Package 'PriceIndices'

March 20, 2021

Title Calculating Bilateral and Multilateral Price Indexes

Version 0.0.4

Description Preparing a scanner data set for price dynamics calculations (data selecting, data classification, data matching, data filtering). Computing bilateral and multilateral indexes. For details on these methods see: de Haan and Krsinich (2017) <doi:10.1111/roiw.12304> and Diewert and Fox (2020) <doi:10.1080/07350015.2020.1816176>.

Depends R (>= 3.5.0)

Imports lubridate (>= 1.7.4), dplyr (>= 0.8.3), ggplot2 (>= 3.2.0), reshape, reclin, stringr, xgboost, caret, strex

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jacek Białek [aut, cre] (<https://orcid.org/0000-0002-0952-5327>)

Maintainer Jacek Białek <jacek.bialek@uni.lodz.pl>

Repository CRAN

Date/Publication 2021-03-20 20:00:02 UTC

R topics documented:

`agmean` *Calculating the bilateral AG Mean price index*

Description

This function returns a value (or vector of values) of the bilateral AG Mean price index.

Usage

```
agmean(data, start, end, sigma = 0.7, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
sigma	The elasticity of substitution parameter (as numeric)
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral AG Mean price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Lent J., & Dorfman,A. H. (2009). *Using a Weighted Average of Base Period Price Indexes to Approximate a Superlative Index.* Journal of Official Statistics, 25(1), 139-149.

Examples

```
agmean(sugar, start="2019-01", end="2020-01",sigma=0.5)
agmean(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

available — *Providing values from the indicated column that occur at least once in one of the compared periods or in a given time interval*

Description

The function returns all values from the indicated column (defined by the type parameter) which occur at least once in one of the compared periods or in a given time interval.

Usage

```
available(data, period1, period2, type = "prodID", interval = FALSE)
```

Arguments

data	The user's data frame. It must contain a column time (as Date in format: year-month-day,e.g. '2020-12-01') and also a column indicated by the type parameter.
period1	The first period (as character) limited to the year and month, e.g. "2019-03".
period2	The second period (as character) limited to the year and month, e.g. "2019-04".
type	This parameters defines the column which is used in the procedure. Possible values of the type parameter are: retID, prodID, codeIN, codeOUT or description.
interval	A logical parameter indicating whether the procedure is to work for the whole time period between period1 and period2 (then it is TRUE).

Value

The function returns all values from the indicated column (defined by the type parameter) which occur at least once in one of the compared periods or in a given time interval. Possible values of the type parameter are: retID, prodID, codeIN, codeOUT or description. If the interval parameter is set to FALSE, then the function compares only periods defined by period1 and period2. Otherwise the whole time period between period1 and period2 is considered.

Examples

```
available(milk, period1="2018-12", period2="2019-12", interval=TRUE)
available(milk, period1="2018-12", period2="2019-12", type="description")
```

---

banajree *Calculating the bilateral Banajree price index*

Description

This function returns a value (or vector of values) of the bilateral Banajree price index.

Usage

```
banajree(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Banajree price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function)..

References

Banajree, K. S. (1977). *On the factorial approach providing the true index of cost of living.* Gottingen : Vandenhoeck und Ruprecht.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
banajree(sugar, start="2018-12", end="2019-12")
banajree(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

bialek *Calculating the bilateral Bialek price index*

Description

This function returns a value (or vector of values) of the bilateral Bialek price index.

Usage

```
bialek(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

`interval` A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Bialek price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Von der Lippe, P. (2012). *Some short notes on the price index of Jacek Bialek*. Econometrics (Ekonometria). 1(35), 76-83.

Bialek, J. (2013). *Some Remarks on the Original Price Index Inspired by the Notes of Peter von der Lippe*. Econometrics (Ekonometria), 3(41), 40-54.

Bialek, J. (2014). *Simulation Study of an Original Price Index Formula*. Communications in Statistics - Simulation and Computation, 43(2), 285-297

Examples

```
bialek(sugar, start="2018-12", end="2019-12")
bialek(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

bmw *Calculating the unweighted BMW price index*

Description

This function returns a value (or vector of values) of the unweighted Balk-Mehrhoff-Walsh (BMW) price index.

Usage

```
bmw(data, start, end, interval = FALSE)
```

Arguments

`data` User's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.

start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the unweighted bilateral BMW price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Mehrhoff, J.(2007). *A linear approximation to the Jevons index*. In: Von der Lippe (2007): Index Theory and Price Statistics, Peter Lang: Berlin, Germany.

(2018). *Harmonised Index of Consumer Prices (HICP). Methodological Manual*. Publication Office of the European union, Luxembourg.

Examples

```
bmw(sugar, start="2018-12", end="2019-12")
bmw(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

carli *Calculating the unweighted Carli price index*

Description

This function returns a value (or vector of values) of the unweighted bilateral Carli price index.

Usage

```
carli(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric) and prodID (as numeric, factor or character). A column quantities (as positive numeric) is also needed because this function uses unit values as monthly prices.

`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the unweighted bilateral Carli price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Carli, G. (1804). *Del valore e della proporzione de'metalli monetati*. Scrittori Classici Italiani di Economia Politica, 13, 297-336.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
carli(sugar, start="2018-12", end="2019-12")
carli(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

ccdi *Calculating the multilateral GEKS price index based on the Tornqvist formula (typical notation: GEKS-T or CCDI)*

Description

This function returns a value of the multilateral CCDI price index, i.e. the GEKS price index based on the superlative Tornqvist index formula.

Usage

```
ccdi(data, start, end, wstart = start, window = 13)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`wstart`	The beginning of the time interval (which is used by multilateral methods) limited to the year and month, e.g. "2020-01".
`window`	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

Value

This function returns a value of the multilateral CCDI price index (to be more precise: the GEKS index based on the Tornqvist formula) which considers the time window defined by `wstart` and `window` parameters. It measures the price dynamics by comparing period `end` to period `start` (both `start` and `end` must be inside the considered time window). To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Caves, D.W., Christensen, L.R. and Diewert, W.E. (1982). *Multilateral comparisons of output, input, and productivity using superlative index numbers.* Economic Journal 92, 73-86.

Examples

```
ccdi(milk, start="2019-01", end="2019-08",window=10)
ccdi(milk, start="2018-12", end="2019-12")
```

---

ccdi_fbew *Extending the multilateral CCDI price index by using the FBEW method.*

Description

This function returns a value of the multilateral CCDI price index (GEKS based on the Tornqvist formula) extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
ccdi_fbew(data, start, end)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral CCDI price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method. The FBEW method uses a time window with a fixed base month every year (December). The window is enlarged every month with one month in order to include information from a new month. The full window length (13 months) is reached in December of each year. The function measures the price dynamics between periods end and start. The month of the start parameter must be December. If the distance between end and start exceeds 13 months, then internal Decembers play a role of chain-linking months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Caves, D.W., Christensen, L.R. and Diewert, W.E. (1982). *Multilateral comparisons of output, input, and productivity using superlative index numbers.* Economic Journal 92, 73-86.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
ccdi_fbew(milk, start="2018-12", end="2019-08")
```

---

ccdi_fbmw *Extending the multilateral CCDI price index by using the FBMW method.*

Description

This function returns a value of the multilateral CCDI price index (GEKS based on the Tornqvist formula) extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
ccdi_fbmw(data, start, end)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral CCDI price index extended by using the FBMW (Fixed Base Moving Window) method. It measures the price dynamics between periods end and start and it uses a 13-month time window with a fixed base month taken as year(end)-1. If the distance between end and start exceeds 13 months, then internal Decembers play a role of chain-linking months. The month of the start parameter must be December. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Caves, D.W., Christensen, L.R. and Diewert, W.E. (1982). *Multilateral comparisons of output, input, and productivity using superlative index numbers.* Economic Journal 92, 73-86.

Lamboray, C.(2017). *The Geary Khamis index and the Lehr index: how much do they differ?* Paper presented at the 15th Ottawa Group meeting, 10-12 May 2017, Elville am Rhein, Germany.

Examples

```
ccdi_fbmw(milk, start="2019-12", end="2020-04")
```

---

ccdi_splice — *Extending the multilateral CCDI price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral CCDI price index (GEKS based on the Tornqvist formula) extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see References).

Usage

```
ccdi_splice(
  data,
  start,
  end,
  window = 13,
  splice = "movement",
  interval = FALSE
)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
splice	A character string indicating the splicing method. Available options are: "movement", "window","half","mean", "window_published","half_published","mean_published".
interval	A logical value indicating whether the function is to provide the price index comparing the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base multilateral indices are to be presented (the fixed base month is defined by start).

Value

This function returns a value or values (depending on interval parameter) of the multilateral CCDI price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see References). The time window starts in start and should consist of at least two months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Caves, D.W., Christensen, L.R. and Diewert, W.E. (1982). *Multilateral comparisons of output, input, and productivity using superlative index numbers.* Economic Journal 92, 73-86.

de Haan, J., van der Grient, H.A. (2011). *Eliminating chain drift in price indexes based on scanner data.* Journal of Econometrics, 161, 36-46.

Krsinich, F. (2014). *The FEWS Index: Fixed Effects with a Window Splice? Non-Revisable Quality-Adjusted Price Indices with No Characteristic Information.* Paper presented at the UNECE-ILO Meeting of the Group of Experts on Consumer Price Indices, 2-4 May 2016, Geneva, Switzerland.

de Haan, J.(2015). *A Framework for Large Scale Use of Scanner Data in the Dutch CPI.* Paper presented at the 14th Ottawa Group meeting, Tokyo, Japan.

Diewert, W.E., and Fox, K.J. (2017). *Substitution Bias in Multilateral Methods for CPI Construction using Scanner Data.* Discussion paper 17-02, Vancouver School of Economics, The University of British Columbia, Vancouver, Canada.

Examples

```
ccdi_splice(milk, start="2018-12", end="2020-02",splice="half")
ccdi_splice(milk, start="2018-12", end="2020-02",window=10,interval=TRUE)
```

`chagmean` *Calculating the monthly chained AG Mean price index*

Description

This function returns a value (or vector of values) of the monthly chained AG Mean price index.

Usage

```
chagmean(data, start, end, sigma = 0.7, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`sigma`	The elasticity of substitution parameter (as numeric).
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained AG Mean price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Lent J., & Dorfman,A. H. (2009). *Using a Weighted Average of Base Period Price Indexes to Approximate a Superlative Index.* Journal of Official Statistics, 25(1), 139-149.

Examples

```
chagmean(sugar, start="2019-01", end="2019-04",sigma=0.5)
chagmean(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chbanajree` *Calculating the monthly chained Banajree price index*

Description

This function returns a value (or vector of values) of the monthly chained Banajree price index.

Usage

```
chbanajree(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Banajree price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Banajree, K. S. (1977). *On the factorial approach providing the true index of cost of living.* Gottingen : Vandenhoeck und Ruprecht.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chbanajree(sugar, start="2018-12", end="2019-04")
chbanajree(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chbialek *Calculating the monthly chained Bialek price index*

Description

This function returns a value (or vector of values) of the monthly chained Bialek price index.

Usage

```
chbialek(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Bialek price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Von der Lippe, P. (2012). *Some short notes on the price index of Jacek Bialek*. Econometrics (Ekonometria). 1(35), 76-83.

Bialek, J. (2013). *Some Remarks on the Original Price Index Inspired by the Notes of Peter von der Lippe*. Econometrics (Ekonometria), 3(41), 40-54.

Bialek, J. (2014). *Simulation Study of an Original Price Index Formula*. Communications in Statistics - Simulation and Computation, 43(2), 285-297

Examples

```
chbialek(sugar, start="2018-12", end="2019-04")
chbialek(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chbmw *Calculating the monthly chained BMW price index*

Description

This function returns a value (or vector of values) of the monthly chained Balk-Mehrhoff-Walsh (BMW) price index.

Usage

```
chbmw(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained BMW price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Mehrhoff, J.(2007). *A linear approximation to the Jevons index*. In: Von der Lippe (2007): Index Theory and Price Statistics, Peter Lang: Berlin, Germany.

(2018). *Harmonised Index of Consumer Prices (HICP). Methodological Manual*. Publication Office of the European union, Luxembourg.

Examples

```
chbmw(sugar, start="2018-12", end="2019-04")
chbmw(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chcarli` *Calculating the monthly chained Carli price index*

Description

This function returns a value (or vector of values) of the monthly chained Carli price index.

Usage

```
chcarli(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Carli price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Carli, G. (1804). *Del valore e della proporzione de'metalli monetati*. Scrittori Classici Italiani di Economia Politica, 13, 297-336.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chcarli(sugar, start="2018-12", end="2019-04")
chcarli(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chcswd` *Calculating the monthly chained CSWD price index*

Description

This function returns a value (or vector of values) of the monthly chained Carruthers-Sellwood-Ward-Dalen (CSWD) price index.

Usage

```
chcswd(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained CSWD price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Carruthers, A.G., Sellwood, D. J, Ward, P. W. (1980). *Recent developments in the retail price index*. Journal of the Royal Statistical Society. Series D (The Statisticain), 29(1), 1-32.

Dalen, J. (1992). *Recent developments in the retail price index*. The Statistician, 29(1), 1-32.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chcswd(sugar, start="2018-12", end="2019-04")
chcswd(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chdavies *Calculating the monthly chained Davies price index*

Description

This function returns a value (or vector of values) of the monthly chained Davies price index.

Usage

```
chdavies(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Davies price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Davies, G. R. (1924). *The Problem of a Standard Index Number Formula.* Journal of the American Statistical Association, 19 (146), 180-188.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chdavies(sugar, start="2018-12", end="2019-04")
chdavies(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chdrobisch` *Calculating the monthly chained Drobisch price index*

Description

This function returns a value (or vector of values) of the monthly chained Drobisch price index.

Usage

```
chdrobisch(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Drobisch price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Drobisch, M. W. (1871). *Ueber einige Einwurfe gegen die in diesen Jahrbuchern veroffentlichte neue Methode, die Veranderungen der Waarenpreise und des Geldwerths zu berechten*.Jahrbucher fur Nationalokonomie und Statistik, Vol. 16, s. 416-427.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chdrobisch(sugar, start="2018-12", end="2019-04")
chdrobisch(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chdutot` *Calculating the monthly chained Dutot price index*

Description

This function returns a value (or vector of values) of the monthly chained Dutot price index.

Usage

```
chdutot(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Dutot price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Dutot, C. F., (1738). *Reflexions Politiques sur les Finances et le Commerce*. The Hague: Les Freres Vaillant et Nicolas Prevost, Vol. 1.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chdutot(sugar, start="2018-12", end="2019-04")
chdutot(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chfisher` *Calculating the monthly chained Fisher price index*

Description

This function returns a value (or vector of values) of the monthly chained Fisher price index.

Usage

```
chfisher(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Fisher price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Fisher, I. (1922). *The Making of Index Numbers*. Boston: Houghton Mifflin.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chfisher(sugar, start="2018-12", end="2019-04")
chfisher(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chgeary_khamis *Calculating the monthly chained Geary-Khamis price index*

Description

This function returns a value (or vector of values) of the monthly chained Geary-Khamis price index.

Usage

```
chgeary_khamis(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Geary-Khamis price index depending on the `interval` parameter (please use `gk` function to calculate the multilateral Geary-Khamis price index). If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Geary, R. G. (1958). *A Note on Comparisons of Exchange Rates and Purchasing Power between Countries.* Journal of the Royal Statistical Society, Series A, 121, 97-99.

Khamis, S. H. (1970). *Properties and Conditions for the Existence of a new Type of Index Number.* Sankhya Series B32, 81-98.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chgeary_khamis(sugar, start="2018-12", end="2019-04")
chgeary_khamis(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chgeohybrid *Calculating the the monthly chained geohybrid price index*

Description

This function returns a value (or vector of values) of the monthly chained geohybrid price index. The geohybrid index was proposed by Bialek (2020) and it uses correlation coefficients between prices and quantities.

Usage

```
chgeohybrid(data, start, end, base = start, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`base`	The prior period used in the geohybrid price index formula (as character) limited to the year and month, e.g. "2020-01"
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained geohybrid price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Bialek, J. (2020). *Proposition of a Hybrid Price Index Formula for the Consumer Price Index Measurement*. Equilibrium. Quarterly Journal of Economics and Economic Policy, 15(4), 697-716.

Examples

```
chgeohybrid(sugar, start="2019-12", end="2020-05", base="2018-12")
chgeohybrid(milk, start="2019-12", end="2020-08", base="2018-12", interval=TRUE)
```

---

chgeolaspeyres *Calculating the monthly chained geo-logarithmic Laspeyres price index*

Description

This function returns a value (or vector of values) of the monthly chained geo-logarithmic Laspeyres price index.

Usage

```
chgeolaspeyres(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained geo-logarithmic Laspeyres price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chgeolaspeyres(sugar, start="2018-12", end="2019-04")
chgeolaspeyres(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chgeolowe` *Calculating the monthly chained geometric Lowe price index*

Description

This function returns a value (or vector of values) of the monthly chained geometric Lowe price index.

Usage

```
chgeolowe(data, start, end, base = start, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`base`	The prior period used in the geometric Lowe price index formula (as character) limited to the year and month, e.g. "2020-01"
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained geometric Lowe price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chgeolowe(sugar, start="2019-01", end="2019-04",base="2018-12")
chgeolowe(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chgeopaasche` *Calculating the monthly chained geo-logarithmic Paasche price index*

Description

This function returns a value (or vector of values) of the monthly chained geo-logarithmic Paasche price index.

Usage

```
chgeopaasche(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained geo-logarithmic Paasche price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chgeopaasche(sugar, start="2018-12", end="2019-04")
chgeopaasche(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chgeoyoung *Calculating the monthly chained geometric Young price index*

Description

This function returns a value (or vector of values) of the monthly chained geometric Young price index.

Usage

```
chgeoyoung(data, start, end, base = start, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`base`	The prior period used in the geometric Young price index formula (as character) limited to the year and month, e.g. "2020-01".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained geometric Young price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Young, A. H. (1992). *Alternative Measures of Change in Real Output and Prices.* Survey of Current Business, 72, 32-48.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chgeoyoung(sugar, start="2019-01", end="2019-04",base="2018-12")
chgeoyoung(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chharmonic *Calculating the monthly chained harmonic price index*

Description

This function returns a value (or vector of values) of the monthly chained "unnamed" harmonic price index.

Usage

```
chharmonic(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained harmonic price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chharmonic(sugar, start="2018-12", end="2019-04")
chharmonic(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chhybrid *Calculating the the monthly chained hybrid price index*

Description

This function returns a value (or vector of values) of the monthly chained hybrid price index. The hybrid index was proposed by Bialek (2020) and it uses correlation coefficients between prices and quantities.

Usage

```
chhybrid(data, start, end, base = start, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`base`	The prior period used in the hybrid price index formula (as character) limited to the year and month, e.g. "2020-01"
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained hybrid price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Bialek, J. (2020). *Proposition of a Hybrid Price Index Formula for the Consumer Price Index Measurement*. Equilibrium. Quarterly Journal of Economics and Economic Policy, 15(4), 697-716.

Examples

```
chhybrid(sugar, start="2019-12", end="2020-05", base="2018-12")
chhybrid(milk, start="2019-12", end="2020-08", base="2018-12", interval=TRUE)
```

`chjevons` *Calculating the monthly chained Jevons price index*

Description

This function returns a value (or vector of values) of the monthly chained Jevons price index

Usage

```
chjevons(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Jevons price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Jevons, W. S., (1865). *The variation of prices and the value of the currency since 1782*. J. Statist. Soc. Lond., 28, 294-320.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chjevons(sugar, start="2018-12", end="2019-04")
chjevons(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chlaspeyres *Calculating the monthly chained Laspeyres price index*

Description

This function returns a value (or vector of values) of the monthly chained Laspeyres price index.

Usage

```
chlaspeyres(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Laspeyres price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Laspeyres, E. (1871). *Die Berechnung einer mittleren Waarenpreissteigerung*. Jahrbucher fur Nationalokonomie und Statistik 16, 296-314.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chlaspeyres(sugar, start="2018-12", end="2019-04")
chlaspeyres(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chlehr *Calculating the monthly chained Lehr price index*

Description

This function returns a value (or vector of values) of the monthly chained Lehr price index.

Usage

```
chlehr(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Lehr price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Lehr, J. (1885). *Beitrage zur Statistik der Preise, insbesondere des Geldes und des Holzes.* J. D. Sauerlander, Frankfurt am Main.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chlehr(sugar, start="2018-12", end="2019-04")
chlehr(milk, start="2018-12", end="2020-01", TRUE)
```

---

chlloyd_moulton *Calculating the monthly chained Lloyd-Moulton price index*

Description

This function returns a value (or vector of values) of the monthly chained Lloyd-Moulton price index.

Usage

```
chlloyd_moulton(data, start, end, sigma = 0.7, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`sigma`	The elasticity of substitution parameter (as numeric).
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Lloyd-Moulton price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Lloyd, P. J. (1975). *Substitution Effects and Biases in Nontrue Price Indices.* The American Economic Review, 65, 301-313.

Moulton, B. R. (1996). *Constant Elasticity Cost-of-Living Index in Share-Relative Form.* Washington DC: U. S. Bureau of Labor Statistics, mimeograph

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chlloyd_moulton(sugar, start="2018-12", end="2019-04",sigma=0.9)
chlloyd_moulton(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chlowe *Calculating the monthly chained Lowe price index*

Description

This function returns a value (or vector of values) of the monthly chained Lowe price index.

Usage

```
chlowe(data, start, end, base = start, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`base`	The prior period used in the Lowe price index formula (as character) limited to the year and month, e.g. "2020-01".

interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Lowe price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chlowe(sugar, start="2019-01", end="2019-04",base="2018-12")
chlowe(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chmarshall_edgeworth *Calculating the monthly chained Marshall-Edgeworth price index*

Description

This function returns a value (or vector of values) of the monthly chained Marshall-Edgeworth price index.

Usage

```
chmarshall_edgeworth(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

`interval` A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Marshall-Edgeworth price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Marshall, A. (1887). *Remedies for Fluctuations of General Prices*. Contemporary Review, 51, 355-375.

Edgeworth, F. Y. (1887). *Measurement of Change in Value of Money I*. The first Memorandum presented to the British Association for the Advancement of Science; reprinted in Papers Relating to Political Economy, Vol. 1, New York, Burt Franklin, s. 1925.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chmarshall_edgeworth(sugar, start="2018-12", end="2019-04")
chmarshall_edgeworth(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chpaasche *Calculating the monthly chained Paasche price index*

Description

This function returns a value (or vector of values) of the monthly chained Paasche price index.

Usage

```
chpaasche(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Paasche price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Paasche, H. (1874). *Uber die Preisentwicklung der letzten Jahre nach den Hamburger Borsennotirungen*. Jahrbucher fur Nationalokonomie und Statistik, 12, 168-178.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chpaasche(sugar, start="2018-12", end="2019-04")
chpaasche(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chpalgrave *Calculating the monthly chained Palgrave price index*

Description

This function returns a value (or vector of values) of the monthly chained Palgrave price index.

Usage

```
chpalgrave(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Palgrave price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Palgrave, R. H. I. (1886). *Currency and Standard of Value in England, France and India and the Rates of Exchange Between these Countries.* Memorandum submitted to the Royal Commission on Depression of trade and Industry, Third Report, Appendix B, 312-390.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chpalgrave(sugar, start="2018-12", end="2019-04")
chpalgrave(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

chsato_vartia *Calculating the monthly chained Vartia-II (Sato-Vartia) price index*

Description

This function returns a value (or vector of values) of the monthly chained Vartia-II (Sato-Vartia) price index.

Usage

```
chsato_vartia(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Vartia-II (Sato-Vartia) price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Sato, K. (1976). *The Ideal Log-Change Index Number.* The Review of Economics and Statistics, 58(2), 223-228.

Vartia, Y. 0. (1976). *Ideal Log-Change Index Numbers* . Scandinavian Journal of Statistics 3(3), 121-126.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chsato_vartia(sugar, start="2018-12", end="2019-04")
chsato_vartia(milk, start="2018-12", end="2020-01", interval=TRUE)
```

chstuvel *Calculating the monthly chained Stuvel price index*

Description

This function returns a value (or vector of values) of the monthly chained Stuvel price index.

Usage

```
chstuvel(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Stuvel price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Stuvel, G. (1957). *A New Index Number Formula.* Econometrica, 25, 123-131.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chstuvel(sugar, start="2018-12", end="2019-04")
chstuvel(milk, start="2018-12", end="2020-01", interval=TRUE)
```

`chtornqvist` *Calculating the monthly chained Tornqvist price index*

Description

This function returns a value (or vector of values) of the monthly chained Tornqvist price index.

Usage

```
chtornqvist(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Tornqvist price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Tornqvist, L. (1936). *The Bank of Finland's Consumption Price Index*. Bank of Finland Monthly Bulletin 10, 1-8.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chtornqvist(sugar, start="2018-12", end="2019-04")
chtornqvist(milk, start="2018-12", end="2020-01", interval=TRUE)
```

chvartia *Calculating the monthly chained Vartia-I price index*

Description

This function returns a value (or vector of values) of the monthly chained Vartia-I price index.

Usage

```
chvartia(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Vartia-I price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Vartia, Y. 0. (1976). *Ideal Log-Change Index Numbers* . Scandinavian Journal of Statistics 3(3), 121-126.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chvartia(sugar, start="2018-12", end="2019-04")
chvartia(milk, start="2018-12", end="2020-01", interval=TRUE)
```

chwalsh *Calculating the monthly chained Walsh price index*

Description

This function returns a value (or vector of values) of the monthly chained Walsh price index.

Usage

```
chwalsh(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Walsh price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Walsh, C. M. (1901). *The Measurement of General Exchange Value*. The MacMillan Company, New York.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
chwalsh(sugar, start="2018-12", end="2019-04")
chwalsh(milk, start="2018-12", end="2020-01", interval=TRUE)
```

chyoung *Calculating the monthly chained Young price index*

Description

This function returns a value (or vector of values) of the monthly chained Young price index.

Usage

```
chyoung(data, start, end, base = start, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
base	The prior period used in the Young price index formula (as character) limited to the year and month, e.g. "2020-01".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the monthly chained Young price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Young, A. H. (1992). *Alternative Measures of Change in Real Output and Prices.* Survey of Current Business, 72, 32-48.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
chyoung(sugar, start="2019-01", end="2019-04",base="2018-12")
chyoung(milk, start="2018-12", end="2020-01", interval=TRUE)
```

coffee *Real data set on sold coffee*

Description

A collection of scanner data on the sale of coffee in one of Polish supermarkets in the period from December 2017 to October 2020

Usage

```
coffee
```

Format

A data frame with 6 columns and 42561 rows. The used variables are as follows:

`time` - Dates of transactions (Year-Month-Day)

`prices` - Prices of sold products [PLN]

`quantities` - Quantities of sold products [kg]

`prodID` - Unique product codes (data set contains 79 different prodIDs)

`retID` - Unique codes identifying outlets/retailer sale points (data set contains 20 different retIDs)

`description` Descriptions of sold coffee products (data set contains 3 different product descriptions)

compare_final_indices *A general function for graphical comparison of price indices*

Description

This function returns a figure with plots of previously calculated price indices.

Usage

```
compare_final_indices(finalindices = list(), names = c())
```

Arguments

`finalindices`	A list of data frames with previously calculated price indices. Each data frame must consist of two columns, i.e. the first column must includes dates limited to the year and month (e.g.: "2020-04") and the second column must indicate price index values for corresponding dates. The above-mentioned single data frame may be created manually in the previous step or it may be a result of functions: `price_index` or `final_index`. All considered data frames must have an identical number of rows.
`names`	A vector of character strings describing names of presented indices.

Value

This function returns a figure with plots of previously calculated price indices. It allows for graphical comparison of price index values which were previously calculated and now are provided as data frames (see `finalindices` parameter).

Examples

```
## Caluclating two indices by using two different package functions:
index1<-final_index(datasets=list(milk), start="2018-12",
end="2019-12",formula="walsh",interval=TRUE)
index2<-price_index(milk,start="2018-12", end="2019-12",
formula="geks",interval=TRUE)
## Graphical comparison of these two indices
compare_final_indices(finalindices=list(index1,index2),
names=c("Walsh index", "GEKS index"))
```

---

compare_indices *A function for graphical comparison of price indices*

Description

This function returns a figure with plots of selected price indices.

Usage

```
compare_indices(
  data,
  start,
  end,
  bilateral = c(),
  bindex = c(),
  base = c(),
  cesindex = c(),
  sigma = c(),
  simindex = c(),
  fbmulti = c(),
  fbwindow = c(),
  splicemulti = c(),
  splicewindow = c(),
  splice = c(),
  namebilateral = bilateral,
  namebindex = bindex,
  namecesindex = cesindex,
  namesimindex = simindex,
  namefbmulti = fbmulti,
  namesplicemulti = splicemulti
)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric) and prodID (as numeric, factor or character). A column quantities is also essential (as positive numeric) because unit values are calculated.
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
bilateral	A vector of character strings indicating bilateral price index formulas that are to be calculated. To see available options please use the link: PriceIndices.
bindex	A vector of character strings indicating Lowe- or Young-type price index formulas that are to be calculated. Available options are: young, geoyoung, lowe and geolowe.
base	The vector of prior periods used in the Young- or Lowe-type price indices. Each element of the vector (as character) must be limited to the year and month, e.g. "2020-01".
cesindex	A vector of character strings indicating CES price index formulas that are to be calculated. To see available options, please use the link: PriceIndices.
sigma	The vector of elasticity of substitution parameters used in the Lloyed-Moulton and AG Mean indices.
simindex	A vector of character strings indicating multilateral price index formulas based on relative price and quantity similarity that are to be calculated. To see available options, please use the link: PriceIndices.
fbmulti	A vector of character strings indicating multilateral price index formulas that are to be calculated. The available set of indices includes full-window multilateral indices or their FBEW and FBMW extensions.To see available options, please use the link: PriceIndices.
fbwindow	A vector of integers. Each element of the vector defines the length of the time window of the corresponding multilateral index (if it is selected by fbmulti).
splicemulti	The vector of character strings indicating multilateral price index formulas are to be extended by using splicing methods. To see available options please use the link: PriceIndices.
splicewindow	A vector of integers. Each element of the vector defines the length of the time window of the corresponding multilateral index (if it is selected by splicemulti).
splice	A vector of character strings. Each element of the vector indicates the splicing method is to be used for the corresponding multilateral index (if it is selected by splicemulti). Available values of vector elements are: "movement", "window","half","mean".
namebilateral	A vector of character strings describing names of bilateral price indices that are to be displayed. If this vector is empty, then default names are used.
namebindex	A vector of character strings describing names of Young- and/or Lowe-type price indices are to be displayed. If this vector is empty then default names are used.

namecesindex — A vector of character strings describing names of CES price indices that are to be displayed. If this vector is empty, then default names are used.

namesimindex — A vector of character strings describing names of multilateral price index formulas based on relative price and quantity similarity that are to be displayed. If this vector is empty, then default names are used.

namefbmulti — A vector of character strings describing names of full-window multilateralindices or their FBEW and FBMW extensions that are to be displayed. If this vector is empty, then default names are used.

namesplicemulti — A vector of character strings describing names of multilateral splice indices that are to be displayed. If this vector is empty, then default names are used.

Value

This function calculates selected bilateral or/and multilateral price indices and returns a figure with plots of these indices (together with dates on X-axis and a corresponding legend). The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use functions: final_index and compare_final_indices).

Examples

```
compare_indices(milk, start="2018-12", end="2019-04",
bilateral=c("jevons"),fbmulti=c("tpd"),fbwindow=c(6))
compare_indices(milk, start="2018-12", end="2019-05",
fbmulti=c("tpd","geks"),fbwindow=c(10,12))
```

---

cswd *Calculating the unweighted CSWD price index*

Description

This function returns a value (or vector of values) of the unweighted Carruthers-Sellwood-Ward-Dalen (CSWD) price index.

Usage

```
cswd(data, start, end, interval = FALSE)
```

Arguments

data — The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric) and prodID (as numeric, factor or character). A column quantities (as positive numeric) is also needed because this function uses unit values as monthly prices.

start — The base period (as character) limited to the year and month, e.g. "2020-03".

`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the unweighted bilateral CSWD price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Carruthers, A.G., Sellwood, D. J, Ward, P. W. (1980). *Recent developments in the retail price index*. Journal of the Royal Statistical Society. Series D (The Statisticain), 29(1), 1-32.

Dalen, J. (1992). *Recent developments in the retail price index*. The Statistician, 29(1), 1-32.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
cswd(sugar, start="2018-12", end="2019-12")
cswd(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

dataCOICOP *An artificial scanner data set containing 10 elementary product groups*

Description

A collection of artificial scanner data on the sale of tomatoes, fruit juices, low fat milk, full fat milk, sugar, chocolate, yoghurt, coffee, eggs and salt in the period from December 2018 to October 2020

Usage

```
dataCOICOP
```

Format

A data frame with 8 columns and 96600 rows (some rows are not complete). The used variables are as follows:

`time` - Dates of transactions (Year-Month-Day)

`prices` - Prices of sold products [EUR]

`quantities` - Quantities of sold products [unit defined in the 'unit' column]

`prodID` - Retailer product codes

`retID` - Unique codes identifying outlets/retailer sale points (10 retIDs)

`description` Descriptions of sold products

`unit` Sales units, e.g.: kg, ml, etc.

`coicop` Identifiers of COICOP groups (10 groups)

---

dataMATCH *Artificial data set on product sales*

Description

A collection of scanner data on the sale of sample artificial products.

Usage

```
dataMATCH
```

Format

A data frame with 7 columns and 30 rows. The used variables are as follows:

`time` - Dates of transactions (Year-Month-Day)

`prices` - Prices of sold products [PLN]

`quantities` - Quantities of sold products [liters]

`codeIN` - Unique internal (retailer) product codes (data set contains 5 different codeINs)

`codeOUT` - Unique external product codes (data set contains 5 different codeOUTs)

`retID` - Unique codes identifying outlets/retailer sale points (data set contains 2 different retIDs)

`description` Descriptions of sold products (data set contains 3 different product descriptions)

dataU — *Artificial, small scanner data set*

Description

A collection of artificial scanner data on 6 products sold in Dec, 2018. Product descriptions contain the information about their grammage and unit.

Usage

```
dataU
```

Format

A data frame with 5 columns and 6 rows. The used variables are as follows:

`time` - Dates of transactions (Year-Month-Day)

`prices` - Prices of sold products [PLN]

`quantities` - Quantities of sold products [item]

`prodID` - Unique product codes

`description` Descriptions of sold products (data set contains 6 different product descriptions)

data_check — *Checking the user's data frame*

Description

The function checks if the argument `data` points to a data frame which is suitable for further price index calculation. In particular, the function checks whether the indicated data frame contains the required columns and whether they are of the appropriate type (if not, the function returns FALSE and an appropriate comment).

Usage

```
data_check(data)
```

Arguments

`data`	Any R object but ultimately it is a data frame.

Value

The function returns TRUE if the data frame indicated by the `data` parameter is suitable for the calculation of price indices and returns FALSE otherwise.

Examples

```
data_check(milk)
data_check(iris)
```

`data_classifying` *Predicting product COICOP levels via the machine learning model*

Description

This function predicts product COICOP levels via the selected machine learning model.

Usage

```
data_classifying(model = list(), data)
```

Arguments

`model`	A list of 8 elements which identify the previously built machine learning model (the list is obtained via the `model_classification` function).
`data`	A data set for the model (products with their characteristics). This data set must contain all the columns which were used in the built model.

Value

This function provides the indicated data set with an additional column, i.e. `coicop_predicted`, which is obtained by using the selected machine learning model.

Examples

```
#Building the model
my.grid=list(eta=c(0.01,0.02,0.05),subsample=c(0.5))
data_train<-dplyr::filter(dataCOICOP,dataCOICOP$time<=as.Date("2020-08-01"))
data_test<-dplyr::filter(dataCOICOP,dataCOICOP$time>as.Date("2020-08-01"))
ML<-model_classification(data_train,data_test,grid=my.grid,
indicators=c("prodID","unit","description"),key_words=c("milk"),rounds=50)
#Data classification
data_classifying(ML, data_test)
```

`data_filtering` *Filtering a data set for further price index calculations*

Description

This function returns a filtered data set, i.e. a reduced user's data frame with the same columns and rows limited by a criterion defined by `filters`.

Usage

```
data_filtering(
  data,
  start,
  end,
  filters = c(),
  plimits = c(),
  pquantiles = c(),
  dplimits = c(),
  lambda = 1.25,
  interval = FALSE,
  retailers = FALSE
)
```

Arguments

`data`	The user's data frame with information about products to be filtered. It must contain columns: `time` (as Date in format: year-month-day, e.g. '2020-12-01'), `prices` (as positive numeric) and `quantities` (as positive numeric).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`filters`	A vector of filter names (options are: `extremeprices`, `dumpprices` and/or `lowsales`).
`plimits`	A two-dimensional vector of thresholds for minimum and maximum price change (it works if one of the chosen filters is `extremeprices` filter).
`pquantiles`	A two-dimensional vector of quantile levels for minimum and maximum price change (it works if one of the chosen filters is `extremeprices` filter).
`dplimits`	A two-dimensional vector of thresholds for maximum price drop and maximum ependiture drop (it works if one of the chosen filters is `dumpprices` filter).
`lambda`	The lambda parameter for `lowsales` filter (see `References` below).
`interval`	A logical value indicating whether the filtering process concerns only two periods defined by `start` and `end` parameters (then the `interval` is set to FALSE) or whether that function is to filter products sold during the whole time interval <start, end>, i.e. any subsequent months are compared.
`retailers`	A logical parameter indicating whether filtering should be done for each outlet (`retID`) separately. If it is set to FALSE, then there is no need to consider the `retID` column.

Value

This function returns a filtered data set (a reduced user's data frame). If the set of `filters` is empty, then the function returns the original data frame (defined by the `data` parameter) limited to considered months. On the other hand, if all filters are chosen, i.e. `filters=c(extremeprices,dumpprices,lowsales)`, then these filters work independently and a summary result is returned. Please note that both variants of `extremeprices` filter can be chosen at the same time, i.e. `plimits` and `pquantiles`, and they work also independently.

References

Van Loon, K., Roels, D. (2018) *Integrating big data in Belgian CPI*. Meeting of the Group of Experts on Consumer Price Indices, Geneva.

Examples

```
data_filtering(milk,start="2018-12",end="2019-03",
filters=c("extremeprices"),pquantiles=c(0.01,0.99),interval=TRUE)
data_filtering(milk,start="2018-12",end="2019-03",
filters=c("extremeprices","lowsales"), plimits=c(0.25,2))
```

---

data_matching *Matching products*

Description

This function returns a data set defined in the first parameter (`data`) with an additional column (`prodID`). Two products are treated as being matched if they have the same `prodID` value.

Usage

```
data_matching(
  data,
  start,
  end,
  interval = FALSE,
  variables = c(),
  codeIN = TRUE,
  codeOUT = TRUE,
  description = TRUE,
  onlydescription = FALSE,
  precision = 0.95
)
```

Arguments

data	The user's data frame with information about products to be matched. It must contain columns: time (as Date in format: year-month-day, e.g. '2020-12-01') and at least one of the following columns: codeIN (as numeric, factor or character), codeOUT (as numeric, factor or character) and description (as character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the matching process concerns only two periods defined by start and end parameters (then the interval is set to FALSE) or whether that function is to match products sold during the whole time interval <start, end>.
variables	The optional parameter describing the vector of additional column names. Values of these additional columns must be identical for matched products.
codeIN	A logical value, e.g. if there are retailer (internal) product codes (as numeric or character) written in codeIN column and there is a need to use that column while data matching, then that parameter should be set to TRUE. Otherwise it is set to FALSE.
codeOUT	A logical value, e.g. if there are external product codes, such as GTIN or SKU (as numeric or character) written in codeOUT column and there is a need to use that column while data preparing then, that parameter should be set to TRUE. Otherwise it is set to FALSE.
description	A logical value, e.g. if there are product labels (as character) written in description column and there is a need to use that column while data preparing, then that parameter should be set to TRUE. Otherwise it is set to FALSE.
onlydescription	A logical value indicating whether products with identical labels (described in the description) are to be matched.
precision	A threshold value for the Jaro-Winkler distance measure when comparing labels (its value must belong to the interval [0,1]). Two labels are treated as similar enough if their Jaro-Winkler distance exceeds the precision value.

Value

This function returns a data set defined in the first parameter (data) with an additional column (prodID). Two products are treated as being matched if they have the same prodID value. The procedure of generating the above-mentioned additional column depends on the set of chosen columns for matching. In most extreme case, when the onlydescription parameter value is TRUE, two products are also matched if they have identical descriptions. Other cases are as follows: Case 1: Parameters codeIN, codeOUT and description are set to TRUE. Products with two identical codes or one of the codes identical and an identical description are automatically matched. Products are also matched if they have identical one of codes and the Jaro-Winkler distance of their descriptions is bigger than the precision value.Case 2: Only one of the parameters: codeIN or codeOUT are set to TRUE and also the description parameter is set to TRUE. Products with an identical chosen code and an identical description are automatically matched. In the second stage, products are also matched if they have an identical chosen code and the Jaro-Winkler distance of their descriptions is bigger than the precision value. Case 3: Parameters codeIN and codeOUT are set to TRUE and

the parameter description is set to FALSE. In this case, products are matched if they have both codes identical. Case 4: Only the parameter description is set to TRUE. This case requires the onlydescription parameter to be TRUE and then the matching process is based only on product labels (two products are matched if they have identical descriptions). Case 5: Only one of the parameters: codeIN or codeOUT are set to TRUE and the description parameter is set to FALSE. In this case, the only reasonable option is to return the prodID column which is identical with the chosen code column. Please note that if the set of column names defined in the variables parameter is not empty, then the values of these additional columns must be identical while product matching.

Examples

```
data_matching(dataMATCH, start="2018-12",end="2019-02",onlydescription=TRUE,interval=TRUE)
data_matching(dataMATCH, start="2018-12",end="2019-02",precision=0.98, interval=TRUE)
```

---

data_norm *Normalization of grammage units and recalculation of prices and quantities with respect to these units*

Description

The function normalizes grammage units of products and recalculates product prices and quantities with respect to these normalized grammage units.

Usage

```
data_norm(
  data = data.frame(),
  rules = list(c("ml", "l", 1000), c("g", "kg", 1000)),
  all = TRUE
)
```

Arguments

data	The user's data frame. The data frame must contain the following columns: prices (as positive numeric), quantities (as positive numeric), grammage (as character) and unit (as character).
rules	User rules for transforming grammage, unit, prices and quantities of products. For instance, a rule ("ml","l",1000) changes the 'old' grammage unit: ml into the new one: l on the basis of the provided relation: 1000ml=1l. As a consequence, for each product which is sold in liters l , the unit price and quantity are calculated.
all	A logical value indicating whether the resulting data frame is to be limited to products with detected grammage. Its default value is TRUE which means that not transformed rows (products) are also returned.

Value

The function returns the user's data frame with two transformed columns: `grammage` and `unit`, and two rescaled columns: `prices` and `quantities`. The above-mentioned transformation and rescaling take into consideration the user `rules`. Recalculated prices and quantities concern grammage units defined as the second parameter in the given rule.

Examples

```
# Preparing a data set
data<-data_unit(dataU,units=c("g","ml","kg","l"),multiplication="x")
# Normalization of grammage units
data_norm(data, rules=list(c("ml","l",1000),c("g","kg",1000)))
```

---

data_preparing *Preparing a data set for further data processing or price index calculations*

Description

This function returns a prepared data frame based on the user's data set. The resulting data frame is ready for further data processing (such as data selecting, matching or filtering) and it is also ready for price index calculations (if only it contains required columns).

Usage

```
data_preparing(
  data,
  time = NULL,
  prices = NULL,
  quantities = NULL,
  prodID = NULL,
  retID = NULL,
  description = NULL,
  codeIN = NULL,
  codeOUT = NULL,
  grammage = NULL,
  unit = NULL,
  additional = c()
)
```

Arguments

`data`	The user's data frame to be prepared. The user must indicate columns: `time` (as Date or character type, allowed formats are, eg.: '2020-03' or '2020-12-28'), `prices` and `quantities` (as numeric). Optionally, the user may also indicate columns: `prodID`, `codeIN`, `codeOUT`, `retID` (as numeric, factor or character), `description` (as character), `grammage` (as numeric or character), `unit` (as character) and other columns specified by the `additional` parameter.

time	A character name of the column which provides transaction dates.
prices	A character name of the column which provides product prices.
quantities	A character name of the column which provides product quantities.
prodID	A character name of the column which provides product IDs. The prodID column should include unique product IDs used for product matching (as numeric or character). It is not obligatory to consider this column while data preparing but it is required while price index calculating (to obtain it, please see data_matching).
retID	A character name of the column which provides outlet IDs (retailer sale points). The retID column should include unique outlet IDs used for aggregating subindices over outlets. It is not obligatory to consider this column while data preparing but it is required while final price index calculating (to obtain it, please see the final_index or final_index2 function).
description	A character name of the column which provides product descriptions. It is not obligatory to consider this column while data preparing but it is required while product selecting (please see the data_selecting function).
codeIN	A character name of the column which provides internal product codes (from the retailer). It is not obligatory to consider this column while data preparing but it may be required while product matching (please see the data_matching function).
codeOUT	A character name of the column which provides external product codes (e.g. GTIN or SKU). It is not obligatory to consider this column while data preparing but it may be required while product matching (please see the data_matching function).
grammage	A character name of the numeric column which provides the grammage of products
unit	A character name of the column which provides the unit of the grammage of products
additional	A character vector of names of additional columns to be considered while data preparing (records with missing values are deleted).

Value

The resulting data frame is free from missing values, zero or negative prices and quantities. As a result, column time is set to be Date type (in format: 'Year-Month-01'), columns prices and quantities are set to be numeric. If the column description is selected, then it is set to be character type. If columns: prodID, retID, codeIN or codeOUT are selected, then they are set to be factor type.

Examples

```
data_preparing(milk, time="time",prices="prices",quantities="quantities")
data_preparing(dataCOICOP, time="time",
prices="prices",quantities="quantities",additional="coicop")
```

`data_selecting` *Selecting products from the user's data set for further price index calculations*

Description

The function returns a subset of the user's data set obtained by selection based on keywords and phrases.

Usage

```
data_selecting(
  data,
  include = c(),
  must = c(),
  exclude = c(),
  sensitivity = FALSE,
  coicop = NULL
)
```

Arguments

`data`	The user's data frame. It must contain a column `description` (as character).
`include`	A vector consisting of words and phrases. The function reduces the data set to one in which the `description` column contains any of these values.
`must`	A vector consisting of words and phrases. The function reduces the data set to one in which the `description` column contains each of these values.
`exclude`	A vector consisting of words and phrases. The function reduces the data set to one in which the `description` column does not contain any of these values.
`sensitivity`	A logical parameter indicating whether sensitivity to lowercase and uppercase letters is taken into consideration (if yes, its value is TRUE).
`coicop`	An optional parameter indicating a value for an additional column `coicop` which is added to the resulting data frame

Value

The function returns a subset of the user's data set obtained by selection based on keywords and phrases defined by parameters: `include`, `must` and `exclude` (an additional column `coicop` is optional). Providing values of these parameters, please remember that the procedure distinguishes between uppercase and lowercase letters only when `sensitivity` is set to TRUE.

Examples

```
data_selecting(milk, include=c("milk"), must=c("UHT"))
data_selecting(milk, must=c("milk"), exclude=c("paust"))
```

data_unit *Providing information about the grammage and unit of products*

Description

The function returns the grammage and unit of products as two additional columns.

Usage

```
data_unit(
  data = data.frame(),
  units = c("g", "ml", "kg", "l"),
  multiplication = "x",
  space = 1
)
```

Arguments

data	The user's data frame. The data frame must contain the description column (as character).
units	Units of products which are to be detected
multiplication	A sign of the multiplication used in product descriptions
space	A maximum space between the product grammage and its unit

Value

The function returns the user's data frame with two additional columns: grammage and unit (both are character type). The values of these columns are extracted from product descriptions on the basis of provided units. Please note, that the function takes into consideration a sign of the multiplication, e.g. if the product description contains: '20x50 g', we obtain: grammage: 100 and unit: g for that product (for multiplication set to 'x').

Examples

```
data_unit(dataU,units=c("g","ml","kg","l"),multiplication="x")
```

davies *Calculating the bilateral Davies price index*

Description

This function returns a value (or vector of values) of the bilateral Davies price index.

Usage

```
davies(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Davies price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Davies, G. R. (1924). *The Problem of a Standard Index Number Formula.* Journal of the American Statistical Association, 19 (146), 180-188.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
davies(sugar, start="2018-12", end="2019-12")
davies(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

dissimilarity *Calculating the relative price and/or quantity dissimilarity measure between periods*

Description

This function returns a value of the relative price and/or quantity dissimilarity measure.

Usage

```
dissimilarity(data, period1, period2, type = "p")
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
period1	The first period (as character) limited to the year and month, e.g. '2019-03'.
period2	The second period (as character) limited to the year and month, e.g. '2019-04'.
type	The parameter indicates what type of dissimilarity measure is to be calculated. Possible values of the type parameter are: p (for the price dissimilarity measure calculation), q (for the quantity dissimilarity measure calculation) or pq (for the dSPQ measure calculation, i.e. the measure of relative price and quantity dissimilarity - see References).

Value

This function returns a value of the relative price (dSP) and/or quantity (dSQ) dissimilarity measure. In a special case, when the type parameter is set to pq, the function provides the value of dSPQ measure (the relative price and quantity dissimilarity measure calculated as min(dSP,dSQ).

References

Diewert, E. (2020). *The Chain Drift Problem and Multilateral Indexes.* Chapter 6 in: Consumer Price Index Theory (draft)

Examples

```
dissimilarity(milk, period1="2018-12",period2="2019-12",type="q")
dissimilarity(milk, period1="2018-12",period2="2019-12",type="pq")
```

---

dissimilarity_fig *Presenting the relative price and/or quantity dissimilarity measure over time*

Description

This function presents values of the relative price and/or quantity dissimilarity measure over time.

Usage

```
dissimilarity_fig(
  data,
  start,
  end,
  type = "p",
  benchmark = "end",
  figure = TRUE
)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. '2019-03'.
end	The research period (as character) limited to the year and month, e.g. '2019-07'.
type	The parameter indicates what type of dissimilarity measure is to be calculated. Possible values of the type parameter are: p (for the price dissimilarity measure calculation), q (for the quantity dissimilarity measure calculation) or pq (for the dSPQ measure calculation, i.e. the measure of relative price and quantity dissimilarity - see References).
benchmark	The benchmark period (as character) limited to the year and month, e.g. '2019-07'.
figure	A logical parameter indicating the resulting object. If it is TRUE, the function presents the above-mentioned dissimilarities over time via a figure. Otherwise, the function returns a dataframe.

Value

This function presents values of the relative price and/or quantity dissimilarity measure over time. The user can choose a benchmark period (defined by benchmark) and the type of dissimilarity measure is to be calculated (defined by type). The obtained results of dissimilarities over time can be presented in a dataframe form or via a figure (the default value of figure is TRUE, which results in a figure).

References

Diewert, E. (2020). *The Chain Drift Problem and Multilateral Indexes.* Chapter 6 in: Consumer Price Index Theory (draft)

Examples

```
dissimilarity_fig(milk, start="2018-12",end="2019-12",type="q",figure=FALSE)
dissimilarity_fig(milk, start="2018-12",end="2019-12",type="pq",benchmark="start")
```

---

drobisch *Calculating the bilateral Drobisch price index*

Description

This function returns a value (or vector of values) of the bilateral Drobisch price index.

Usage

```
drobisch(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Drobisch price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Drobisch, M. W. (1871). *Ueber einige Einwurfe gegen die in diesen Jahrbuchern veroffentlichte neue Methode, die Veranderungen der Waarenpreise und des Geldwerths zu berechten*.Jahrbucher fur Nationalokonomie und Statistik, Vol. 16, s. 416-427.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
drobisch(sugar, start="2018-12", end="2019-12")
drobisch(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

dutot *Calculating the unweighted Dutot price index*

Description

This function returns a value (or vector of values) of the unweighted bilateral Dutot price index.

Usage

```
dutot(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric) and prodID (as numeric, factor or character). A column quantities (as positive numeric) is also needed because this function uses unit values as monthly prices.
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the unweighted bilateral Dutot price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Dutot, C. F., (1738). *Reflexions Politiques sur les Finances et le Commerce*. The Hague: Les Freres Vaillant et Nicolas Prevost, Vol. 1.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
dutot(sugar, start="2018-12", end="2019-12")
dutot(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

final_index *The most general package function to compute the price dynamics*

Description

This function returns a value or values of the selected (final) price index taking into consideration aggregation over product subgroups and/or over outlets.

Usage

```
final_index(
  datasets = list(),
  start,
  end,
  formula = "fisher",
  window = 13,
  splice = "movement",
  base = start,
  sigma = 0.7,
  aggrret = "tornqvist",
  aggrsets = "tornqvist",
  interval = FALSE
)
```

Arguments

datasets	The user's list of data frames with subgroups of sold products. Each data frame must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric), prodID (as numeric, factor or character) and retID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
formula	The character string indicating the (final or main) price index formula is to be calculated. To see available options please use the link: PriceIndices.
window	The length of the time window if the multilateral index is selected (as positive integer: typically multilateral methods are based on the 13-month time window and thus the default value is 13).
splice	A character string indicating the splicing method (if the multilateral splicing index is selected). Available options are: "movement", "window","half", "mean" and also "window_published", "half_published" and "mean_published".
base	The prior period used in the Young- or Lowe-type price indices (as character) limited to the year and month, e.g. "2020-01".
sigma	The elasticity of substitution parameter used in the Lloyed-Moulton and AG Mean indices (as numeric).
aggrret	A character string indicating the formula for aggregation over outlets (retailer sale points). Available options are: "none", "laspeyres", "paasche", "geolaspeyres", "geopaasche", "fisher", "tornqvist", "arithmetic" and "geometric". The first option means that there is no aggregating over outlets. The last two options mean unweighted methods of aggregating, i.e. the arithmetic or geometric mean is used.
aggrsets	A character string indicating the formula for aggregation over product subgroups. Available options are: "none", "laspeyres", "paasche", "geolaspeyres", "geopaasche", "fisher", "tornqvist", "arithmetic" and "geometric". The first option means that there is no aggregating over product subgroups. The last two

options mean unweighted methods of aggregating, i.e. the arithmetic or geometric mean is used.

`interval` A logical value indicating whether the function is to provide the price index comparing the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be presented (the fixed base month is defined by `start`).

Value

This function returns a value or values of the selected (final) price index taking into consideration aggregation over product subgroups and/or over outlets (retailer sale points defined in `retID` column) . To be more precise: if both types of aggregation are selected, then for each subgroup of products and for each outlet (point of sale) price indices are calculated separately and then aggregated (according to the aggregation methods indicated) to the form of the final price index. If the `interval` parameter is set to TRUE then it returns a data frame with two columns: dates and final index values (after optional aggregating). Please note that different index formulas may use different time intervals (or time periods) for calculations and each time, aggregation over outlets is done for the set of retIDs being available during the whole considered time interval.

Examples

```
 final_index(datasets=list(milk),start="2018-12",end="2020-02",
formula="walsh",aggrret="paasche",aggrsets="none")
## defining two subgroups of milk
g1<-dplyr::filter(milk, milk$description=="full-fat milk UHT")
g2<-dplyr::filter(milk, milk$description=="low-fat milk UHT")
## Final price index calculations (for the whole time interval)
## with aggregating over subgroups g1 and g2 and over outlets
## Please note that the default value (formula) for aggregating over outlets is "tornqvist""
final_index(datasets=list(g1,g2), start="2018-12",
end="2019-12",formula="fisher",aggrsets="geometric",interval=TRUE)
```

`final_index2` *The most general package function to compute the price dynamics*

Description

This function returns a value or values of the selected (final) price index taking into consideration aggregation over product subgroups and/or over outlets. Optionally, the function returns a data frame or a figure presenting calculated indices, i.e. the price index for the whole data set and price indices for product subgroups.

Usage

```
final_index2(
  data = data.frame(),
  by,
  all = FALSE,
```

```
  start,
  end,
  formula = "fisher",
  window = 13,
  splice = "movement",
  base = start,
  sigma = 0.7,
  aggrret = "tornqvist",
  aggrsets = "tornqvist",
  interval = FALSE,
  figure = FALSE
)
```

Arguments

data	The user's data frame with subgroups of sold products (see by parameter). Each data frame must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric), prodID (as numeric, factor or character) and retID (as numeric, factor or character). An additional column indicated via by parameter is also needed.
by	The column name indicating grouping variable, i.e. this column is used for creating subgroups of products.
all	A logical value indicating whether the the selected price index is to be calculated only for the whole set of products or also for created subgroups of products (then all is set to TRUE).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
formula	The character string indicating the (final or main) price index formula is to be calculated. To see available options please use the link: PriceIndices.
window	The length of the time window if the multilateral index is selected (as positive integer: typically multilateral methods are based on the 13-month time window and thus the default value is 13).
splice	A character string indicating the splicing method (if the multilateral splicing index is selected). Available options are: "movement", "window","half", "mean" and also "window_published", "half_published" and "mean_published".
base	The prior period used in the Young- or Lowe-type price indices (as character) limited to the year and month, e.g. "2020-01".
sigma	The elasticity of substitution parameter used in the Lloyed-Moulton and AG Mean indices (as numeric).
aggrret	A character string indicating the formula for aggregation over outlets (retailer sale points). Available options are: "none", "laspeyres", "paasche", "geolaspeyres", "geopaasche", "fisher", "tornqvist", "arithmetic" and "geometric". The first option means that there is no aggregating over outlets. The last two options mean unweighted methods of aggregating, i.e. the arithmetic or geometric mean is used.

aggrsets	A character string indicating the formula for aggregation over product subgroups. Available options are: "none", "laspeyres", "paasche", "geolaspeyres", "geopaasche", "fisher", "tornqvist", "arithmetic" and "geometric". The first option means that there is no aggregating over product subgroups. The last two options mean unweighted methods of aggregating, i.e. the arithmetic or geometric mean is used.
interval	A logical value indicating whether the function is to provide the price index comparing the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be presented (the fixed base month is defined by start).
figure	A logical value indicating whether the function returns a figure presenting all calculated indices (it works if all and interval are set to TRUE)

Value

This function returns a value or values of the selected (final) price index taking into consideration aggregation over product subgroups and/or over outlets (retailer sale points defined in retID column). Optionally, the function returns a data frame or a figure presenting calculated indices, i.e. the price index for the whole data set and price indices for product subgroups.To be more precise: if both types of aggregation are selected, then for each subgroup of products and for each outlet (point of sale) price indices are calculated separately and then aggregated (according to the aggregation methods indicated) to the form of the final price index. If the interval parameter is set to TRUE then it returns a data frame (or a figure) with dates and final index values (after optional aggregating). Please note that different index formulas may use different time intervals (or time periods) for calculations and each time, aggregation over outlets is done for the set of retIDs being available during the whole considered time interval.

Examples

```
final_index2(data=coffee, by="description",all=TRUE,start="2018-12",end="2019-12",
formula="fisher",interval=TRUE,aggrsets="laspeyres",aggrret="none",figure=FALSE)
final_index2(data=coffee, by="retID",all=TRUE,start="2018-12",end="2019-12",
formula="fisher",interval=TRUE,aggrsets="none",aggrret="none",figure=TRUE)
```

---

fisher *Calculating the bilateral Fisher price index*

Description

This function returns a value (or vector of values) of the bilateral Fisher price index.

Usage

```
fisher(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Fisher price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating,please use the final_index or the final_index2 function).

References

Fisher, I. (1922). *The Making of Index Numbers*. Boston: Houghton Mifflin.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
fisher(sugar, start="2018-12", end="2019-12")
fisher(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

geary_khamis *Calculating the bilateral Geary-Khamis price index*

Description

This function returns a value (or vector of values) of the bilateral Geary-Khamis price index.

Usage

```
geary_khamis(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Geary-Khamis price index depending on the interval parameter (please use gk function to calculate the multilateral Geary-Khamis price index). If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Geary, R. G. (1958). *A Note on Comparisons of Exchange Rates and Purchasing Power between Countries.* Journal of the Royal Statistical Society, Series A, 121, 97-99.

Khamis, S. H. (1970). *Properties and Conditions for the Existence of a new Type of Index Number.* Sankhya Series B32, 81-98.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
geary_khamis(sugar, start="2018-12", end="2019-12")
geary_khamis(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

geks *Calculating the multilateral GEKS price index*

Description

This function returns a value of the multilateral GEKS price index (to be more precise: the GEKS index based on the Fisher formula).

Usage

```
geks(data, start, end, wstart = start, window = 13)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
wstart	The beginning of the time interval (which is used by multilateral methods) limited to the year and month, e.g. "2020-01".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

Value

This function returns a value of the multilateral GEKS price index (to be more precise: the GEKS index based on the Fisher formula) which considers the time window defined by wstart and window parameters. It measures the price dynamics by comparing period end to period start (both start and end must be inside the considered time window). To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Examples

```
geks(milk, start="2019-01", end="2019-08",window=10)
geks(milk, start="2018-12", end="2019-12")
```

geksj *Calculating the multilateral GEKS price index based on the Jevons formula (typical notation: GEKS-J)*

Description

This function returns a value of the multilateral GEKS-J price index (to be more precise: the GEKS index based on the Jevons formula).

Usage

```
geksj(data, start, end, wstart = start, window = 13)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character). A column quantities is needed because this function uses unit values as monthly prices.
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
wstart	The beginning of the time interval (which is used by multilateral methods) limited to the year and month, e.g. "2020-01".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

Value

This function returns a value of the multilateral GEKS-J price index (to be more precise: the GEKS index based on the Jevons formula) which considers the time window defined by wstart and window parameters. It measures the price dynamics by comparing period end to period start (both start and end must be inside the considered time window). To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Examples

```
geksj(milk, start="2019-01", end="2019-08",window=10)
geksj(milk, start="2018-12", end="2019-12")
```

`geksj_fbew` *Extending the multilateral GEKS-J price index by using the FBEW method.*

Description

This function returns a value of the multilateral GEKS-J price index (i.e. the GEKS price index based on the Jevons formula) extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
geksj_fbew(data, start, end)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` is needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral GEKS-J price index (i.e. the GEKS price index based on the Jevons formula) extended by using the FBEW (Fixed Base Monthly Expanding Window) method. The FBEW method uses a time window with a fixed base month every year (December). The window is enlarged every month with one month in order to include information from a new month. The full window length (13 months) is reached in December of each year. The function measures the price dynamics between periods `end` and `start`. The month of the `start` parameter must be December. If the distance between `end` and `start` exceeds 13 months, then internal Decembers play a role of chain-linking months. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
geksj_fbew(milk, start="2018-12", end="2019-08")
```

`geksj_fbmw` *Extending the multilateral GEKS-J price index by using the FBMW method.*

Description

This function returns a value of the multilateral GEKS-J price index (i.e. the GEKS price index based on the Jevons formula) extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
geksj_fbmw(data, start, end)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` is needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral GEKS-J price index (i.e. the GEKS price index based on the Jevons formula) extended by using the FBMW (Fixed Base Moving Window) method. It measures the price dynamics between periods `end` and `start` and it uses a 13-month time window with a fixed base month taken as `year(end)-1`. If the distance between `end` and `start` exceeds 13 months, then internal Decembers play a role of chain-linking months. The month of the `start` parameter must be December. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Lamboray, C.(2017). *The Geary Khamis index and the Lehr index: how much do they differ?* Paper presented at the 15th Ottawa Group meeting, 10-12 May 2017, Elville am Rhein, Germany.

Examples

```
geksj_fbmw(milk, start="2019-12", end="2020-04")
```

`geksj_splice` *Extending the multilateral GEKS-J price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-J price index (GEKS based on the Jevons formula) extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`).

Usage

```
geksj_splice(
  data,
  start,
  end,
  window = 13,
  splice = "movement",
  interval = FALSE
)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` is needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
splice	A character string indicating the splicing method. Available options are: "movement", "window","half","mean", "window_published","half_published","mean_published".
interval	A logical value indicating whether the function is to provide the price index comparing the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base multilateral indices are to be presented (the fixed base month is defined by start).

Value

This function returns a value or values (depending on interval parameter) of the multilateral GEKS-J price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see References). The time window starts in start and should consist of at least two months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Chessa, A. G. (2019). *A Comparison of Index Extension Methods for Multilateral Methods.* Paper presented at the 16th Meeting of the Ottawa Group on Price Indices, 8-10 May 2019, Rio de Janeiro, Brazil.

de Haan, J., van der Grient, H.A. (2011). *Eliminating chain drift in price indexes based on scanner data.* Journal of Econometrics, 161, 36-46.

Krsinich, F. (2014). *The FEWS Index: Fixed Effects with a Window Splice? Non-Revisable Quality-Adjusted Price Indices with No Characteristic Information.* Paper presented at the UNECE-ILO Meeting of the Group of Experts on Consumer Price Indices, 2-4 May 2016, Geneva, Switzerland.

de Haan, J.(2015). *A Framework for Large Scale Use of Scanner Data in the Dutch CPI.* Paper presented at the 14th Ottawa Group meeting, Tokyo, Japan.

Diewert, W.E., and Fox, K.J. (2017). *Substitution Bias in Multilateral Methods for CPI Construction using Scanner Data.* Discussion paper 17-02, Vancouver School of Economics, The University of British Columbia, Vancouver, Canada.

Examples

```
geksj_splice(milk, start="2018-12", end="2020-02",splice="half")
geksj_splice(milk, start="2018-12", end="2020-02",window=10,interval=TRUE)
```

geksl *Calculating the multilateral GEKS-L price index*

Description

This function returns a value of the multilateral GEKS-L price index (to be more precise: the GEKS index based on the Laspeyres formula).

Usage

```
geksl(data, start, end, wstart = start, window = 13)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
wstart	The beginning of the time interval (which is used by multilateral methods) limited to the year and month, e.g. "2020-01".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

Value

This function returns a value of the multilateral GEKS-L price index (to be more precise: the GEKS index based on the Laspeyres formula) which considers the time window defined by wstart and window parameters. It measures the price dynamics by comparing period end to period start (both start and end must be inside the considered time window). To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Examples

```
geksl(milk, start="2019-01", end="2019-08",window=10)
geksl(milk, start="2018-12", end="2019-12")
```

`geksl_fbew` *Extending the multilateral GEKS-L price index by using the FBEW method.*

Description

This function returns a value of the multilateral GEKS-L price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
geksl_fbew(data, start, end)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral GEKS-L price index (the GEKS index based on the Laspeyres formula) extended by using the FBEW (Fixed Base Monthly Expanding Window) method. The FBEW method uses a time window with a fixed base month every year (December). The window is enlarged every month with one month in order to include information from a new month. The full window length (13 months) is reached in December of each year. The function measures the price dynamics between periods `end` and `start`. The month of the `start` parameter must be December. If the distance between `end` and `start` exceeds 13 months, then internal Decembers play a role of chain-linking months. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating,please use the `final_index` or the `final_index2` function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
geksl_fbew(milk, start="2018-12", end="2019-08")
```

`geksl_fbmw` *Extending the multilateral GEKS-L price index by using the FBMW method.*

Description

This function returns a value of the multilateral GEKS-L price index extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
geksl_fbmw(data, start, end)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral GEKS-L price index (the GEKS index based on the Laspeyres formula) extended by using the FBMW (Fixed Base Moving Window) method. It measures the price dynamics between periods `end` and `start` and it uses a 13-month time window with a fixed base month taken as `year(end)-1`. If the distance between `end` and `start` exceeds 13 months, then internal Decembers play a role of chain-linking months. The month of the `start` parameter must be December. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Lamboray, C.(2017). *The Geary Khamis index and the Lehr index: how much do they differ?* Paper presented at the 15th Ottawa Group meeting, 10-12 May 2017, Elville am Rhein, Germany.

Examples

```
geksl_fbmw(milk, start="2019-12", end="2020-04")
```

`geksl_splice` *Extending the multilateral GEKS-L price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-L price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`).

Usage

```
geksl_splice(
  data,
  start,
  end,
  window = 13,
  splice = "movement",
  interval = FALSE
)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`window`	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

`splice` A character string indicating the splicing method. Available options are: "movement", "window","half","mean", "window_published","half_published","mean_published".

`interval` A logical value indicating whether the function is to provide the price index comparing the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base multilateral indices are to be presented (the fixed base month is defined by `start`).

Value

This function returns a value or values (depending on `interval` parameter) of the multilateral GEKS-L price index (the GEKS index based on the Laspeyres formula) extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`). The time window starts in `start` and should consist of at least two months. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Chessa, A. G. (2019). *A Comparison of Index Extension Methods for Multilateral Methods.* Paper presented at the 16th Meeting of the Ottawa Group on Price Indices, 8-10 May 2019, Rio de Janeiro, Brazil.

de Haan, J., van der Grient, H.A. (2011). *Eliminating chain drift in price indexes based on scanner data.* Journal of Econometrics, 161, 36-46.

Krsinich, F. (2014). *The FEWS Index: Fixed Effects with a Window Splice? Non-Revisable Quality-Adjusted Price Indices with No Characteristic Information.* Paper presented at the UNECE-ILO Meeting of the Group of Experts on Consumer Price Indices, 2-4 May 2016, Geneva, Switzerland.

de Haan, J.(2015). *A Framework for Large Scale Use of Scanner Data in the Dutch CPI.* Paper presented at the 14th Ottawa Group meeting, Tokyo, Japan.

Diewert, W.E., and Fox, K.J. (2017). *Substitution Bias in Multilateral Methods for CPI Construction using Scanner Data.* Discussion paper 17-02, Vancouver School of Economics, The University of British Columbia, Vancouver, Canada.

Examples

```
geksl_splice(milk, start="2018-12", end="2020-02",splice="half")
geksl_splice(milk, start="2018-12", end="2020-02",window=10,interval=TRUE)
```

geksw *Calculating the multilateral GEKS price index based on the Walsh formula (GEKS-W)*

Description

This function returns a value of the multilateral GEKS-W price index, i.e. the GEKS price index based on the superlative Walsh index formula.

Usage

```
geksw(data, start, end, wstart = start, window = 13)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
wstart	The beginning of the time interval (which is used by multilateral methods) limited to the year and month, e.g. "2020-01".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

Value

This function returns a value of the multilateral GEKS-W price index (to be more precise: the GEKS index based on the Walsh formula) which considers the time window defined by wstart and window parameters. It measures the price dynamics by comparing period end to period start (both start and end must be inside the considered time window). To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Walsh, C. M. (1901). *The Measurement of General Exchange Value*. The MacMillan Company, New York.

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Examples

```
geksw(milk, start="2019-01", end="2019-08",window=10)
geksw(milk, start="2018-12", end="2019-12")
```

`geksw_fbew` *Extending the multilateral GEKS-W price index by using the FBEW method.*

Description

This function returns a value of the multilateral GEKS-W price index (GEKS based on the Walsh formula) extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
geksw_fbew(data, start, end)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral GEKS-W price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method. The FBEW method uses a time window with a fixed base month every year (December). The window is enlarged every month with one month in order to include information from a new month. The full window length (13 months) is reached in December of each year. The function measures the price dynamics between periods `end` and `start`. The month of the `start` parameter must be December. If the distance between `end` and `start` exceeds 13 months, then internal Decembers play a role of chain-linking months. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Walsh, C. M. (1901). *The Measurement of General Exchange Value*. The MacMillan Company, New York.

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
geksw_fbew(milk, start="2018-12", end="2019-08")
```

`geksw_fbmw` *Extending the multilateral GEKS-W price index by using the FBMW method.*

Description

This function returns a value of the multilateral GEKS-W price index (GEKS based on the Walsh formula) extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
geksw_fbmw(data, start, end)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral GEKS-W price index extended by using the FBMW (Fixed Base Moving Window) method. It measures the price dynamics between periods `end` and `start` and it uses a 13-month time window with a fixed base month taken as `year(end)-1`. If the distance between `end` and `start` exceeds 13 months, then internal Decembers play a role of chain-linking months. The month of the `start` parameter must be December. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Walsh, C. M. (1901). *The Measurement of General Exchange Value*. The MacMillan Company, New York.

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Lamboray, C.(2017). *The Geary Khamis index and the Lehr index: how much do they differ?* Paper presented at the 15th Ottawa Group meeting, 10-12 May 2017, Elville am Rhein, Germany.

Examples

```
geksw_fbmw(milk, start="2019-12", end="2020-04")
```

`geksw_splice` *Extending the multilateral GEKS-W price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-W price index (GEKS based on the Walsh formula) extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`).

Usage

```
geksw_splice(
  data,
  start,
  end,
  window = 13,
  splice = "movement",
  interval = FALSE
)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".

end	The research period (as character) limited to the year and month, e.g. "2020-04".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
splice	A character string indicating the splicing method. Available options are: "movement", "window","half","mean", "window_published","half_published","mean_published".
interval	A logical value indicating whether the function is to provide the price index comparing the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base multilateral indices are to be presented (the fixed base month is defined by start).

Value

This function returns a value or values (depending on interval parameter) of the multilateral GEKS-W price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see References). The time window starts in start and should consist of at least two months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Chessa, A. G. (2019). *A Comparison of Index Extension Methods for Multilateral Methods.* Paper presented at the 16th Meeting of the Ottawa Group on Price Indices, 8-10 May 2019, Rio de Janeiro, Brazil.

de Haan, J., van der Grient, H.A. (2011). *Eliminating chain drift in price indexes based on scanner data.* Journal of Econometrics, 161, 36-46.

Krsinich, F. (2014). *The FEWS Index: Fixed Effects with a Window Splice? Non-Revisable Quality-Adjusted Price Indices with No Characteristic Information.* Paper presented at the UNECE-ILO Meeting of the Group of Experts on Consumer Price Indices, 2-4 May 2016, Geneva, Switzerland.

de Haan, J.(2015). *A Framework for Large Scale Use of Scanner Data in the Dutch CPI.* Paper presented at the 14th Ottawa Group meeting, Tokyo, Japan.

Diewert, W.E., and Fox, K.J. (2017). *Substitution Bias in Multilateral Methods for CPI Construction using Scanner Data.* Discussion paper 17-02, Vancouver School of Economics, The University of British Columbia, Vancouver, Canada.

Examples

```
geksw_splice(milk, start="2018-12", end="2020-02",splice="half")
geksw_splice(milk, start="2018-12", end="2020-02",window=10,interval=TRUE)
```

geks_fbew *Extending the multilateral GEKS price index by using the FBEW method.*

Description

This function returns a value of the multilateral GEKS price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
geks_fbew(data, start, end)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral GEKS price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method. The FBEW method uses a time window with a fixed base month every year (December). The window is enlarged every month with one month in order to include information from a new month. The full window length (13 months) is reached in December of each year. The function measures the price dynamics between periods end and start. The month of the start parameter must be December. If the distance between end and start exceeds 13 months, then internal Decembers play a role of chain-linking months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
geks_fbew(milk, start="2018-12", end="2019-08")
```

`geks_fbmw` *Extending the multilateral GEKS price index by using the FBMW method.*

Description

This function returns a value of the multilateral GEKS price index extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
geks_fbmw(data, start, end)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral GEKS price index extended by using the FBMW (Fixed Base Moving Window) method. It measures the price dynamics between periods `end` and `start` and it uses a 13-month time window with a fixed base month taken as `year(end)-1`. If the distance between `end` and `start` exceeds 13 months, then internal Decembers play a role of chain-linking months. The month of the `start` parameter must be December. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Lamboray, C.(2017). *The Geary Khamis index and the Lehr index: how much do they differ?* Paper presented at the 15th Ottawa Group meeting, 10-12 May 2017, Elville am Rhein, Germany.

Examples

```
geks_fbmw(milk, start="2019-12", end="2020-04")
```

`geks_splice` *Extending the multilateral GEKS price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`).

Usage

```
geks_splice(
  data,
  start,
  end,
  window = 13,
  splice = "movement",
  interval = FALSE
)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`window`	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
`splice`	A character string indicating the splicing method. Available options are: "movement", "window","half","mean", "window_published","half_published","mean_published".
`interval`	A logical value indicating whether the function is to provide the price index comparing the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base multilateral indices are to be presented (the fixed base month is defined by `start`).

Value

This function returns a value or values (depending on `interval` parameter) of the multilateral GEKS price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`). The time window starts in `start` and should consist of at least two months. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Chessa, A. G. (2019). *A Comparison of Index Extension Methods for Multilateral Methods.* Paper presented at the 16th Meeting of the Ottawa Group on Price Indices, 8-10 May 2019, Rio de Janeiro, Brazil.

de Haan, J., van der Grient, H.A. (2011). *Eliminating chain drift in price indexes based on scanner data.* Journal of Econometrics, 161, 36-46.

Krsinich, F. (2014). *The FEWS Index: Fixed Effects with a Window Splice? Non-Revisable Quality-Adjusted Price Indices with No Characteristic Information.* Paper presented at the UNECE-ILO Meeting of the Group of Experts on Consumer Price Indices, 2-4 May 2016, Geneva, Switzerland.

de Haan, J.(2015). *A Framework for Large Scale Use of Scanner Data in the Dutch CPI.* Paper presented at the 14th Ottawa Group meeting, Tokyo, Japan.

Diewert, W.E., and Fox, K.J. (2017). *Substitution Bias in Multilateral Methods for CPI Construction using Scanner Data.* Discussion paper 17-02, Vancouver School of Economics, The University of British Columbia, Vancouver, Canada.

Examples

```
geks_splice(milk, start="2018-12", end="2020-02",splice="half")
geks_splice(milk, start="2018-12", end="2020-02",window=10,interval=TRUE)
```

---

generate *Generating an artificial scanner dataset*

Description

This function provides artificial scanner datasets where prices and quantities are lognormally distributed.

Usage

```
generate(
  pmi = c(),
  psigma = c(),
  qmi = c(),
```

```
  qsigma = c(),
  prec = c(2, 0),
  n = 100,
  n0 = 1,
  r = 1,
  r0 = 1,
  start,
  days = FALSE
)
```

Arguments

pmi	A numeric vector indicating mi parameters for lognormally distributed prices from the subsequent months.
psigma	A numeric vector indicating sigma parameters for lognormally distributed prices from the subsequent months.
qmi	A numeric vector indicating mi parameters for lognormally distributed quantities from the subsequent months.
qsigma	A numeric vector indicating sigma parameters for lognormally distributed quantities from the subsequent months.
prec	A two-dimensional numeric vector indicating precision, i.e. the number of decimal places, for presenting prices and quantities.
n	An integer parameter indicating the number of products which are to be generated.
n0	An integer parameter indicating the first (the smallest) prodID.
r	An integer parameter indicating the number of outlets (retailer sale points) for which prices and quantities are to be generated.
r0	n0 An integer parameter indicating the first (the smallest) retID.
start	The first period in the generated data frame (as character) limited to the year and month, e.g. '2019-12'.
days	A logical parameter indicating whether the trading day in a given month is to be randomised. The default value of days is FALSE, which means that each transaction for a given month takes place on the first day of the month.

Value

This function returns an artificial scanner dataset where prices and quantities are lognormally distributed. The characteristics for these lognormal distributions are set by pmi, sigma, qmi and qsigma parameters. This function works for a fixed number of products and outlets (see n and r parameters). The generated dataset is ready for further price index calculations.

Examples

```
generate(pmi=c(1.02,1.03,1.04),psigma=c(0.05,0.09,0.02),qmi=c(3,4,4),
qsigma=c(0.1,0.1,0.15),start="2020-01",days=TRUE)
generate(pmi=c(1.02,1.03,1.04),psigma=c(0.05,0.09,0.02),qmi=c(6,6,7),
qsigma=c(0.1,0.1,0.15),start="2020-01",n=1000,n0=132578,r=10)
```

geohybrid *Calculating the bilateral geohybrid price index*

Description

This function returns a value (or vector of values) of the bilateral geohybrid price index. The geohybrid index was proposed by Bialek (2020) and it uses correlation coefficients between prices and quantities.

Usage

```
geohybrid(data, start, end, base = start, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
base	The prior period used in the geohybrid price index formula (as character) limited to the year and month, e.g. "2020-01"
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral geohybrid price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Bialek, J. (2020). *Proposition of a Hybrid Price Index Formula for the Consumer Price Index Measurement*. Equilibrium. Quarterly Journal of Economics and Economic Policy, 15(4), 697-716.

Examples

```
geohybrid(sugar, start="2019-12", end="2020-08", base="2018-12")
geohybrid(milk, start="2019-12", end="2020-08", base="2018-12", interval=TRUE)
```

`geolaspeyres` *Calculating the bilateral geo-logarithmic Laspeyres price index*

Description

This function returns a value (or vector of values) of the bilateral geo-logarithmic Laspeyres price index.

Usage

```
geolaspeyres(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral geo-logarithmic Laspeyres price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
geolaspeyres(sugar, start="2018-12", end="2019-12")
geolaspeyres(milk, start="2018-12", end="2020-01", interval=TRUE)
```

geolowe *Calculating the bilateral geometric Lowe price index*

Description

This function returns a value (or vector of values) of the bilateral geometric Lowe price index.

Usage

```
geolowe(data, start, end, base = start, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
base	The prior period used in the geometric Lowe price index formula (as character) limited to the year and month, e.g. "2020-01"
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral geometric Lowe price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
geolowe(sugar, start="2019-01", end="2020-01",base="2018-12")
geolowe(milk, start="2018-12", end="2020-01", interval=TRUE)
```

geopaasche *Calculating the bilateral geo-logarithmic Paasche price index*

Description

This function returns a value (or vector of values) of the bilateral geo-logarithmic Paasche price index.

Usage

```
geopaasche(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral geo-logarithmic Paasche price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
geopaasche(sugar, start="2018-12", end="2019-12")
geopaasche(milk, start="2018-12", end="2020-01", interval=TRUE)
```

geoyoung *Calculating the bilateral geometric Young price index*

Description

This function returns a value (or vector of values) of the bilateral geometric Young price index.

Usage

```
geoyoung(data, start, end, base = start, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
base	The prior period used in the geometric Young price index formula (as character) limited to the year and month, e.g. "2020-01"
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral geometric Young price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Young, A. H. (1992). *Alternative Measures of Change in Real Output and Prices.* Survey of Current Business, 72, 32-48.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
geoyoung(sugar, start="2019-01", end="2020-01",base="2018-12")
geoyoung(milk, start="2018-12", end="2020-01", interval=TRUE)
```

gk *Calculating the multilateral Geary-Khamis price index*

Description

This function returns a value of the multilateral Geary-Khamis price index.

Usage

```
gk(data, start, end, wstart = start, window = 13)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
wstart	The beginning of the time interval (which is used by multilateral methods) limited to the year and month, e.g. "2020-01".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

Value

This function returns a value of the multilateral Geary-Khamis price index which considers the time window defined by wstart and window parameters. The Geary-Khamis price index is calculated by using a special iterative algorithm from Chessa (2016). It measures the price dynamics by comparing period end to period start (both start and end must be inside the considered time window). To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Geary, R. G. (1958). *A Note on Comparisons of Exchange Rates and Purchasing Power between Countries.* Journal of the Royal Statistical Society, Series A, 121, 97-99.

Khamis, S. H. (1970). *Properties and Conditions for the Existence of a new Type of Index Number.* Sankhya Series B32, 81-98.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
gk(milk, start="2019-01", end="2019-08",window=10)
gk(milk, start="2018-12", end="2019-12")
```

---

gk_fbew *Extending the multilateral Geary-Khamis price index by using the FBEW method.*

Description

This function returns a value of the multilateral Geary-Khamis price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
gk_fbew(data, start, end)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral Geary-Khamis price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method. The FBEW method uses a time window with a fixed base month every year (December). The window is enlarged every month with one month in order to include information from a new month. The full window length (13 months) is reached in December of each year. The function measures the price dynamics between periods end and start. The month of the start parameter must be December. If the distance between end and start exceeds 13 months, then internal Decembers play a role of chain-linking months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Geary, R. G. (1958). *A Note on Comparisons of Exchange Rates and Purchasing Power between Countries.* Journal of the Royal Statistical Society, Series A, 121, 97-99.

Khamis, S. H. (1970). *Properties and Conditions for the Existence of a new Type of Index Number.* Sankhya Series B32, 81-98.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
gk_fbew(milk, start="2018-12", end="2019-08")
```

---

gk_fbmw *Extending the multilateral Geary-Khamis price index by using the FBMW method.*

Description

This function returns a value of the multilateral Geary-Khamis price index extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
gk_fbmw(data, start, end)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral Geary-Khamis price index extended by using the FBMW (Fixed Base Moving Window) method. It measures the price dynamics between periods end and start and it uses a 13-month time window with a fixed base month taken as year(end)-1. If the distance between end and start exceeds 13 months, then internal Decembers play a role of chain-linking months. The month of the start parameter must be December. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Geary, R. G. (1958). *A Note on Comparisons of Exchange Rates and Purchasing Power between Countries.* Journal of the Royal Statistical Society, Series A, 121, 97-99.

Khamis, S. H. (1970). *Properties and Conditions for the Existence of a new Type of Index Number.* Sankhya Series B32, 81-98.

Lamboray, C.(2017). *The Geary Khamis index and the Lehr index: how much do they differ?* Paper presented at the 15th Ottawa Group meeting, 10-12 May 2017, Elville am Rhein, Germany.

Examples

```
gk_fbmw(milk, start="2019-12", end="2020-04")
```

`gk_splice` *Extending the multilateral Geary-Khamis price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral Geary-Khamis price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`).

Usage

```
gk_splice(data, start, end, window = 13, splice = "movement", interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`window`	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
`splice`	A character string indicating the splicing method. Available options are: "movement", "window","half","mean", "window_published","half_published","mean_published".
`interval`	A logical value indicating whether the function is to provide the price index comparing the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base multilateral indices are to be presented (the fixed base month is defined by `start`).

Value

This function returns a value or values (depending on `interval` parameter) of the multilateral Geary-Khamis price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`). The time window starts in `start` and should consist of at least two months. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Chessa, A. G. (2019). *A Comparison of Index Extension Methods for Multilateral Methods.* Paper presented at the 16th Meeting of the Ottawa Group on Price Indices, 8-10 May 2019, Rio de Janeiro, Brazil.

de Haan, J., van der Grient, H.A. (2011). *Eliminating chain drift in price indexes based on scanner data.* Journal of Econometrics, 161, 36-46.

Krsinich, F. (2014). *The FEWS Index: Fixed Effects with a Window Splice? Non-Revisable Quality-Adjusted Price Indices with No Characteristic Information.* Paper presented at the UNECE-ILO Meeting of the Group of Experts on Consumer Price Indices, 2-4 May 2016, Geneva, Switzerland.

de Haan, J.(2015). *A Framework for Large Scale Use of Scanner Data in the Dutch CPI.* Paper presented at the 14th Ottawa Group meeting, Tokyo, Japan.

Diewert, W.E., and Fox, K.J. (2017). *Substitution Bias in Multilateral Methods for CPI Construction using Scanner Data.* Discussion paper 17-02, Vancouver School of Economics, The University of British Columbia, Vancouver, Canada.

Examples

```
gk_splice(milk, start="2018-12", end="2020-02",splice="half")
gk_splice(milk, start="2018-12", end="2020-02",window=10,interval=TRUE)
```

`harmonic` *Calculating the unweighted harmonic price index*

Description

This function returns a value (or vector of values) of the unweighted "unnamed" harmonic price index.

Usage

```
harmonic(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the unweighted bilateral harmonic price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
harmonic(sugar, start="2018-12", end="2019-12")
harmonic(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

hybrid *Calculating the bilateral hybrid price index*

Description

This function returns a value (or a vector of values) of the bilateral hybrid price index. The hybrid index was proposed by Bialek (2020) and it uses correlation coefficients between prices and quantities.

Usage

```
hybrid(data, start, end, base = start, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. '2020-03'.
`end`	The research period (as character) limited to the year and month, e.g. '2020-04'.
`base`	The prior period used in the hybrid price index formula (as character) limited to the year and month, e.g. '2020-01'.
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or a vector of values) of the bilateral hybrid price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices`, `final_index` or `final_index2`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or `final_index2` function).

References

Bialek, J. (2020). *Proposition of a Hybrid Price Index Formula for the Consumer Price Index Measurement*. Equilibrium. Quarterly Journal of Economics and Economic Policy, 15(4), 697-716.

Examples

```
hybrid(sugar, start="2019-12", end="2020-08", base="2018-12")
hybrid(milk, start="2019-12", end="2020-08", base="2018-12", interval=TRUE)
```

---

jevons *Calculating the unweighted Jevons price index*

Description

This function returns a value (or vector of values) of the unweighted bilateral Jevons price index.

Usage

```
jevons(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also needed because this function uses unit values as monthly prices.
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the unweighted bilateral Jevons price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Jevons, W. S., (1865). *The variation of prices and the value of the currency since 1782*. J. Statist. Soc. Lond., 28, 294-320.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
jevons(milk, start="2018-12", end="2020-01")
jevons(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

laspeyres *Calculating the bilateral Laspeyres price index*

Description

This function returns a value (or vector of values) of the bilateral Laspeyres price index.

Usage

```
laspeyres(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Laspeyres price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Laspeyres, E. (1871). *Die Berechnung einer mittleren Waarenpreissteigerung*. Jahrbucher fur Nationalokonomie und Statistik 16, 296-314.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
laspeyres(sugar, start="2018-12", end="2019-12")
laspeyres(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

lehr *Calculating the bilateral Lehr price index*

Description

This function returns a value (or vector of values) of the bilateral Lehr price index.

Usage

```
lehr(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Lehr price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Lehr, J. (1885). *Beitrage zur Statistik der Preise, insbesondere des Geldes und des Holzes.* J. D. Sauerlander, Frankfurt am Main.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
lehr(sugar, start="2018-12", end="2019-12")
lehr(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

lloyd_moulton *Calculating the bilateral Lloyd-Moulton price index*

Description

This function returns a value (or vector of values) of the bilateral Lloyd-Moulton price index.

Usage

```
lloyd_moulton(data, start, end, sigma = 0.7, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`sigma`	The elasticity of substitution parameter (as numeric).
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Lloyd-Moulton price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Lloyd, P. J. (1975). *Substitution Effects and Biases in Nontrue Price Indices.* The American Economic Review, 65, 301-313.

Moulton, B. R. (1996). *Constant Elasticity Cost-of-Living Index in Share-Relative Form.* Washington DC: U. S. Bureau of Labor Statistics, mimeograph

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
lloyd_moulton(sugar, start="2018-12", end="2019-12",sigma=0.9)
lloyd_moulton(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

load_model *Loading the machine learning model from the disk*

Description

This function loads a list of machine learning model elements from the disk, i.e. the needed 8 files are read.

Usage

```
load_model(dir = "ML_model")
```

Arguments

`dir`	The name of the directory from which the machine learning model is to be loaded. The directory must be in the working directory.

Value

This function loads a list of ML model elements from the disk, i.e. the needed 8 files are read from the directory selected by `dir`. After loading the model it can be used for product classification by using `data_classifying` function.

Examples

```
#Setting a temporal directory as a working directory
## Not run: wd<-tempdir()
## Not run: setwd(wd)
#Building the model
## Not run: my.grid=list(eta=c(0.01,0.02,0.05),subsample=c(0.5))
## Not run: data_train<-dplyr::filter(dataCOICOP,dataCOICOP$time<=as.Date("2020-08-01"))
## Not run: data_test<-dplyr::filter(dataCOICOP,dataCOICOP$time>as.Date("2020-08-01"))
## Not run: ML<-model_classification(data_train,data_test,grid=my.grid,
indicators=c("prodID","unit","description"),key_words=c("milk"),rounds=50)
## End(Not run)
#Saving the model
## Not run: save_model(ML, dir="My_model")
#Loading the model
## Not run: ML_fromPC<-load_model("My_model")
#COICOP predicting
## Not run: data_classifying(ML_fromPC, data_test)
```

`lowe` *Calculating the bilateral Lowe price index*

Description

This function returns a value (or vector of values) of the bilateral Lowe price index.

Usage

```
lowe(data, start, end, base = start, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`base`	The prior period used in the Lowe price index formula (as character) limited to the year and month, e.g. "2020-01".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Lowe price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
lowe(sugar, start="2019-01", end="2020-01",base="2018-12")
lowe(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

marshall_edgeworth *Calculating the bilateral Marshall-Edgeworth price index*

Description

This function returns a value (or vector of values) of the bilateral Marshall-Edgeworth price index.

Usage

```
marshall_edgeworth(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Marshall-Edgeworth price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Marshall, A. (1887). *Remedies for Fluctuations of General Prices*. Contemporary Review, 51, 355-375.

Edgeworth, F. Y. (1887). *Measurement of Change in Value of Money I*. The first Memorandum presented to the British Association for the Advancement of Science; reprinted in Papers Relating to Political Economy, Vol. 1, New York, Burt Franklin, s. 1925.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
marshall_edgeworth(sugar, start="2018-12", end="2019-12")
marshall_edgeworth(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

matched

Providing values from the indicated column that occur simultaneously in the compared periods or in a given time interval.

Description

The function returns all values from the indicated column (defined by the `type` parameter) which occur simultaneously in the compared periods or in a given time interval.

Usage

```
matched(data, period1, period2, type = "prodID", interval = FALSE)
```

Arguments

`data`	The user's data frame. It must contain a column `time` (as Date in format: year-month-day, e.g. '2020-12-01') and also a column indicated by the `type` parameter.
`period1`	The first period (as character) limited to the year and month, e.g. "2019-03".
`period2`	The second period (as character) limited to the year and month, e.g. "2019-04".

type	This parameters defines the column which is used in the procedure. Possible values of the type parameter are: retID, prodID, codeIN, codeOUT or description.
interval	A logical parameter indicating whether the procedure is to work for the whole time period between period1 and period2 (then it is TRUE).

Value

The function returns all values from the indicated column (defined by the type parameter) which occur simultaneously in the compared periods or in a given time interval. Possible values of the type parameter are: retID, prodID, codeIN, codeOUT or description. If the interval parameter is set to FALSE, then the function compares only periods defined by period1 and period2. Otherwise the whole time period between period1 and period2 is considered.

Examples

```
matched(milk, period1="2018-12", period2="2019-12", interval=TRUE)
matched(milk, period1="2018-12", period2="2019-12", type="description")
```

---

matched_fig *Providing a matched_index() function dependant on time*

Description

The function provides a data frame or a figure presenting the matched_index function calculated for the column defined by the type parameter and for each month from the considered time interval

Usage

```
matched_fig(data, start, end, type = "prodID", fixedbase = TRUE, figure = TRUE)
```

Arguments

data	The user's data frame. It must contain a column time (as Date in format: year-month-day,e.g. '2020-12-01') and also a column indicated by the type parameter.
start	The base period (as character) limited to the year and month, e.g. "2019-03".
end	The research period (as character) limited to the year and month, e.g. "2019-04".
type	This parameter defines the column which is used in the procedure. Possible values of the type parameter are: retID, prodID, codeIN, codeOUT or description.
fixedbase	A logical parameter indicating whether the procedure is to work for subsequent months from the considered time interval (fixedbase=FALSE). Otherwise the month defined by start plays a role of fixed base month (fixedbase=TRUE)
figure	A logical parameter indicating whether the function returns a figure (TRUE) or a data frame (FALSE) with matched_index values.

Value

The function returns a data frame or a figure presenting the matched_index function calculated for the column defined by the type parameter and for each month from the considered time interval. The interval is set by start and end parameters. The returned object (data frame or figure) depends on the value of figure parameter. The returned values belong to [0,1].

Examples

```
matched_fig(milk, start="2018-12", end="2019-12")
matched_fig(milk, start="2018-12", end="2019-12", figure=FALSE)
```

---

matched_index *Providing the ratio of number of matched values from the indicated column to the number of all available values from this column*

Description

The function returns a ratio of number of values from the indicated column that occur simultaneously in the compared periods or in a given time interval to the number of all available values from the above-mentioned column (defined by the type parameter) at the same time.

Usage

```
matched_index(data, period1, period2, type = "prodID", interval = FALSE)
```

Arguments

data	The user's data frame. It must contain a column time (as Date in format: year-month-day,e.g. '2020-12-01') and also a column indicated by the type parameter.
period1	The first period (as character) limited to the year and month, e.g. "2019-03".
period2	The second period (as character) limited to the year and month, e.g. "2019-04".
type	This parameter defines the column which is used in the procedure. Possible values of the type parameter are: retID, prodID, codeIN, codeOUT or description.
interval	A logical parameter indicating whether the procedure is to work for the whole time period between period1 and period2 (then it is TRUE).

Value

The function returns a ratio of number of values from the indicated column that occur simultaneously in the compared periods or in a given time interval to the number of all available values from the above-mentioned column (defined by the type parameter) at the same time. Possible values of the type parameter are: retID, prodID or description. If the interval parameter is set to FALSE, then the function compares only periods defined by period1 and period2. Otherwise the whole time period between period1 and period2 is considered. The returned value belongs to [0,1].

Examples

```
matched_index(milk, period1="2018-12", period2="2019-12", interval=TRUE)
matched_index(milk, period1="2018-12", period2="2019-12", type="retID")
```

`milk` *Real data set on sold milk*

Description

A collection of scanner data on the sale of milk in one of Polish supermarkets in the period from December 2018 to August 2020

Usage

```
milk
```

Format

A data frame with 6 columns and 4281 rows. The used variables are as follows:

`time` - Dates of transactions (Year-Month-Day)

`prices` - Prices of sold products [PLN]

`quantities` - Quantities of sold products [liters]

`prodID` - Unique product codes (data set contains 67 different prodIDs)

`retID` - Unique codes identifying outlets/retailer sale points (data set contains 5 different retIDs)

`description` Descriptions of sold milk products (data set contains 6 different product descriptions)

`model_classification` *Building the machine learning model for product classification*

Description

This function provides a trained machine learning model to classify products into coicop groups. In addition, the function returns the characteristics of the model and figures describing the learning process.

Usage

```
model_classification(
  data_train = data.frame(),
  data_test = data.frame(),
  indicators = c(),
  key_words = c(),
  sensitivity = FALSE,
  p = 0.9,
  w = 0.2,
  rounds = 200,
  grid = list()
)
```

Arguments

data_train	Training data set for the model. This set must contain all the columns defined by the indicators parameter and the coicop column (with matched coicop groups to all products). If the key_words vector is non-empty, the set should also contain a description column. Ideally, the indicators should be of the numerical type. If the indicator is not of the numerical type, it will be converted to this type.
data_test	A test set that is used to validate the machine learning model. This set should have the same structure as the training set, but it is not obligatory. If the test set is not specified by the user then the test set is drawn from the training set (see p parameter).
indicators	A vector of column names to be considered in building a machine learning model.
key_words	A vector of keywords or phrases that will be recognized in the description column. For each such keyword and or phrase, a new binary variable (column) will be created and included in the machine model training process.
sensitivity	A logical parameter that indicates whether lowercase or uppercase letters are to be distinguished when the key_words vector is not empty.
p	A parameter related to creating the testing set, if it has not been specified by the user. The test set is then created on the basis of a coicop-balanced subsample of the training set. The size of this subsample is 100p percents of the training set size.
w	A parameter for determining the measure of choosing the optimal machine learning model. For each combination of parameters specified in the grid list, the error rate of the trained model is calculated on the basis of the error on the training set (error_L=1-accuracy_L) and the error on the testing set (error_T=1-accuracy_T). Final accuracy of the model is estimated as: w accuracy_L + (1-w) accuracy_T.
rounds	The maximum number of iterations during the training stage.
grid	The list of vectors of parameters which are taken into consideration during the Extreme Gradient Boosting training. The default value of this list is as follows: grid=list(eta=c(0.05,0.1,0.2),max_depth=c(6),min_child_weight=c(1),max_delta_st

The complete list of parameters for the used Tree Booster is available online here.

Value

In general, this function provides a trained machine learning model to classify products into coicop groups. In addition, the function returns the characteristics of the model and figures describing the learning process. The machine learning process is based on the XGBoost algorithm (from the XGBoost package) which is an implementation of gradient boosted decision trees designed for speed and performance. The function takes into account each combination of model parameters (specified by the grid list) and provides, inter alia, an optimally trained model (a model that minimizes the error rate calculated on the basis of a fixed value of the w parameter). After all, the function returns a list of the following objects: model - the optimally trained model; best_parameters - a set of parameters of the optimal model; indicators - a vector of all indicators used; key_words - a vector of all key words and phrases used; coicops - a dataframe with categorized COICOPs; sensitivity - a value of the used 'sensitivity' parameter; figure_training - a plot of the error levels calculated for the training set and the testing set during the learning process of the returned model (error = 1 - accuracy); figure_importance - a plot of the relative importance of the used indicators.

References

Tianqi Chen and Carlos Guestrin (2016). *XGBoost: A Scalable Tree Boosting System*. 22nd SIGKDD Conference on Knowledge Discovery and Data Mining.

Examples

```
my.grid=list(eta=c(0.01,0.02,0.05),subsample=c(0.5))
data_train<-dplyr::filter(dataCOICOP,dataCOICOP$time<=as.Date("2020-08-01"))
data_test<-dplyr::filter(dataCOICOP,dataCOICOP$time>as.Date("2020-08-01"))
ML<-model_classification(data_train,data_test,grid=my.grid,
indicators=c("prodID","unit","description"),key_words=c("milk"),rounds=50)
ML$best_parameters
ML$indicators
ML$figure_training
ML$figure_importance
```

---

paasche *Calculating the bilateral Paasche price index*

Description

This function returns a value (or vector of values) of the bilateral Paasche price index.

Usage

```
paasche(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Paasche price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Paasche, H. (1874). *Uber die Preisentwicklung der letzten Jahre nach den Hamburger Borsennotirungen*. Jahrbucher fur Nationalokonomie und Statistik, 12, 168-178.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
paasche(sugar, start="2018-12", end="2019-12")
paasche(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

palgrave *Calculating the bilateral Palgrave price index*

Description

This function returns a value (or vector of values) of the bilateral Palgrave price index.

Usage

```
palgrave(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Palgrave price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Palgrave, R. H. I. (1886). *Currency and Standard of Value in England, France and India and the Rates of Exchange Between these Countries.* Memorandum submitted to the Royal Commission on Depression of trade and Industry, Third Report, Appendix B, 312-390.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
palgrave(sugar, start="2018-12", end="2019-12")
palgrave(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

pqcor *Providing a correlation coefficient for price and quantity of sold products*

Description

The function returns correlation between price and quantity of sold products with given IDs.

Usage

```
pqcor(data, period, set = c(), figure = FALSE)
```

Arguments

data	The user's data frame. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character) with unique product IDs.
period	The time period (as character) limited to the year and month, e.g. "2019-03".
set	The set of unique product IDs to be used for determining correlation between price and quantity of sold products (see also data_matching). If the set is empty, the function works for all products being available in period.
figure	A logical parameter indicating whether the function returns a figure (TRUE) or a data frame (FALSE) with correlations between price and quantity of sold products.

Value

The function returns Pearson's correlation coefficient between price and quantity of products with given IDs and sold in period.

Examples

```
pqcor(milk, period="2019-03")
pqcor(milk, period="2019-03",figure=TRUE)
```

---

pqcor_fig *Providing correlations between price and quantity of sold products*

Description

The function returns Pearson's correlation coefficients between price and quantity of sold products with given IDs.

Usage

```
pqcor_fig(data, start, end, figure = TRUE, set = c())
```

Arguments

data	The user's data frame. It must contain columns: time (as Date in format: year-month-day, e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character) with unique product IDs.
start	The beginning of the considered time interval (as character) limited to the year and month, e.g. "2020-03".
end	The end of the considered time interval (as character) limited to the year and month, e.g. "2020-04".

figure	A logical parameter indicating whether the function returns a figure (TRUE) or a data frame (FALSE) with price-quantity correlations.
set	The set of unique product IDs to be used for determining correlation between prices and quantities of sold products (see also data_matching). If the set is empty, the function works for all products being available in period.

Value

The function returns Pearson's correlation coefficients between price and quantity of products with given IDs and sold in the time interval: <start,end>. Correlation coefficients are calculated for each month separately. Results are presented in tabular or graphical form depending on the figure parameter.

Examples

```
pqcor_fig(milk, start="2018-12", end="2019-12", figure=FALSE)
pqcor_fig(milk, start="2018-12", end="2019-12", figure=TRUE)
```

PriceIndices *The list of package functions and their demonstration*

Description

The **PriceIndices** package is a tool for Bilateral and Multilateral Price Index Calculations. A demonstration of package functions is here: README. The package documentation can be found HERE. The list of package functions is as follows:

Data sets in the package and generating artificial scanner data sets

dataMATCH

dataCOICOP

milk

sugar

coffee

dataU

generate

tindex

Functions for data processing

data_check

data_preparing

data_unit

data_norm

data_selecting

data_classifying

model_classification

save_model

load_model

data_matching

data_filtering

Functions providing dataset characteristics

available

matched

matched_index

matched_fig

prices

quantities

sales

sales_groups

sales_groups2

pqcor

pqcor_fig

dissimilarity_fig

Functions for bilateral unweighted price index calculation

bmw

carli

cswd

dutot

jevons

harmonic

Functions for bilateral weighted price index calculation

agmean

banajree

bialek

davies

drobisch

fisher

geary_khamis

geolaspeyres

geolowe

geopaasche

geoyoung

geohybrid

hybrid

laspeyres

lehr

lloyd_moulton

lowe

marshall_edgeworth

paasche

palgrave

sato_vartia

stuvel

tornqvist

vartia

walsh

young

Functions for chain price index calculation

chbmw

chcarli

chcswd

chdutot

chjevons

chharmonic

chagmean

chbanajree

chbialek

davies

chdrobisch

chfisher

chgeary_khamis

chgeolaspeyres

chgeolowe

chgeopaasche
chgeoyoung
chgeohybrid
chhybrid
chlaspeyres
chlehr
chlloyd_moulton
chlowe
chmarshall_edgeworth
chpaasche
chpalgrave
chsato_vartia
chstuvel
chtornqvist
chvartia
chwalsh
chyoung

Functions for multilateral price index calculation

ccdi
geks
geksj
geksw
geksl
wgeksl
gk
QU
tpd
SPQ

Functions for extending multilateral price indices by using splicing methods

ccdi_splice
geks_splice
geksj_splice
geksw_splice
geksl_splice
wgeksl_splice
gk_splice
tpd_splice

Functions for extending multilateral price indices by using the FBEW method

ccdi_fbew

geks_fbew

geksj_fbew

geksw_fbew

geksl_fbew

wgeksl_fbew

gk_fbew

tpd_fbew

Functions for extending multilateral price indices by using the FBMW method

ccdi_fbmw

geks_fbmw

geksj_fbmw

geksw_fbmw

geksl_fbmw

wgeksl_fbmw

gk_fbmw

tpd_fbmw

General functions for price index calculations

price_index

price_indices

final_index

final_index2

Functions for graphical comparison of price indices

compare_indices

compare_final_indices

prices *Providing prices (unit values) of sold products*

Description

The function returns prices (unit values) of sold products with given IDs.

Usage

```
prices(data, period, set = c())
```

Arguments

data	The user's data frame. It must contain columns: time (as Date in format: year-month-day, e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character) with unique product IDs.
period	The time period (as character) limited to the year and month, e.g. "2019-03".
set	The set of unique product IDs to be used for determining prices of sold products (see also data_matching). If the set is empty, the function returns prices of all products being available in period.

Value

The function analyzes the user's data frame and returns prices (unit value) of products with given ID and being sold in the time period indicated by the period parameter.

Examples

```
prices(milk, period="2019-06")
prices(milk, period="2019-12",set=c(400032, 71772, 82919))
```

price_index *A general function to compute a price index*

Description

This function returns a value or values of the selected price index.

Usage

```
price_index(
  data,
  start,
  end,
  formula = "fisher",
  window = 13,
  splice = "movement",
  base = start,
  sigma = 0.7,
  interval = FALSE
)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric) and prodID (as numeric, factor or character). A column quantities (as positive numeric) is also essential even if the selected index is an unweighted formula (unit values are calculated).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
formula	The character string indicating the price index formula is to be calculated. To see available options please use the link: PriceIndices.
window	The length of the time window if the multilateral index is selected (as positive integer: typically multilateral methods are based on the 13-month time window).
splice	A character string indicating the splicing method (if the multilateral splicing index is selected). Available options are: "movement", "window","half","mean" and their additional variants: "window_published", "half_published" and "mean_published".
base	The prior period used in the Young- or Lowe-type price indices (as character) limited to the year and month, e.g. "2020-01".
sigma	The elasticity of substitution parameter used in the Lloyed-Moulton and AG Mean indices (as numeric).
interval	A logical value indicating whether the function is to provide the price index comparing the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be presented (the fixed base month is defined by start).

Value

This function returns a value or values of the selected price index. If the interval parameter is set to TRUE then it returns a data frame with two columns: dates and index values. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

Examples

```
price_index(milk, start="2018-12", end="2020-02",formula="walsh",interval=FALSE)
price_index(milk, start="2018-12",end="2020-02",formula="tpd_splice",
splice="half",interval=TRUE)
```

`price_indices` *A very general function to compute one or more price indices*

Description

This function returns a value or values of the selected price indices.

Usage

```
price_indices(
  data,
  start,
  end,
  bilateral = c(),
  bindex = c(),
  base = c(),
  cesindex = c(),
  sigma = c(),
  simindex = c(),
  fbmulti = c(),
  fbwindow = c(),
  splicemulti = c(),
  splicewindow = c(),
  splice = c(),
  namebilateral = bilateral,
  namebindex = bindex,
  namecesindex = cesindex,
  namesimindex = simindex,
  namefbmulti = fbmulti,
  namesplicemulti = splicemulti,
  interval = FALSE
)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric) and `prodID` (as numeric, factor or character). A column `quantities` (as positive numeric) is also essential even if the selected index is an unweighted formula (unit values are calculated).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".

end	The research period (as character) limited to the year and month, e.g. "2020-04".
bilateral	A vector of character strings indicating bilateral price index formulas that are to be calculated. To see available options please use the link: PriceIndices.
bindex	A vector of character strings indicating Lowe- or Young-type price index formulas that are to be calculated. Available options are: young,geoyoung,lowe and geolowe.
base	The vector of prior periods used in the Young- or Lowe-type price indices. Each element of the vector (as character) must be limited to the year and month, e.g. "2020-01".
cesindex	A vector of character strings indicating CES price index formulas that are to be calculated. To see available options, please use the link: PriceIndices.
sigma	The vector of elasticity of substitution parameters used in the Lloyed-Moulton and AG Mean indices.
simindex	A vector of character strings indicating multilateral price index formulas based on relative price and quantity similarity that are to be calculated. To see available options, please use the link: PriceIndices.
fbmulti	A vector of character strings indicating multilateral price index formulas that are to be calculated. The available set of indices includes full-window multilateral indices or their FBEW and FBMW extensions.To see available options, please use the link: PriceIndices.
fbwindow	A vector of integers. Each element of the vector defines the length of the time window of the corresponding multilateral index (if it is selected by fbmulti).
splicemulti	A vector of character strings indicating multilateral price index formulas that are to be extended by using splicing methods. To see available options. please use the link: PriceIndices.
splicewindow	A vector of integers. Each element of the vector defines the length of the time window of the corresponding multilateral index (if it is selected by splicemulti).
splice	A vector of character strings. Each element of the vector indicates the splicing method is to be used for the corresponding multilateral index (if it is selected by splicemulti). Available values of vector elements are: "movement", "window","half","mean" and their additional variants: "window_published", "half_published" and "mean_published".
namebilateral	A vector of character strings describing names of bilateral price indices that are to be displayed. If this vector is empty, then default names are used.
namebindex	A vector of character strings describing names of Young- and/or Lowe-type price indices are to be displayed. If this vector is empty, then default names are used.
namecesindex	A vector of character strings describing names of CES price indices that are to be displayed. If this vector is empty, then default names are used.
namesimindex	A vector of character strings describing names of multilateral price index formulas based on relative price and quantity similarity that are to be displayed. If this vector is empty, then default names are used.
namefbmulti	A vector of character strings describing names of full-window multilateralindices or their FBEW and FBMW extensions that are to be displayed. If this vector is empty, then default names are used.

namesplicemulti
: A vector of character strings describing names of multilateral splice indices that are to be displayed. If this vector is empty, then default names are used.

interval
: A logical value indicating whether the function is to provide the price index comparing the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be presented (the fixed base month is defined by start).

Value

This general function returns a value or values of the selected price indices. If the interval parameter is set to TRUE, then it returns a data frame where its first column indicates dates and the remaining columns show corresponding values of all selected price indices. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

Examples

```
price_indices(milk, start="2018-12",end="2019-04",bilateral=c("jevons"),
fbmulti=c("tpd"),fbwindow=c(6),interval=TRUE)
price_indices(milk, start="2018-12", end="2019-05",
fbmulti=c("tpd","geks"),fbwindow=c(10,12),interval=TRUE)
```

QU *Calculating the quality adjusted unit value index (QU index)*

Description

This function returns a value of the quality adjusted unit value index (QU index) for a given set of adjustment factors.

Usage

```
QU(data, start, end, v)
```

Arguments

data
: The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).

start
: The base period (as character) limited to the year and month, e.g. "2020-03".

end
: The research period (as character) limited to the year and month, e.g. "2020-04".

v
: The data frame with adjustment factors for at least all matched prodIDs. It must contain two columns: prodID (as numeric or character) with unique product IDs and values (as positive numeric) with corresponding adjustment factors.

Value

This function returns a value of the quality adjusted unit value index (QU index) for a given set of adjustment factors (adjusted factors must be available for all matched prodIDs).

References

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
## Creating a data frame with artificial adjustment factors
## (random numbers from uniform distribution U[1,2])
prodID<-unique(milk$prodID)
values<-stats::runif(length(prodID),1,2)
v<-data.frame(prodID,values)
## Calculating the QU index for the created data frame 'v'
QU(milk, start="2018-12", end="2019-12", v)
```

---

quantities *Providing quantities of sold products*

Description

The function returns quantities of sold products with given IDs.

Usage

```
quantities(data, period, set = c())
```

Arguments

`data`	The user's data frame. It must contain columns: `time` (as Date in format: year-month-day, e.g. '2020-12-01'), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character) with unique product IDs.
`period`	The time period (as character) limited to the year and month, e.g. "2019-03".
`set`	The set of unique product IDs to be used for determining quantities of sold products (see also `data_matching`). If the `set` is empty, the function returns quantities of all products being available in `period`.

Value

The function analyzes the user's data frame and returns quantities of products with given `ID` and being sold in the time period indicated by the `period` parameter.

Examples

```
quantities(milk, period="2019-06")
quantities(milk, period="2019-12",set=c(400032, 71772, 82919))
```

sales *Providing values of product sales*

Description

The function returns values of sales of products with given IDs.

Usage

```
sales(data, period, set = c(), shares = FALSE, hist = FALSE)
```

Arguments

data	The user's data frame. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character) with unique product IDs.
period	The time period (as character) limited to the year and month, e.g. "2019-03".
set	The set of unique product IDs to be used for determining product sales values (see also data_matching). If the set is empty, then the function returns sale values of all products being available in period.
shares	A logical parameter indicating whether the function is to return shares of product sales.
hist	A logical parameter indicating whether the function is to return histogram of product sales.

Value

The function analyzes the user's data frame and returns values of sales of products with given IDs and being sold in time period indicated by the period parameter.

Examples

```
sales(milk, period="2019-06", shares=TRUE, hist=TRUE)
sales(milk, period="2019-12",set=unique(milk$prodID)[1])
```

sales_groups *Providing information about sales of products from one or more datasets*

Description

The function returns values of sales of products from one or more datasets or the corresponding barplot for these sales.

Usage

```
sales_groups(
  datasets = list(),
  start,
  end,
  shares = FALSE,
  barplot = FALSE,
  names = c()
)
```

Arguments

datasets	A list of user's data frames. Each data frame must contain columns: time (as Date in format: year-month-day, e.g. '2020-12-01'), prices (as positive numeric) and quantities (as positive numeric).
start	The beginning of the considered time interval (as character) limited to the year and month, e.g. "2020-03".
end	The end of the considered time interval (as character) limited to the year and month, e.g. "2020-04".
shares	A logical parameter indicating whether the function is to calculate shares of product sales
barplot	A logical parameter indicating whether the function is to return barplot for product sales.
names	A vector of characters describing product groups defined by datasets.

Value

The function returns values of sales of products from one or more datasets or the corresponding barplot for these sales (if barplot is TRUE). Alternatively, it calculates the sale shares (if shares is TRUE).

Examples

```
## Creating 3 subgroups of milk:
ctg<-unique(milk$description)
categories<-c(ctg[1],ctg[2],ctg[3])
```

```
milk1<-dplyr::filter(milk, milk$description==categories[1])
milk2<-dplyr::filter(milk, milk$description==categories[2])
milk3<-dplyr::filter(milk, milk$description==categories[3])
## Sample use of this function:
sales_groups(datasets=list(milk1,milk2,milk3),start="2019-04",end="2019-04",shares=TRUE)
sales_groups(datasets=list(milk1,milk2,milk3),start="2019-04",end="2019-07",
barplot=TRUE, names=categories)
```

`sales_groups2` *Providing information about sales of products*

Description

The function returns values of sales of products or the corresponding barplot for these sales.

Usage

```
sales_groups2(
  data = data.frame(),
  by,
  start,
  end,
  shares = FALSE,
  barplot = FALSE,
  names = c()
)
```

Arguments

`data`	The user's data frame with subgroups of sold products (see by parameter). The data frame must contain columns: `time` (as Date in format: year-month-day, e.g. '2020-12-01'), `prices` (as positive numeric) and `quantities` (as positive numeric). An additional column indicated via by parameter is also needed.
`by`	The column name indicating grouping variable, i.e. this column is used for creating subgroups of products.
`start`	The beginning of the considered time interval (as character) limited to the year and month, e.g. "2020-03".
`end`	The end of the considered time interval (as character) limited to the year and month, e.g. "2020-04".
`shares`	A logical parameter indicating whether the function is to calculate shares of product sales
`barplot`	A logical parameter indicating whether the function is to return barplot for product sales.
`names`	A vector of characters describing product groups defined by `datasets`.

Value

The function returns values of sales of products or the corresponding barplot for these sales (if `barplot` is TRUE). Alternatively, it calculates the sale shares (if `shares` is TRUE).

Examples

```
outlets<-as.character(unique(milk$retID))
sales_groups2(milk,by="retID",start="2019-04",end="2019-04",
shares=TRUE,barplot=TRUE,names=outlets)
```

`sato_vartia` *Calculating the bilateral Vartia-II (Sato-Vartia) price index*

Description

This function returns a value (or vector of values) of the bilateral Vartia-II (Sato-Vartia) price index.

Usage

```
sato_vartia(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Vartia-II (Sato-Vartia) price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Sato, K. (1976). *The Ideal Log-Change Index Number.* The Review of Economics and Statistics, 58(2), 223-228.

Vartia, Y. 0. (1976). *Ideal Log-Change Index Numbers* . Scandinavian Journal of Statistics 3(3), 121-126.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
sato_vartia(sugar, start="2018-12", end="2019-12")
sato_vartia(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

save_model *Saving the machine learning model on the disk*

Description

This function saves a list of machine learning model elements on the disk, i.e. the resulting 8 files are written.

Usage

```
save_model(model = list(), dir = "ML_model")
```

Arguments

model	A list of 8 elements which identify the previously built machine learning model (the list is obtained via the model_classification function).
dir	The name of the directory where the selected model should be saved. The directory with all necessary files will be created in the working directory.

Value

This function saves a list of ML model elements on the disk, i.e. the resulting 8 files are written into the new directory specified by dir. The list should be obtained previously using the model_classification function. After saving the model, it can be loaded at any time by using the load_model function.

Examples

```
#Setting a temporal directory as a working director
## Not run: wd<-tempdir()
## Not run: setwd(wd)
#Building the model
#Building the model
## Not run: my.grid=list(eta=c(0.01,0.02,0.05),subsample=c(0.5))
## Not run: data_train<-dplyr::filter(dataCOICOP,dataCOICOP$time<=as.Date("2020-08-01"))
## Not run: data_test<-dplyr::filter(dataCOICOP,dataCOICOP$time>as.Date("2020-08-01"))
## Not run: ML<-model_classification(data_train,data_test,grid=my.grid,
indicators=c("prodID","unit","description"),key_words=c("milk"),rounds=50)
## End(Not run)
#Saving the model
## Not run: save_model(ML, dir="My_model")
```

SPQ *Calculating the multilateral SPQ price index*

Description

This function returns a value of the multilateral SPQ price index which is based on the relative price and quantity dissimilarity measure.

Usage

```
SPQ(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. '2019-03'.
`end`	The research period (as character) limited to the year and month, e.g. '2019-07'.
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

This function returns a value of the multilateral SPQ price index which is based on the relative price and quantity dissimilarity measure (see `References`). If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices`,

final_index or final_index2. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or final_index2 function).

References

Diewert, E. (2020). *The Chain Drift Problem and Multilateral Indexes.* Chapter 6 in: Consumer Price Index Theory (draft)

Examples

```
SPQ(sugar, start="2018-12",end="2019-02")
SPQ(milk, start="2018-12",end="2019-12",interval=TRUE)
```

---

stuvel *Calculating the bilateral Stuvel price index*

Description

This function returns a value (or vector of values) of the bilateral Stuvel price index.

Usage

```
stuvel(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Stuvel price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Stuvel, G. (1957). *A New Index Number Formula.* Econometrica, 25, 123-131.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
stuvel(sugar, start="2018-12", end="2019-12")
stuvel(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

sugar *Real data set on sold sugar*

Description

A collection of scanner data on the sale of sugar in one of Polish supermarkets in the period from December 2017 to October 2020

Usage

```
sugar
```

Format

A data frame with 6 columns and 7666 rows. The used variables are as follows:

`time` - Dates of transactions (Year-Month-Day)

`prices` - Prices of sold products [PLN]

`quantities` - Quantities of sold products [kg]

`prodID` - Unique product codes (data set contains 11 different prodIDs)

`retID` - Unique codes identifying outlets/retailer sale points (data set contains 20 different retIDs)

`description` Descriptions of sold sugar products (data set contains 3 different product descriptions)

`tindex` *Calculating theoretical (expected) values of the unweighted price index*

Description

This function calculates the theoretical value of the unweighted price index for lognormally distributed prices.

Usage

```
tindex(pmi = c(), psigma = c(), start, ratio = TRUE)
```

Arguments

`pmi`	A numeric vector indicating `mi` parameters for lognormally distributed prices from the subsequent months.
`psigma`	A numeric vector indicating `sigma` parameters for lognormally distributed prices from the subsequent months.
`start`	The first period in the generated data frame (as character) limited to the year and month, e.g. '2019-12'.
`ratio`	A logical parameter indicating how we define the theoretical unweighted price index. If it is set to TRUE, then the resulting value is a ratio of expected price values from compared months; otherwise the resulting value is the expected value of the ratio of prices from compared months.

Value

This function calculates the theoretical value of the unweighted price index for lognormally distributed prices (the month defined by `start` parameter plays a role of the fixed base period). The characteristics for these lognormal distributions are set by `pmi` and `sigma` parameters. The `ratio` parameter allows to control the definition of resulting theoretical price index values. The function provides a data frame consisting of dates and corresponding expected values of the theoretical unweighted price index. The generated dataset is ready for further price index calculations.

Examples

```
tindex(pmi=c(1,1.2,1.3),psigma=c(0.1,0.2,0.15),start="2020-01")
tindex(pmi=c(1,1.2,1.3),psigma=c(0.1,0.2,0.15),start="2020-01",ratio=FALSE)
```

`tornqvist` *Calculating the bilateral Tornqvist price index*

Description

This function returns a value (or vector of values) of the bilateral Tornqvist price index.

Usage

```
tornqvist(data, start, end, interval = FALSE)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2020-03".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`interval`	A logical value indicating whether the function is to compare the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval `<start,end>` are considered and `start` defines the base period (`interval` is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Tornqvist price index depending on the `interval` parameter. If the `interval` parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Tornqvist, L. (1936). *The Bank of Finland's Consumption Price Index*. Bank of Finland Monthly Bulletin 10, 1-8.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
tornqvist(sugar, start="2018-12", end="2019-12")
tornqvist(milk, start="2018-12", end="2020-01", interval=TRUE)
```

tpd *Calculating the multilateral TPD price index*

Description

This function returns a value of the multilateral TPD (Time Product Dummy) price index.

Usage

```
tpd(data, start, end, wstart = start, window = 13)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
wstart	The beginning of the time interval (which is used by multilateral methods) limited to the year and month, e.g. "2020-01".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

Value

This function returns a value of the multilateral TPD price index which considers the time window defined by wstart and window parameters. It measures the price dynamics by comparing period end to period start (both start and end must be inside the considered time window). Please note that a Weighted Least Squares (WLS) regression is run with the expenditure shares in each period serving as weights.To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

de Haan, J. and F. Krsinich (2014). *Time Dummy Hedonic and Quality-Adjusted Unit Value Indexes: Do They Really Differ?* Paper presented at the Society for Economic Measurement Conference, 18-20 August 2014, Chicago, U.S.

Examples

```
tpd(milk, start="2019-01", end="2019-08",window=10)
tpd(milk, start="2018-12", end="2019-12")
```

tpd_fbew *Extending the multilateral TPD price index by using the FBEW method.*

Description

This function returns a value of the multilateral TPD price index (Time Product Dummy index) extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
tpd_fbew(data, start, end)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral TPD price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method. The FBEW method uses a time window with a fixed base month every year (December). The window is enlarged every month with one month in order to include information from a new month. The full window length (13 months) is reached in December of each year. The function measures the price dynamics between periods end and start. The month of the start parameter must be December. If the distance between end and start exceeds 13 months, then internal Decembers play a role of chain-linking months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

de Haan, J. and F. Krsinich (2014). *Time Dummy Hedonic and Quality-Adjusted Unit Value Indexes: Do They Really Differ?* Paper presented at the Society for Economic Measurement Conference, 18-20 August 2014, Chicago, U.S.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
tpd_fbew(milk, start="2018-12", end="2019-08")
```

tpd_fbmw *Extending the multilateral TPD price index by using the FBMW method.*

Description

This function returns a value of the multilateral TPD price index (Time Product Dummy index) extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
tpd_fbmw(data, start, end)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral TPD price index extended by using the FBMW (Fixed Base Moving Window) method. It measures the price dynamics between periods end and start and it uses a 13-month time window with a fixed base month taken as year(end)-1. If the distance between end and start exceeds 13 months, then internal Decembers play a role of chain-linking months. The month of the start parameter must be December. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

de Haan, J. and F. Krsinich (2014). *Time Dummy Hedonic and Quality-Adjusted Unit Value Indexes: Do They Really Differ?* Paper presented at the Society for Economic Measurement Conference, 18-20 August 2014, Chicago, U.S.

Lamboray, C.(2017). *The Geary Khamis index and the Lehr index: how much do they differ?* Paper presented at the 15th Ottawa Group meeting, 10-12 May 2017, Elville am Rhein, Germany.

Examples

```
tpd_fbmw(milk, start="2019-12", end="2020-04")
```

`tpd_splice` *Extending the multilateral TPD price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral TPD price index (Time Product Dummy index) extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see `References`).

Usage

```
tpd_splice(
  data,
  start,
  end,
  window = 13,
  splice = "movement",
  interval = FALSE
)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".
`window`	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
`splice`	A character string indicating the splicing method. Available options are: "movement", "window","half","mean", "window_published","half_published","mean_published".
`interval`	A logical value indicating whether the function is to provide the price index comparing the research period defined by `end` to the base period defined by `start` (then `interval` is set to FALSE) or all fixed base multilateral indices are to be presented (the fixed base month is defined by `start`).

Value

This function returns a value or values (depending on `interval` parameter) of the multilateral TPD price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on

published indices (see References). The time window starts in start and should consist of at least two months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Chessa, A. G. (2019). *A Comparison of Index Extension Methods for Multilateral Methods.* Paper presented at the 16th Meeting of the Ottawa Group on Price Indices, 8-10 May 2019, Rio de Janeiro, Brazil.

de Haan, J., van der Grient, H.A. (2011). *Eliminating chain drift in price indexes based on scanner data.* Journal of Econometrics, 161, 36-46.

de Haan, J. and F. Krsinich (2014). *Time Dummy Hedonic and Quality-Adjusted Unit Value Indexes: Do They Really Differ?* Paper presented at the Society for Economic Measurement Conference, 18-20 August 2014, Chicago, U.S.

Krsinich, F. (2014). *The FEWS Index: Fixed Effects with a Window Splice? Non-Revisable Quality-Adjusted Price Indices with No Characteristic Information.* Paper presented at the UNECE-ILO Meeting of the Group of Experts on Consumer Price Indices, 2-4 May 2016, Geneva, Switzerland.

de Haan, J.(2015). *A Framework for Large Scale Use of Scanner Data in the Dutch CPI.* Paper presented at the 14th Ottawa Group meeting, Tokyo, Japan.

Diewert, W.E., and Fox, K.J. (2017). *Substitution Bias in Multilateral Methods for CPI Construction using Scanner Data.* Discussion paper 17-02, Vancouver School of Economics, The University of British Columbia, Vancouver, Canada.

Examples

```
tpd_splice(milk, start="2018-12", end="2020-02",splice="half")
tpd_splice(milk, start="2018-12", end="2020-02",window=10,interval=TRUE)
```

---

vartia *Calculating the bilateral Vartia-I price index*

Description

This function returns a value (or vector of values) of the bilateral Vartia-I price index.

Usage

```
vartia(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Vartia-I price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Vartia, Y. 0. (1976). *Ideal Log-Change Index Numbers* . Scandinavian Journal of Statistics 3(3), 121-126.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
vartia(sugar, start="2018-12", end="2019-12")
vartia(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

walsh *Calculating the bilateral Walsh price index*

Description

This function returns a value (or vector of values) of the bilateral Walsh price index.

Usage

```
walsh(data, start, end, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Walsh price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Walsh, C. M. (1901). *The Measurement of General Exchange Value*. The MacMillan Company, New York.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Von der Lippe, P. (2007). *Index Theory and Price Statistics*. Peter Lang: Berlin, Germany.

Examples

```
walsh(sugar, start="2018-12", end="2019-12")
walsh(milk, start="2018-12", end="2020-01", interval=TRUE)
```

---

wgeksl *Calculating the multilateral WGEKS-L price index*

Description

This function returns a value of the multilateral weighted WGEKS-L price index (to be more precise: the weighted GEKS index based on the Laspeyres formula).

Usage

```
wgeksl(data, start, end, wstart = start, window = 13)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
wstart	The beginning of the time interval (which is used by multilateral methods) limited to the year and month, e.g. "2020-01".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).

Value

This function returns a value of the multilateral weighted WGEKS-L price index (to be more precise: the weighted GEKS index based on the Laspeyres formula) which considers the time window defined by wstart and window parameters. It measures the price dynamics by comparing period end to period start (both start and end must be inside the considered time window). To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Examples

```
wgeksl(milk, start="2019-01", end="2019-08",window=10)
wgeksl(milk, start="2018-12", end="2019-12")
```

---

wgeksl_fbew *Extending the multilateral weighted GEKS-L price index by using the FBEW method.*

Description

This function returns a value of the multilateral weighted GEKS-L price index extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
wgeksl_fbew(data, start, end)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral weighted GEKS-L price index (the weighted GEKS index based on the Laspeyres formula) extended by using the FBEW (Fixed Base Monthly Expanding Window) method. The FBEW method uses a time window with a fixed base month every year (December). The window is enlarged every month with one month in order to include information from a new month. The full window length (13 months) is reached in December of each year. The function measures the price dynamics between periods end and start. The month of the start parameter must be December. If the distance between end and start exceeds 13 months, then internal Decembers play a role of chain-linking months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Chessa, A.G. (2016). *A New Methodology for Processing Scanner Data in the Dutch CPI.* Eurona 1/2016, 49-69.

Examples

```
wgeksl_fbew(milk, start="2018-12", end="2019-08")
```

`wgeksl_fbmw` *Extending the multilateral weighted GEKS-L price index by using the FBMW method.*

Description

This function returns a value of the multilateral weighted GEKS-L price index extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
wgeksl_fbmw(data, start, end)
```

Arguments

`data`	The user's data frame with information about sold products. It must contain columns: `time` (as Date in format: year-month-day,e.g. '2020-12-01'), `prices` (as positive numeric), `quantities` (as positive numeric) and `prodID` (as numeric, factor or character).
`start`	The base period (as character) limited to the year and month, e.g. "2019-12".
`end`	The research period (as character) limited to the year and month, e.g. "2020-04".

Value

This function returns a value of the multilateral weighted GEKS-L price index (the GEKS index based on the Laspeyres formula) extended by using the FBMW (Fixed Base Moving Window) method. It measures the price dynamics between periods `end` and `start` and it uses a 13-month time window with a fixed base month taken as `year(end)-1`. If the distance between `end` and `start` exceeds 13 months, then internal Decembers play a role of chain-linking months. The month of the `start` parameter must be December. To get information about both price index values and corresponding dates, please see functions: `price_index`, `price_indices` or `final_index`. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the `final_index` or the `final_index2` function).

References

Gini, C. (1931). *On the Circular Test of Index Numbers.* Metron 9:9, 3-24.

Elteto, O., and Koves, P. (1964). *On a Problem of Index Number Computation Relating to International Comparisons.* Statisztikai Szemle 42, 507-518.

Szulc, B. (1983). *Linking Price Index Numbers.* In: Price Level Measurement, W. E. Diewert and C. Montmarquette (eds.), 537-566.

Lamboray, C.(2017). *The Geary Khamis index and the Lehr index: how much do they differ?* Paper presented at the 15th Ottawa Group meeting, 10-12 May 2017, Elville am Rhein, Germany.

Examples

```
wgeksl_fbmw(milk, start="2019-12", end="2020-04")
```

wgeksl_splice *Extending the multilateral weighted GEKS-L price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral weighted GEKS-L price index extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see References).

Usage

```
wgeksl_splice(
  data,
  start,
  end,
  window = 13,
  splice = "movement",
  interval = FALSE
)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2019-12".
end	The research period (as character) limited to the year and month, e.g. "2020-04".
window	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
splice	A character string indicating the splicing method. Available options are: "movement", "window","half","mean", "window_published","half_published","mean_published".
interval	A logical value indicating whether the function is to provide the price index comparing the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base multilateral indices are to be presented (the fixed base month is defined by start).

Value

This function returns a value or values (depending on interval parameter) of the multilateral weighted GEKS-L price index (the weighted GEKS index based on the Laspeyres formula) extended by using window splicing methods. Available splicing methods are: movement splice, window splice, half splice, mean splice and their additional variants: window splice on published

indices (WISP), half splice on published indices (HASP) and mean splice on published indices (see References). The time window starts in start and should consist of at least two months. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Chessa, A. G. (2019). *A Comparison of Index Extension Methods for Multilateral Methods.* Paper presented at the 16th Meeting of the Ottawa Group on Price Indices, 8-10 May 2019, Rio de Janeiro, Brazil.

de Haan, J., van der Grient, H.A. (2011). *Eliminating chain drift in price indexes based on scanner data.* Journal of Econometrics, 161, 36-46.

Krsinich, F. (2014). *The FEWS Index: Fixed Effects with a Window Splice? Non-Revisable Quality-Adjusted Price Indices with No Characteristic Information.* Paper presented at the UNECE-ILO Meeting of the Group of Experts on Consumer Price Indices, 2-4 May 2016, Geneva, Switzerland.

de Haan, J.(2015). *A Framework for Large Scale Use of Scanner Data in the Dutch CPI.* Paper presented at the 14th Ottawa Group meeting, Tokyo, Japan.

Diewert, W.E., and Fox, K.J. (2017). *Substitution Bias in Multilateral Methods for CPI Construction using Scanner Data.* Discussion paper 17-02, Vancouver School of Economics, The University of British Columbia, Vancouver, Canada.

Examples

```
wgeksl_splice(milk, start="2018-12", end="2020-02",splice="half")
wgeksl_splice(milk, start="2018-12", end="2020-02",window=10,interval=TRUE)
```

---

young *Calculating the bilateral Young price index*

Description

This function returns a value (or vector of values) of the bilateral Young price index.

Usage

```
young(data, start, end, base = start, interval = FALSE)
```

Arguments

data	The user's data frame with information about sold products. It must contain columns: time (as Date in format: year-month-day,e.g. '2020-12-01'), prices (as positive numeric), quantities (as positive numeric) and prodID (as numeric, factor or character).
start	The base period (as character) limited to the year and month, e.g. "2020-03".

end	The research period (as character) limited to the year and month, e.g. "2020-04".
base	The prior period used in the Young price index formula (as character) limited to the year and month, e.g. "2020-01"
interval	A logical value indicating whether the function is to compare the research period defined by end to the base period defined by start (then interval is set to FALSE) or all fixed base indices are to be calculated. In this latter case, all months from the time interval <start,end> are considered and start defines the base period (interval is set to TRUE).

Value

The function returns a value (or vector of values) of the bilateral Young price index depending on the interval parameter. If the interval parameter is set to TRUE, the function returns a vector of price index values without dates. To get information about both price index values and corresponding dates, please see functions: price_index, price_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (to consider these types of aggregating, please use the final_index or the final_index2 function).

References

Young, A. H. (1992). *Alternative Measures of Change in Real Output and Prices.* Survey of Current Business, 72, 32-48.

(2004). *Consumer Price Index Manual. Theory and practice*. ILO/IMF/OECD/UNECE/Eurostat/The World Bank, International Labour Office (ILO), Geneva.

Examples

```
young(sugar, start="2019-01", end="2020-01",base="2018-12")
young(milk, start="2018-12", end="2020-01", interval=TRUE)
```

Index