the Färe-Primont index are returned.
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

When tech.change is set to FALSE, this overrides the effect of tech.reg.

Setting scaled = FALSE (no rescaling of data) may lead to numerical problems in solving LP problems while optimizing DEA models. In extreme cases it may also prevent models from being optimized.

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 (when price information is specified) measures in levels and changes, as well as a summary shadow prices (if `shadow = TRUE`), is printed.

This list contains the following items:

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

`REV`	Revenues *(when* `w.vars` *and* `p.vars` *are specified)*
`COST`	Costs *(when* `w.vars` *and* `p.vars` *are specified)*
`PROF`	Profitability *(when* `w.vars` *and* `p.vars` *are specified)*
`P`	Aggregated output prices *(when* `w.vars` *and* `p.vars` *are specified)*
`W`	Aggregated input prices *(when* `w.vars` *and* `p.vars` *are specified)*
`TT`	Terms of trade (i.e. `P/W`) *(when* `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 *(*`orientation = "out"`*)*
`OSE`	Output-oriented scale efficiency score *(*`orientation = "out"`*)*
`OME`	Output-oriented mix efficiency score *(*`orientation = "out"`*)*
`ROSE`	Residual output-oriented scale efficiency score *(*`orientation = "out"`*)*
`OSME`	Output-oriented scale-mix efficiency score *(*`orientation = "out"`*)*
`ITE`	Input-oriented technical efficiency score *(*`orientation = "in"`*)*
`ISE`	Input-oriented scale efficiency score *(*`orientation = "in"`*)*
`IME`	Input-oriented mix efficiency score *(*`orientation = "in"`*)*
`RISE`	Residual input-oriented scale efficiency score *(*`orientation = "in"`*)*
`ISME`	Input-oriented scale-mix efficiency score *(*`orientation = "in"`*)*
`OTE.ITE`	Geometric mean of `OTE` and `ITE` *(*`orientation = "in-out"`*)*
`OSE.ISE`	Geometric mean of `OSE` and `ISE` *(*`orientation = "in-out"`*)*
`OME.IME`	Geometric mean of `OME` and `IME` *(*`orientation = "in-out"`*)*
`ROSE.RISE`	Geometric mean of `ROSE` and `RISE` *(*`orientation = "in-out"`*)*
`OSME.ISME`	Geometric mean of `OSME` and `ISME` *(*`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` Returned only if `shadow = TRUE`. It contains the deflated cost input (`x.vars`) shadow prices and the deflated revenue output (`y.vars`) shadow prices of the 'representative observation' used to compute the Färe-Primont index.

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

- `Levels` function extracts individual productivity and profitability **levels**;
- `Changes` function extracts individual productivity and profitability **change indices**; and
- If `shadow = TRUE`, the `Shadowp` function extracts input and output **deflated shadow prices**.

Warning

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

The Färe-Primont index may be sensitive to the rescaling.

For extreme efficient observations, the problem of multiple solutions may arise and the values of shadow prices may differ depending on the linear programming solver used (here **lpSolveAPI**).

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. URL: https://espace.library.uq.edu.au/view/UQ:369053

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: https://espace.library.uq.edu.au/view/UQ:368915

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. doi:10.1093/ajae/aas023

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, provided that `shadow = TRUE`.

See also `lowe` for computations with an alternative transitive index.

Examples

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

## End(Not run)
## Färe-Primont productivity and profitability, with price information
```

```
## Not run:
 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), 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.

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

Deflated shadow prices of inputs and outputs can also be 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 = TRUE,
  shadow = 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. If `FALSE`, technological change is prohibited. See also the `Details` section.
`tech.reg`	Logical. If `TRUE` (default), the model allows for negative technological change (i.e. technological regress). If `FALSE`, only positive technological change (i.e. technological progress) is allowed. 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 "in-out", 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. When the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).
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. If TRUE (default), input and output quantities are rescaled. If FALSE, a warning message is displayed when very large (>1e5) and/or very small (<1e-4) values are present in the input and output quantity variables. See also the Details section.
shadow	Logical. Default is FALSE (no shadow prices are returned). When set to TRUE, input and output shadow prices are returned. These shadow prices are informative only and may be subject to the linear programming solver used.
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

When tech.change is set to FALSE, this overrides the effect of tech.reg.

Setting scaled = FALSE (no rescaling of data) may lead to numerical problems in solving LP problems while optimizing DEA models. In extreme cases it may also prevent models from being optimized.

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 shadow prices (if shadow = TRUE), 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 *(*orientation = "out"*)*
OSE	Output-oriented scale efficiency score *(*orientation = "out"*)*
RAE	Revenue allocative efficiency *(*orientation = "out"*)* *(equivalent to output-oriented mix efficiency score)*
ROSE	Residual output-oriented scale efficiency score *(*orientation = "out"*)*
OSME	Output-oriented scale-mix efficiency score *(*orientation = "out"*)*
ITE	Input-oriented technical efficiency score *(*orientation = "in"*)*
ISE	Input-oriented scale efficiency score *(*orientation = "in"*)*
CAE	Cost allocative efficiency *(*orientation = "in"*)* *(equivalent to input-oriented mix efficiency score)*
RISE	Residual input-oriented scale efficiency score *(*orientation = "in"*)*
ISME	Input-oriented scale-mix efficiency score *(*orientation = "in"*)*
OTE.ITE	Geometric mean of OTE and ITE *(*orientation = "in-out"*)*
OSE.ISE	Geometric mean of OSE and ISE *(*orientation = "in-out"*)*
RAE.CAE	Geometric mean of RAE and CAE *(*orientation = "in-out"*)*
ROSE.RISE	Geometric mean of ROSE and RISE *(*orientation = "in-out"*)*
OSME.ISME	Geometric mean of OSME and ISME *(*orientation = "in-out"*)*
RME	Residual mix efficiency score
RE	Revenue efficiency *(*orientation = "out"*)*
CE	Cost efficiency *(*orientation = "in"*)*
RE.CE	Geometric mean of RE and CE *(*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	Returned only if shadow = TRUE. It contains the deflated cost input (x.vars) shadow prices and the deflated revenue output (y.vars) shadow prices.

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

- Levels function extracts individual productivity and profitability **levels**;
- Changes function extracts individual productivity and profitability **change indices**; and
- If shadow = TRUE, the Shadowp function extracts individual input and output **deflated shadow prices**.

Warning

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

The Fisher index may be sensitive to the rescaling.

For extreme efficient observations, the problem of multiple solutions may arise and the values of shadow prices may differ depending on the linear programming solver used (here **lpSolveAPI**).

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. doi:10.1007/BF00158354

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: https://espace.library.uq.edu.au/view/UQ:368915

See Also

See `Levels` to retrieve a data frame with individual Fisher productivity and profitability in levels and components.
See `Changes` to retrieve a data frame with individual Fisher productivity and profitability changes and components.
See `Shadowp` to retrieve individual deflated input and output shadow prices, provided that `shadow = TRUE`.

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")
  Fisher.prod

## End(Not run)
```

`hicksmoorsteen` *Hicks-Moorsteen productivity and profitability index*

Description

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

The Hicks-Moorsteen index is the geometric average of its components, i.e. Malmquist-*hs* and Malmquist-*it* indices.

Deflated shadow prices of inputs and outputs used to compute Malmquist-*hs* and Malmquist-*it* indices can also be returned.

Usage

```
hicksmoorsteen(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 = TRUE, components = FALSE)

## S3 method for class 'HicksMoorsteen'
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, and in this case productivity only is measured.
`p.vars`	Output price variables (Optional). Can be a vector of text strings or integers. `NULL` by default, and in this case productivity only is measured.
`tech.change`	Logical. If `TRUE` (default), the model allows for technological change. If `FALSE`, technological change is prohibited. See also the `Details` section.
`tech.reg`	Logical. If `TRUE` (default), the model allows for negative technological change (i.e. technological regress). If `FALSE`, only positive technological change (i.e. technological progress) is allowed. 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 "in-out", 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. When the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).
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. If TRUE (default), input and output quantities are rescaled. If FALSE, a warning message is displayed when very large (>1e5) and/or very small (<1e-4) values are present in the input and output quantity variables. See also the Details section.
components	Logical. Default is FALSE (only Hicks-Moorsteen indices are returned). When set to TRUE, the components Malmquist-*hs* and Malmquist-*it* indices are also returned (in terms of levels, changes, along with shadow prices used to compute Malmquist-*hs* and Malmquist-*it*).
x	An object of class 'HicksMoorsteen'.
digits	The minimum number of significant digits to be printed in values. Default = max(3, getOption("digits") - 3).
...	Currently not used.

Details

The Hicks-Moorsteen index is the geometric average of Malmquist-*hs* and Malmquist-*it* indices. For a firm *i* Malmquist-*it* computes the productivity index based on the reference year *t*. For a firm *h*, Malmquist-*hs* computes the productivity index based on the reference year *s* (i.e. *t-1*). Therefore, the Malmquist-*it* index uses the current period shadow prices as aggregators, while the Malmquist-*hs* index uses the previous period shadow prices as aggregators.

When tech.change is set to FALSE, this overrides the effect of tech.reg.

Setting scaled = FALSE (no rescaling of data) may lead to numerical problems in solving LP problems while optimizing DEA models. In extreme cases it may also prevent models from being optimized.

The Hicks-Moorsteen 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

hicksmoorsteen() returns a list of class 'HicksMoorsteen' for which a summary of productivity and profitability (when price information is specified) measures in levels and changes is printed.

This list contains the following elements:

– **HicksMoorsteen**, containing levels and changes related to Hick-Moorsteen index per-se, with:

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

`REV`	Revenues *(when* `w.vars` *and* `p.vars` *are specified)*
`COST`	Costs *(when* `w.vars` *and* `p.vars` *are specified)*
`PROF`	Profitability (*when* `w.vars` *and* `p.vars` *are specified)*
`P`	Aggregated output prices *(when* `w.vars` *and* `p.vars` *are specified)*
`W`	Aggregated input prices *(when* `w.vars` *and* `p.vars` *are specified)*
`TT`	Terms of trade (i.e. `P/W`) *(when* `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 *(*`orientation = "out"`*)*
`OSE`	Output-oriented scale efficiency score *(*`orientation = "out"`*)*
`OME`	Output-oriented mix efficiency score *(*`orientation = "out"`*)*
`ROSE`	Residual output-oriented scale efficiency score *(*`orientation = "out"`*)*
`OSME`	Output-oriented scale-mix efficiency score *(*`orientation = "out"`*)*
`ITE`	Input-oriented technical efficiency score *(*`orientation = "in"`*)*
`ISE`	Input-oriented scale efficiency score *(*`orientation = "in"`*)*
`IME`	Input-oriented mix efficiency score *(*`orientation = "in"`*)*
`RISE`	Residual input-oriented scale efficiency score *(*`orientation = "in"`*)*
`ISME`	Input-oriented scale-mix efficiency score *(*`orientation = "in"`*)*
`OTE.ITE`	Geometric mean of `OTE` and `ITE` *(*`orientation = "in-out"`*)*
`OSE.ISE`	Geometric mean of `OSE` and `ISE` *(*`orientation = "in-out"`*)*
`OME.IME`	Geometric mean of `OME` and `IME` *(*`orientation = "in-out"`*)*
`ROSE.RISE`	Geometric mean of `ROSE` and `RISE` *(*`orientation = "in-out"`*)*
`OSME.ISME`	Geometric mean of `OSME` and `ISME` *(*`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.).

– **MalmquistHS**, only returned when `components = TRUE` and accessible using `Levels`, `Changes`, and `Shadowp`, containing levels, changes, and shadow prices related to Malmquist-*hs* index, with:

`Levels`	Several elements are provided, depending on the `orientation` specified.
`Changes`	Change indices of the different elements of `Levels`.
`Shadowp`	For each observation, input (`x.vars`) and output (`y.vars`) deflated shadow prices used to compute Malmquist-*hs* index are returned.

– MalmquistIT, only returned when `components = TRUE` and accessible using `Levels`, `Changes`, and `Shadowp`, containing levels, changes, and shadow prices related to Malmquist-*it* index, with:

`Levels`	Several elements are provided, depending on the `orientation` specified.
`Changes`	Change indices of the different elements of `Levels` are provided.

`Shadowp` For each observation, input (`x.vars`) and output (`y.vars`) deflated shadow prices used to compute Malmquist-*it* index are returned.

From an object of class `'HicksMoorsteen'` obtained from `hicksmoorsteen()`, the

- `Levels` function extracts individual Hicks-Moorsteen productivity and profitability **levels**;
- `Changes` function extracts individual Hicks-Moorsteen productivity and profitability **change indices**; and
- `Shadowp` function extracts individual input and output **deflated shadow prices** of Malmquist-*hs* and Malmquist-*it* indices, when `components = TRUE`.

Warning

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

The Hicks-Moorsteen index may be sensitive to the rescaling.

The productivity levels are obtained using shadow prices computed using dual (multipliers) DEA models. However, for extreme efficient observations the issue of multiple solutions may arise and the values of shadow prices may differ depending on the linear programming solver used (here **lpSolveAPI**).

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

Briec W., and Kerstens K. (2011). The Hicks-Moorsteen Productivity Index Satisfies the Determinateness Axiom. *The Manchester School*, **79**(4), 765–775. doi:10.1111/j.14679957.2010.02169.x

Caves D.W., Christensen L.R., and Diewert W.E.(1982). The Economic Theory of Index Numbers and the Measurement of Input, Output, and Productivity. *Econometrica*, **50**(6), 1393–1414. URL: http://www.jstor.org/stable/1913388

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. URL: https://espace.library.uq.edu.au/view/UQ:369053

O'Donnell C.J. (2010). Measuring and decomposing agricultural productivity and profitability change. *Australian Journal of Agricultural and Resource Economics*, **54**(4), 527–560. doi:10.1111/j.14678489.2010.00512.x

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: https://espace.library.uq.edu.au/view/UQ:368915

See Also

See Levels to retrieve Hicks-Moorsteen (along with Malmquist-*hs* and Malmquist-*it*) productivity and profitability in levels and components.
See Changes to retrieve Hicks-Moorsteen (along with Malmquist-*hs* and Malmquist-*it*) productivity and profitability changes and components.
See Shadowp to retrieve deflated input and output shadow prices of Malmquist-*hs* and Malmquist-*it*.

Examples

```
## Hicks-Moorsteen productivity, without price information
## Not run:
 Hicks1 <- hicksmoorsteen(data = usagri, id.var = "States", time.var = "Years", x.vars = c(7:10),
  y.vars = c(4:6), rts = "crs", orientation = "in")
  Hicks1

## End(Not run)
## Hicks-Moorsteen productivity and profitability, with price information
## Not run:
Hicks2 <- hicksmoorsteen(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))
  Hicks2

## 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.

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

Deflated shadow prices of inputs and outputs can also be 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 = TRUE,
  shadow = 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. If FALSE, technological change is prohibited. See also the Details section.
tech.reg	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). If FALSE, only positive technological change (i.e. technological progress) is allowed. 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 "in-out", 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. When the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).
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. If TRUE (default), input and output quantities are rescaled. If FALSE, a warning message is displayed when very large (>1e5) and/or very small (<1e-4) values are present in the input and output quantity variables. See also the Details section.
shadow	Logical. Default is FALSE (no shadow prices are returned). When set to TRUE, input and output shadow prices are returned. These shadow prices are informative only and may be subject to the linear programming solver used.
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

When tech.change is set to FALSE, this overrides the effect of tech.reg.

Setting scaled = FALSE (no rescaling of data) may lead to numerical problems in solving LP problems while optimizing DEA models. In extreme cases it may also prevent models from being optimized.

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 shadow prices (if shadow = TRUE), 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 *(*orientation = "out"*)*
OSE	Output-oriented scale efficiency score *(*orientation = "out"*)*
RAE	Revenue allocative efficiency *(*orientation = "out"*)* *(equivalent to output-oriented mix efficiency score)*
ROSE	Residual output-oriented scale efficiency score *(*orientation = "out"*)*
OSME	Output-oriented scale-mix efficiency score *(*orientation = "out"*)*
ITE	Input-oriented technical efficiency score *(*orientation = "in"*)*
ISE	Input-oriented scale efficiency score *(*orientation = "in"*)*
CAE	Cost allocative efficiency *(*orientation = "in"*)* *(equivalent to input-oriented mix efficiency score)*
RISE	Residual input-oriented scale efficiency score *(*orientation = "in"*)*
ISME	Input-oriented scale-mix efficiency score *(*orientation = "in"*)*
OTE.ITE	Geometric mean of OTE and ITE *(*orientation = "in-out"*)*
OSE.ISE	Geometric mean of OSE and ISE *(*orientation = "in-out"*)*
RAE.CAE	Geometric mean of RAE and CAE *(*orientation = "in-out"*)*
ROSE.RISE	Geometric mean of ROSE and RISE *(*orientation = "in-out"*)*
OSME.ISME	Geometric mean of OSME and ISME *(*orientation = "in-out"*)*
RME	Residual mix efficiency score

RE	Revenue efficiency *(*orientation = "out"*)*
CE	Cost efficiency *(*orientation = "in"*)*
RE.CE	Geometric mean of RE and CE *(*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 — Returned only if shadow = TRUE. It contains the deflated cost input (x.vars) shadow prices and the deflated revenue output (y.vars) shadow prices.

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

- Levels function extracts individual productivity and profitability **levels**;
- Changes function extracts individual productivity and profitability **change indices**; and
- If shadow = TRUE, the Shadowp function extracts individual input and output **deflated shadow prices**.

Warning

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

The Laspeyres index may be sensitive to the rescaling.

For extreme efficient observations, the problem of multiple solutions may arise and the values of shadow prices may differ depending on the linear programming solver used (here **lpSolveAPI**).

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: https://espace.library.uq.edu.au/view/UQ:368915

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, provided that shadow = TRUE.

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")
  Laspeyres.prod

## End(Not run)
```

Levels *Productivity and profitability levels*

Description

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

Usage

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

Arguments

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

Details

- An object of class 'FarePrimont' is a result of a call to fareprim.
- An object of class 'Fisher' is a result of a call to fisher.
- An object of class 'HicksMoorsteen' is a result of a call to hicksmoorsteen.
- An object of class 'Laspeyres' is a result of a call to laspeyres.
- An object of class 'Lowe' is a result of a call to lowe.
- An object of class 'Malmquist' is a result of a call to malm.
- An object of class 'Paasche' is a result of a call to paasche.

Value

- In the case of Färe-Primont, Fisher, Laspeyres, Lowe, Malmquist, and Paasche indices, the function returns a **data frame** containing all the elements and observations included in the `"Levels"` component of `object`.
- In the case of Hicks-Moorsteen index:
  - When `components = FALSE` (default) in the call to `hicksmoorsteen`, the function returns a **data frame** containing all the elements and observations included in the `"Levels"` component of the object of class `'HicksMoorsteen'`.
  - When `components = TRUE` in the call to `hicksmoorsteen`, the function returns a **list** of three data frames:
    * **HicksMoorsteen:** A data frame containing all the elements and observations related to `"Levels"` component of the Hicks-Moorsteen index.
    * **MalmquistHS:** A data frame containing all the elements and observations related to `"Levels"` component of the Malmquist-*hs* index.
    * **MalmquistIT:** A data frame containing all the elements and observations related to `"Levels"` component of the Malmquist-*it* index.

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

See Also

For details and information on returned values, see `fareprim`, `fisher`, `hicksmoorsteen`, `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))
  Lowe.levels <- Levels(LOWE)
  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 can also be 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 = TRUE,
  by.id = NULL, by.year = NULL, shadow = FALSE)

## 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. If FALSE, technological change is prohibited. See also the Details section.
tech.reg	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). If FALSE, only positive technological change (i.e. technological progress) is allowed. 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 "in-out", 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. When the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).
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. If TRUE (default), input and output quantities are rescaled. If FALSE, a warning message is displayed when very large (>1e5) and/or very small (<1e-4) values are present in the input and output quantity variables. 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.
shadow	Logical. Default is FALSE (no shadow prices are returned). When set to TRUE, input and output shadow prices are returned. These shadow prices are informative only and may be subject to the linear programming solver used.
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

When tech.change is set to FALSE, this overrides the effect of tech.reg.

Setting scaled = FALSE (no rescaling of data) may lead to numerical problems in solving LP problems while optimizing DEA models. In extreme cases it may also prevent models from being optimized.

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.

The Lowe index is also a fixed-weights-based TFP index as the Färe-Primont. The Lowe index uses the average observed input and output prices as aggregators.

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 shadow prices (if shadow = TRUE), 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 *(`orientation = "out"`)*
`OSE`	Output-oriented scale efficiency score *(`orientation = "out"`)*
`OME`	Output-oriented mix efficiency score *(`orientation = "out"`)*
`ROSE`	Residual output-oriented scale efficiency score *(`orientation = "out"`)*
`OSME`	Output-oriented scale-mix efficiency score *(`orientation = "out"`)*
`ITE`	Input-oriented technical efficiency score *(`orientation = "in"`)*
`ISE`	Input-oriented scale efficiency score *(`orientation = "in"`)*
`IME`	Input-oriented mix efficiency score *(`orientation = "in"`)*
`RISE`	Residual input-oriented scale efficiency score *(`orientation = "in"`)*
`ISME`	Input-oriented scale-mix efficiency score *(`orientation = "in"`)*
`OTE.ITE`	Geometric mean of `OTE` and `ITE` *(`orientation = "in-out"`)*
`OSE.ISE`	Geometric mean of `OSE` and `ISE` *(`orientation = "in-out"`)*
`OME.IME`	Geometric mean of `OME` and `IME` *(`orientation = "in-out"`)*
`ROSE.RISE`	Geometric mean of `ROSE` and `RISE` *(`orientation = "in-out"`)*
`OSME.ISME`	Geometric mean of `OSME` and `ISME` *(`orientation = "in-out"`)*
`RME`	Residual mix efficiency score
`RE`	Revenue efficiency *(`orientation = "out"`)*
`CE`	Cost efficiency *(`orientation = "in"`)*
`RE.CE`	Geometric mean of `RE` and `CE` *(`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.).

`Shadowp` Returned only if `shadow = TRUE`. It contains the deflated cost input (`x.vars`) shadow prices and the deflated revenue output (`y.vars`) shadow prices.

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

- `Levels` function extracts individual productivity and profitability **levels**;
- `Changes` function extracts individual productivity and profitability **change indices**; and
- If `shadow = TRUE`, the `Shadowp` function extracts individual input and output **deflated shadow prices**.

Warning

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

The Lowe index may be sensitive to the rescaling.

For extreme efficient observations, the problem of multiple solutions may arise and the values of shadow prices may differ depending on the linear programming solver used (here **lpSolveAPI**).

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. URL: https://espace.library.uq.edu.au/view/UQ:369053

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: https://espace.library.uq.edu.au/view/UQ:368915

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. doi:10.1093/ajae/aas023

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, provided that `shadow = TRUE`.

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", 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 = TRUE)

## 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). If FALSE, only positive technological change (i.e. technological progress) is allowed. 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. When the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).
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. If TRUE (default), input and output quantities are rescaled. If FALSE, a warning message is displayed when very large (>1e5) and/or very small (<1e-4) values are present in the input and output quantity variables.
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

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 (scaled).

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.

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, or c111i, c100i, c011i, c000i, c110i, c010i. When the returns to scale option (rts) is different from "crs", then v111o and v000o, or v111i and v000i (depending on the orientation) are returned in addition to the distance function estimated under constant returns to scale ("crs"). The prefix "c" stands for constant returns to scale ("crs") and "v" for all 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.

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 *(when* rts != "crs" *and* orientation = "out"*)*
out.scalech	Output scale efficiency change *(when* rts != "crs" *and* orientation = "out"*)*
pure.inp.effch	Pure input efficiency change

(when `rts != "crs"` *and* `orientation = "in"`*)*

`inp.scalech` Input scale efficiency change
(when `rts != "crs"` *and* `orientation = "in"`*)*

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`).
3. `pure.inp.effch` (Pure input efficiency change), and `inp.scalech` (Input scale efficiency change) are components of efficiency change (`effch`).

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 components.

Warning

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

Note

The Malmquist productivity index and components are computed such that both `orientation`'s results provide the same information: growth when index greater than one and decline when index 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 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")
  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.

The Paasche index uses current period prices as aggregators.

Deflated shadow prices of inputs and outputs can also be 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 = TRUE,
  shadow = 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. If FALSE, technological change is prohibited. See also the Details section.
tech.reg	Logical. If TRUE (default), the model allows for negative technological change (i.e. technological regress). If FALSE, only positive technological change (i.e. technological progress) is allowed. 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 "in-out", 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. When the sample size is small, it is recommended to keep the parallel option to its default value (FALSE).
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. If TRUE (default), input and output quantities are rescaled. If FALSE, a warning message is displayed when very large (>1e5) and/or very small (<1e-4) values are present in the input and output quantity variables. See also the Details section.
shadow	Logical. Default is FALSE (no shadow prices are returned). When set to TRUE, input and output shadow prices are returned. These shadow prices are informative only and may be subject to the linear programming solver used.
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

When tech.change is set to FALSE, this overrides the effect of tech.reg.

Setting scaled = FALSE (no rescaling of data) may lead to numerical problems in solving LP problems while optimizing DEA models. In extreme cases it may also prevent models from being optimized.

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 shadow prices (if shadow = TRUE), 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 *(*orientation = "out"*)*
OSE	Output-oriented scale efficiency score *(*orientation = "out"*)*
RAE	Revenue allocative efficiency *(*orientation = "out"*)* *(equivalent to output-oriented mix efficiency score)*
ROSE	Residual output-oriented scale efficiency score *(*orientation = "out"*)*
OSME	Output-oriented scale-mix efficiency score *(*orientation = "out"*)*
ITE	Input-oriented technical efficiency score *(*orientation = "in"*)*
ISE	Input-oriented scale efficiency score *(*orientation = "in"*)*
CAE	Cost allocative efficiency *(*orientation = "in"*)* *(equivalent to input-oriented mix efficiency score)*
RISE	Residual input-oriented scale efficiency score *(*orientation = "in"*)*
ISME	Input-oriented scale-mix efficiency score *(*orientation = "in"*)*
OTE.ITE	Geometric mean of OTE and ITE *(*orientation = "in-out"*)*
OSE.ISE	Geometric mean of OSE and ISE *(*orientation = "in-out"*)*
RAE.CAE	Geometric mean of RAE and CAE *(*orientation = "in-out"*)*
ROSE.RISE	Geometric mean of ROSE and RISE *(*orientation = "in-out"*)*
OSME.ISME	Geometric mean of OSME and ISME *(*orientation = "in-out"*)*
RME	Residual mix efficiency score
RE	Revenue efficiency *(*orientation = "out"*)*
CE	Cost efficiency *(*orientation = "in"*)*
RE.CE	Geometric mean of RE and CE *(*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	Returned only if shadow = TRUE. It contains the deflated cost input (x.vars) shadow prices and the deflated revenue output (y.vars) shadow prices.

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

- Levels function extracts individual productivity and profitability **levels**;
- Changes function extracts individual productivity and profitability **change indices**; and
- If shadow = TRUE, the Shadowp function extracts individual input and output **deflated shadow prices**.

Warning

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

The Paasche index may be sensitive to the rescaling.

For extreme efficient observations, the problem of multiple solutions may arise and the values of shadow prices may differ depending on the linear programming solver used (here **lpSolveAPI**).

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: https://espace.library.uq.edu.au/view/UQ:368915

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, provided that `shadow = TRUE`.

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")
  Paasche.prod

## End(Not run)
```

Shadowp *Shadow prices used in productivity and profitability computations*

Description

From any object created by either `fareprim`, `fisher`, `hicksmoorsteen`, `laspeyres`, `lowe`, or `paasche` function, and provided that the argument `shadow` (or `components` in the case of `hicksmoorsteen`) is set to `TRUE` in the function's call, this function extracts the deflated cost input and revenue output shadow prices.

Usage

```
Shadowp(object, ...)
```

Arguments

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

Details

Shadow prices are derived from dual input- and output-oriented DEA models.

In the case of Fisher, Hicks-Moorsteen, Laspeyres, Lowe and Paasche indices, deflated input and output shadow prices **for each observation** are returned (if the argument `shadow = TRUE` in the function's call).

In the case of Hicks-Moorsteen index, shadow prices used to compute Malmquist-*hs* and Malmquist-*it* **for each observation** are returned (if the argument `components = TRUE` in the function's call).

In the case of Färe-Primont index, the deflated input and output shadow prices **of the representative observation** (i.e. the sample means of quantities and prices) are returned (if the argument `shadow = TRUE` in the function's call).

- An object of class `'FarePrimont'` is a result of a call to `fareprim`.
- An object of class `'Fisher'` is a result of a call to `fisher`.
- An object of class `'HicksMoorsteen'` is a result of a call to `hicksmoorsteen`.
- An object of class `'Laspeyres'` is a result of a call to `laspeyres`.
- An object of class `'Lowe'` is a result of a call to `lowe`.
- An object of class `'Paasche'` is a result of a call to `paasche`.

Value

- In the case of Färe-Primont, Fisher, Laspeyres, Lowe and Paasche indices, the function returns a **data frame** containing all the elements and observations included in the `"Shadowp"` component of `object`.
- In the case of Hicks-Moorsteen index, the function returns a **list** of two data frames containing, for each observation, input and output deflated shadow prices related to Malmquist-*hs* and Malmquist-*it* indices (and therefore not Hicks-Moorsteen shadow prices per se).
  - **MalmquistHS:** A data frame containing, for each observation, input and output shadow prices of the Malmquist-*hs* index.
  - **MalmquistIT:** A data frame containing, for each observation, input and output shadow prices of the Malmquist-*it* index.

Warning

For extreme efficient observations, the problem of multiple solutions may arise and the values of shadow prices may differ depending on the linear programming solver used (here **lpSolveAPI**).

Author(s)

Yann Desjeux, K Hervé Dakpo, Laure Latruffe

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", shadow = 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:

- `p.livestock`: Livestock and animal products' relative price (reference is 1 = Alabama 1996).
- `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 (reference is 1 = Alabama 1996).
- `p.crop`: Crops' relative price (reference is 1 = Alabama 1996).
- `p.other`: Other farm-related productions' relative price (reference is 1 = Alabama 1996).
- `p.capital`: Capital services' relative price (reference is 1 = Alabama 1996).
- `p.land`: Land service flows' relative price (reference is 1 = Alabama 1996).
- `p.labor`: Labor services' relative price (reference is 1 = Alabama 1996).
- `p.materials`: Total intermediate inputs' relative price (reference is 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**(1], 164–179. doi:10.2307/1244458

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. doi:10.1111/j.00029092.2004.00683.x

Examples

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

Index

∗ **datasets**
usagri, 34
∗ **manip**
Changes, 2
Levels, 20
Shadowp, 33
∗ **models**
fareprim, 4
fisher, 8
hicksmoorsteen, 12
laspeyres, 16
lowe, 22
malm, 26
paasche, 29

Changes, 2, *6, 7*, *10, 11*, *14–16*, *19–21*, *24, 25*, *28*, *31, 32*, *34*

fareprim, *2, 3*, 4, *20, 21*, *25*, *33*
fisher, *2, 3*, 8, *20, 21*, *32, 33*

hicksmoorsteen, *2, 3*, 12, *20, 21*, *33*

laspeyres, *2, 3*, *11*, 16, *20, 21*, *32, 33*
Levels, *3*, *6, 7*, *10, 11*, *14–16*, *19, 20*, 20, *24, 25*, *28*, *31, 32*, *34*
lowe, *2, 3*, *7*, *20, 21*, 22, *33*

malm, *2, 3*, *20, 21*, 26

paasche, *2, 3*, *11*, *20, 21*, 29, *33*
print.FarePrimont *(*fareprim*)*, 4
print.Fisher *(*fisher*)*, 8
print.HicksMoorsteen *(*hicksmoorsteen*)*, 12
print.Laspeyres *(*laspeyres*)*, 16
print.Lowe *(*lowe*)*, 22
print.Malmquist *(*malm*)*, 26
print.Paasche *(*paasche*)*, 29

Shadowp, *3*, *6, 7*, *10, 11*, *14–16*, *19–21*, *24, 25*, *31, 32*, 33

usagri, 34