Package 'PriceIndices'

January 20, 2025

Title Calculating Bilateral and Multilateral Price Indexes

Version 0.2.2

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: Diewert and Fox (2020) <doi:10.1080/07350015.2020.1816176>, Białek (2019) <doi:10.2478/jos-2019-0014> or Białek (2020) <doi:10.2478/jos-2020-0037>.

Depends R (>= 3.5.0)

Imports lubridate (>= 1.7.4), dplyr (>= 0.8.3), ggplot2 (>= 3.2.0), reshape, reclin2, 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 2024-10-09 22:10:02 UTC

Contents

`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_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` 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_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 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)
```

---

bennet *Calculating the Bennet price and quantity indicators*

Description

This function returns the Bennet price and quantity indicators and optionally also the price and quantity contributions of individual products.

Usage

```
bennet(
  data,
  start,
  end,
  interval = FALSE,
  matched = FALSE,
  contributions = FALSE,
  prec = 2
)
```

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 parameter indicating whether calculations are to be made for the whole time interval (TRUE) or no (FALSE).
`matched`	A logical parameter indicating whether the matched sample approach is to be used (if yes, the parameter has the value TRUE).

contributions	A logical parameter indicating whether contributions of individual products are to be displayed. If it is TRUE, then contributions are calculated for the the base period start and the current period end.
prec	A numeric vector indicating precision, i.e. the number of decimal places for presenting results.

Value

This function returns the Bennet price and quantity indicators and optionally also the price and quantity contributions of individual products.

References

Bennet, T. L. (1920). *The Theory of Measurement of Changes in Cost of Living*. Journal of the Royal Statistical Society, 83, 455-462.

Białek, J. (2024). *The use of the Bennet indicators and their transitive versions for scanner data analysis*. Statistics in Transition new series, 25(3), 155-173.

Examples

```
bennet(milk, "2018-12", "2019-12", matched=TRUE, contributions=TRUE)
bennet(coffee, start="2018-12", end="2019-03", 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_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` 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_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` 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_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` 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_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` 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_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` 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_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` 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_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 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")
```

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_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 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_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 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_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 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_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` 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_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 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_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` 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_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` 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)
```

---

chdikhanov *Calculating the monthly chained Dikhanov price index*

Description

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

Usage

```
chdikhanov(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 Dikhanov 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_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 function).

References

Dikhanov, Y., (2024). *A New Elementary Index Number*. Paper presented at the 18th Meeting of the Ottawa Group on Price Indices, Ottawa, Canada.

Examples

```
chdikhanov(sugar, start="2018-12", end="2019-04")
chdikhanov(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_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 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_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 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_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 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_indices or final_index. The function does not take into account aggregating over outlets or product subgroups (see the final_index 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_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` 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_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` 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_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 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_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` 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_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` 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_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` 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_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` 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)
```

`chIQMp` *Calculating the monthly chained implicit quadratic mean of order r price index*

Description

This function returns a value (or vector of values) of the monthly chained implicit quadratic mean of order r price index.

Usage

```
chIQMp(data, start, end, r = 2, 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".
`r`	The real and non-zero parameter.
`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 implicit quadratic mean of order r price index - see CPI Manual (2004), Section 17.37, formula 17.32 (page 321).

References

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

Examples

```
chIQMp(sugar, start="2019-01", end="2020-01")
chIQMp(sugar, start="2019-01", end="2020-01", r=1.3, 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_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 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_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 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_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 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_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 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_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 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_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` 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_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` 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_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` 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)
```

`chQMp` *Calculating the monthly chained quadratic mean of order r price index*

Description

This function returns a value (or vector of values) of the monthly chained quadratic mean of order r price index.

Usage

```
chQMp(data, start, end, r = 2, 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".
`r`	The real and non-zero parameter.
`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 quadratic mean of order r price index - see CPI Manual (2004), Section 17.40, formula 17.35 (page 321).

References

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

Examples

```
chQMp(sugar, start="2019-01", end="2020-01")
chQMp(sugar, start="2019-01", end="2020-01", r=1.3, interval=TRUE)
```

---

chQMq *Calculating the monthly chained quadratic mean of order r quantity index*

Description

This function returns a value (or vector of values) of the monthly chained quadratic mean of order r quantity index.

Usage

```
chQMq(data, start, end, r = 2, 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".
`r`	The real and non-zero parameter.
`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 quadratic mean of order r quantity index - see CPI Manual (2004), Section 17.35, formula 17.30 (page 321).

References

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

Examples

```
chQMq(sugar, start="2019-01", end="2020-01")
chQMq(sugar, start="2019-01", end="2020-01", r=1.3, 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_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` 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_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` 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_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` 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_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` 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_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` 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_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` 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` *A 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_distances *Calculating distances between price indices*

Description

The function calculates distances between price indices

Usage

```
compare_distances(
  data = data.frame(),
  measure = "MAD",
  pp = TRUE,
  first = FALSE,
  prec = 3
)
```

Arguments

data	A data frame containg values of indices which are to be compared
measure	A parameter specifying what measure should be used to compare the indexes. Possible parameter values are: "MAD" (Mean Absolute Distance) or "RMSD" (Root Mean Square Distance).
pp	Logical parameter indicating whether the results are to be presented in percentage points (then pp = TRUE).
first	A logical parameter that determines whether the first row of the data frame is to be taken into account when calculating the distance between the indices (then first = TRUE). Usually, the first row concerns the index values for the base period - all indexes are then set to one.
prec	Parameter that determines how many decimal places are to be used in the presentation of results.

Value

The function calculates average distances between price indices and it returns a data frame with these values for each pair of price indices.

Examples

```
#Creating a data frame with unweighted bilateral index values
df<-price_indices(milk,
formula=c("jevons","dutot","carli"),
start="2018-12", end="2019-12",interval=TRUE)
#Calculating average distances between indices (in p.p)
compare_distances(df)
```

compare_indices_df *A function for graphical comparison of price indices*

Description

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

Usage

```
compare_indices_df(
  data,
  names = colnames(data)[2:length(colnames(data))],
  date_breaks = "1 month"
)
```

Arguments

data	The user's data frame with price index values. It must contain columns: time (as character in format: year-month, e.g. '2020-12') and columns with index values.
names	A vector of strings indicating names of indices which are to be used in the figure's legend.
date_breaks	A string giving the distance between breaks on the X axis like "1 month" (default value) or "4 months".

Value

This function returns a figure with plots of previously calculated indices (together with dates on X-axis and a corresponding legend). Indices must be provided as a data frame, where the the first column must includes dates limited to the year and month (e.g.: "2020-04").

Examples

```
df<-price_indices(milk, start = "2018-12", end = "2019-12",
formula=c("laspeyres", "fisher"), interval = TRUE)
compare_indices_df(df)
```

compare_indices_jk *A general function to compare indices by using the jackknife method*

Description

This function presents a comparison of selected indices obtained by using the jackknife method.

Usage

```
compare_indices_jk(
  data,
  start,
  end,
  by = "prodID",
  formula = c(),
  window = c(),
  splice = c(),
  base = c(),
  sigma = c(),
  r = c(),
  names = c(),
  title_iterations = c(),
  title_pseudovalues = c()
)
```

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".
by	A character string which indicates a column name for creating product subgroups (in the classical jackknife method by should indicate prodID). In each, successive repetition, the indicated price indexes are counted on the set of products reduced by the subset determined by the successive element of the column indicated by the by parameter.
formula	A vector of character strings indicating price index formulas that are to be calculated. To see available options please use the link: PriceIndices.
window	A vector of integers. Each element of the vector defines the length of the time window of the corresponding multilateral index.
splice	A vector of character strings. Each element of the vector indicates the splicing method is to be used for the corresponding multilateral index. Available values of vector elements are: "movement", "window","half","mean" and their additional variants: "window_published", "half_published" and "mean_published".
base	The vector of prior periods used in the Young- or Lowe-type price indices or hybrid/geohybrid index. Each element of the vector (as character) must be limited to the year and month, e.g. "2020-01".
sigma	The vector of elasticity of substitution parameters used in the Lloyed-Moulton, AG Mean or GEKS-LM indices (as numeric).
r	The vector of non-zero parameters used in the quadratic mean of order r quantity / price index or in the GEKS-QM index (as numeric).

`names` A vector of strings indicating names of indices which are to be used in the resulting data frame.

`title_iterations`
A character string indicating a title of the created box-plot for iteration index values.

`title_pseudovalues`
A character string indicating a title of the created box-plot for obtained (jackknife) index pseudovalues.

Value

This function presents a comparison of selected indices obtained by using the jackknife method. In particular, it returns a list with four elements: `iterations`, which is a data frame with basic characteristics of the calculated iteration index values (means, standard deviations, coefficients of variation and results for all sample), `pseudovalues`, which is a data frame with basic characteristics of the calculated index pseudovalues obtained in the jackknife procedure (i.e. the jackknife estimators and their standard deviations and coefficients of variation), `figure_iterations` which presents a box-plot for the calculated iteration index values, and `figure_pseudovalues` which presents a box-plot for the calculated index pseudovalues obtained in the jackknife procedure.

References

Quenouille, M.H. (1956). *Notes on bias in estimation*. Biometrika, 43 (3–4), 353–360

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

Examples

```
milk.<-dplyr::filter(milk, milk$prodID %in%
sample(unique(milk$prodID),4))
#creating a list with jackknife results
comparison<-compare_indices_jk(milk.,
formula=c("jevons","fisher"),
start="2018-12",
end="2019-12",
names=c("Jevons","Fisher"),
title_iterations="Box-plots for iteration values (milk products)",
title_pseudovalues="Box-plots for pseudovalues (milk products)")
#displaying results
comparison$iterations
comparison$pseudovalues
comparison$figure_iterations
comparison$figure_pseudovalues
```

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

Description

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

Usage

```
compare_indices_list(data = list(), names = c(), date_breaks = "1 month")
```

Arguments

data	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.
date_breaks	A string giving the distance between breaks on the X axis like "1 month" (default value) or "4 months".

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 a list of data frames (see data parameter).

Examples

```
## Caluclating two indices by using two different package functions:
index1<-final_index(data=milk, start="2018-12",
end="2019-12",formula="walsh",interval=TRUE)
index2<-price_indices(milk,start="2018-12", end="2019-12",
formula="geks",window=13,interval=TRUE)
## Graphical comparison of these two indices
compare_indices_list(data=list(index1,index2),
names=c("Walsh index", "GEKS index"))
```

`compare_to_target` *Calculating distances between considered price indices and the target price index*

Description

The function calculates distances between considered price indices and the target price index

Usage

```
compare_to_target(
  data = data.frame(),
  target,
  measure = "MAD",
  pp = TRUE,
  first = FALSE,
  prec = 3
)
```

Arguments

`data`	A data frame containg values of indices which are to be compared to the target price index
`target`	A data frame or a vector containg values of the target price index
`measure`	A parameter specifying what measure should be used to compare indices. Possible parameter values are: "MAD" (Mean Absolute Distance) or "RMSD" (Root Mean Square Distance).
`pp`	Logical parameter indicating whether the results are to be presented in percentage points (then pp = TRUE).
`first`	A logical parameter that determines whether the first row of the data frame and the first row of the 'target' data frame (or its first element if it is a vector) are to be taken into account when calculating the distance between the indices (then `first` = TRUE). Usually, the first row concerns the index values for the base period - all indexes are then set to one.
`prec`	Parameter that determines how many decimal places are to be used in the presentation of results.

Value

The function calculates average distances between considered price indices and the target price index and it returns a data frame with: average distances on the basis of all values of compared indices ('distance' column), average semi-distances on the basis of values of compared indices which overestimate the target index values ('distance_upper' column) and average semi-distances on the basis of values of compared indices which underestimate the target index values ('distance_lower' column).

Examples

```
#Creating a data frame with example bilateral indices
df<-price_indices(milk,
formula=c("jevons","laspeyres","paasche","walsh"),
start="2018-12",end="2019-12",interval=TRUE)
#Calculating the target Fisher price index
target_index<-fisher(milk,start="2018-12",end="2019-12",interval=TRUE)
#Calculating average distances between considered indices and the Fisher index (in p.p)
compare_to_target(df,target=target_index)
```

---

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_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` 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)
```

`dataAGGR` *A small artificial scanner data set for a demonstration of data aggregation*

Description

A collection of artificial scanner data on milk products sold in three different months

Usage

```
dataAGGR
```

Format

A data frame with 6 columns and 9 rows. The used variables are as follows:

`time` - Dates of transactions (Year-Month-Day: 4 different dates)

`prices` - Prices of sold products [PLN]

`quantities` - Quantities of sold products [l]

`prodID` - Retailer product codes (3 prodIDs)

`retID` - Unique codes identifying outlets/retailer sale points (4 retIDs)

`description` Descriptions of sold products (two subgroups: goat milk, powdered milk)

dataCOICOP *A real scanner data set for the product classification*

Description

A collection of real scanner data on the sale of milk products sold in a period: Dec, 2020 - Feb, 2022.

Usage

```
dataCOICOP
```

Format

A data frame with 10 columns and 139600 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

`description` - Descriptions of sold products (original: in Polish)

`codeIN` - Retailer product codes

`retID` - Unique codes identifying outlets/retailer sale points

`grammage` - Product grammages

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

`category` - Product categories (in English) corresponding to COICOP 6 levels

`coicop6` - Identifiers of local COICOP 6 groups (6 groups)

dataMATCH *An artificial scanner data set for product matching*

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 *An 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_aggregating *Aggregating the user's data frame*

Description

The function aggregates the user's data frame over time and optionally over outlets.

Usage

```
data_aggregating(data, join_outlets = TRUE)
```

Arguments

data	The user's data frame.
join_outlets	A logical value indicating whether the data aggregation over outlets should be also done.

Value

The function aggregates the user's data frame over time and/or over outlets. Consequently, we obtain monthly data, where the unit value is calculated instead of a price for each prodID observed in each month (the time column gets the Date format: "Year-Month-01"). If the parameter join_outlets is TRUE, then the function also performs aggregation over outlets (retIDs) and the retID column is removed from the data frame. The main advantage of using this function is the ability to reduce the size of the data frame and the time needed to calculate the price index. Please note, that unnecessary columns are removed (e.g. description).

Examples

```
#Example 1
data_aggregating(dataAGGR,join_outlets = FALSE)
data_aggregating(dataAGGR,join_outlets = TRUE)
#Example 2 (data frame reduction)
nrow(milk)
nrow(data_aggregating(milk))
```

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 classes via the machine learning model*

Description

This function predicts product class 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. class_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(0.5,0.8))
data_train<-dplyr::filter(dataCOICOP,dataCOICOP$time<=as.Date("2021-10-01"))
data_test<-dplyr::filter(dataCOICOP,dataCOICOP$time==as.Date("2021-11-01"))
ML<-model_classification(data_train,data_test,class="coicop6",grid=my.grid,
indicators=c("description","codeIN", "grammage"),key_words=c("uht"),rounds=60)
#Data classification
data_classifying(ML, data_test)
```

---

data_DOWN_UP_SIZED *An artificial data set on sold coffee*

Description

A collection of scanner data on the sale of coffee in the period from January 2024 to February 2024

Usage

```
data_DOWN_UP_SIZED
```

Format

A data frame with 6 columns and 51 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 product codes (retaler product codes)

`codeOUT` - Unique external product codes (e.g. GTIN, EAN, SKU)

`description` - Descriptions of sold coffee products

---

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 drop in sales value (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_imputing` *Imputing missing and (optionally) zero prices.*

Description

This function imputes missing prices and (optionally) zero prices by using carry forward/backward prices.

Usage

```
data_imputing(data, start, end, zero_prices = TRUE, outlets = 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 numeric), `quantities` (as numeric - for future calculations) and `prodID` (as numeric, factor or character). A column `retID` (as factor, character or numeric) is also needed if the User wants to impute prices over outlets.
`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".
`zero_prices`	A logical parameter indicating whether zero prices are to be imputed too (then it is set to TRUE).
`outlets`	A logical parameter indicating whether imputations are to be done for each outlet separately (then it is set to TRUE).

Value

This function imputes missing prices (unit values) and (optionally) zero prices by using carry forward/backward prices. The imputation can be done for each outlet separately or for aggragated data (see the `outlets` parameter). If a missing product has a previous price then that previous price is carried forward until the next real observation. If there is no previous price then the next real observation is found and carried backward. The quantities for imputed prices are set to zeros. The function returns a data frame (monthly aggregated) which is ready for price index calculations.

Examples

```
# Creating a small data set with zero prices:
time.<-c("2018-12-01","2019-01-01")
time<-as.Date(c(time., time.))
p1<-c(0,23)
p2<-c(14,0)
q1<-c(15,25)
q2<-c(44,79)
quantities<-c(q1,q2)
prices<-c(p1,p2)
prodID<-c(1,1,2,2)
my_data<-data.frame(time, prices, quantities, prodID)
# Price imputing:
data_imputing(my_data, start="2018-12", end="2019-01",
zero_prices=TRUE, outlets=FALSE)

# Preparing a data set with zero and missing prices:
dataMATCH$prodID<-dataMATCH$codeIN
data<-dplyr::select(dataMATCH, time, prices, quantities, prodID, retID)
set1<-data[1:5,]
set1$prices<-0
set2<-data[6:30,]
df<-rbind(set1, set2)
# Price imputing:
data_imputing(df, start="2018-12", end="2019-03",
zero_prices=TRUE, outlets=TRUE)
```

---

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 similarity measure when comparing labels (its value must belong to the interval [0,1]). Two labels are treated as similar enough if their Jaro-Winkler similarity 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 similarity 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 similarity 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 numeric or 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(),
  zero_prices = FALSE,
  zero_quantities = TRUE
)
```

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 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).

zero_prices — A logical parameter indicating whether zero prices are to be acceptable.

zero_quantities — A logical parameter indicating whether zero quantities are to be acceptable.

Value

The resulting data frame is free from: missing values, negative prices (if zero_prices is set to TRUE), zero or negative prices (if zero_prices is set to FALSE), negative quantities (if zero_quantities is set to TRUE) and zero and negative quantities (if zero_prices is set to FALSE). 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="coicop6")
```

`data_reducing` *Reducing products*

Description

The function returns a reduced data set, i.e. a data set containing sufficiently numerous matched products in the indicated groups. The input data set (data frame) must contain matched products over time, i.e. it must contain the `prodID` column (as numeric, factor or character), or product descriptions, i.e. it must contain the `description` column (as character).

Usage

```
data_reducing(
  data,
  start,
  end,
  type = "prodID",
  minN = 2,
  outlets = FALSE,
  by = c(),
  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') and, depending on next parameter values, columns: `prodID` or `description`, and `retID`.
`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".
`type`	This parameter indicates whether group counts are determined by different matched prodIDs (in which case the parameter has the value 'prodID') or different matched descriptions (in which case the parameter has the value 'description').
`minN`	This parameter determines the minimum size of matched products in groups.
`outlets`	This parameter determines whether grouping is to be done for each outlet separately. If so (if it is `TRUE`), the data set must contain a column identifying the outlets (`retID`).
`by`	This parameter specifies the name of the grouping column (as character).

interval	A logical value indicating whether the reducing process concerns only two periods defined by start and end parameters (then the interval is set to FALSE) or whether that function is to reduce products sold during the whole time interval <start, end>.

Value

The function returns a reduced data set, i.e. a data set containing sufficiently numerous matched products in the indicated groups. For each product group created and for selected periods, the procedure checks that the count of identical prodIDs (or identical product descriptions, which does not necessarily mean the same thing) is at least equal to minN. If it is not, such products are eliminated from the data set. The function performs the check either only for the base and current period (in which case the interval parameter is FALSE) or also for all intermediate months (in which case the interval parameter is TRUE). If the user wants to perform this check for each outlet separately, then the outlets parameter should be set to TRUE.

Examples

```
data_reducing(sugar, start="2018-12", end="2019-12",by="description", minN=5)
```

---

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")
```

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 (e.g. "ml|g|kg")
multiplication	A sign of the multiplication used in product descriptions (e.g. "x")

Value

The function returns the user's data frame with two additional columns: grammage and unit. 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: '2x50 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_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 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)
```

dikhanov *Calculating the unweighted Dikhanov price index*

Description

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

Usage

```
dikhanov(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 Dikhanov 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_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 function).

References

Dikhanov, Y., (2024). *A New Elementary Index Number*. Paper presented at the 18th Meeting of the Ottawa Group on Price Indices, Ottawa, Canada.

Examples

```
dikhanov(sugar, start="2018-12", end="2019-12")
dikhanov(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,
  date_breaks = "1 month"
)
```

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.
`date_breaks`	A string giving the distance between breaks on the X axis like "1 month" (default value) or "4 months".

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_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 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_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` 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)
```

---

elasticity *Calculating the elasticity of substitution*

Description

This function returns a value of the elasticity of substitution

Usage

```
elasticity(
  data,
  start,
  end,
  method = "lm",
  left = -10,
  right = 10,
  precision = 1e-06
)
```

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".
method	The index formula for which the CES index will be equated to calculate the elasticity. Acceptable options are lm, f, t, w and sv.
left	The beginning of an interval for estimation of the elasticity of substitution (its default value is -10).
right	The end of an interval for estimation of the elasticity of substitution (its default value is 10).
precision	The precision of estimation (a 'stop' condition for the procedure). A default value of this parameter is 0.000001.

Value

This function returns a value of the elasticity of substitution. If the `method` parameter is set to `lm`, the procedure of estimation solves the equation: LM(sigma)-CW(sigma)=0 numerically, where LM denotes the Lloyd-Moulton price index, the CW denotes a current weight counterpart of the Lloyd-Moulton price index, and sigma is the elasticity of substitution parameter, which is estimated. If the `method` parameter is set to `f`, the Fisher price index formula is used instead of the CW price index. If the `method` parameter is set to `t`, the Tornqvist price index formula is used instead of the CW price index. If the `method` parameter is set to `w`, the Walsh price index formula is used instead of the CW price index. If the `method` parameter is set to `sv`, the Sato-Vartia price index formula is used instead of the CW price index.The procedure continues until the absolute value of this difference is greater than the value of the 'precision' parameter.

References

de Haan, J., Balk, B.M., Hansen, C.B. (2010). *Retrospective Approximations of Superlative Price Indexes for Years Where Expenditure Data Is Unavailable.* In: Biggeri, L., Ferrari, G. (eds) Price Indexes in Time and Space. Contributions to Statistics. Physica-Verlag HD.

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

Examples

```
elasticity(coffee, start = "2018-12", end = "2019-01")
elasticity(coffee, start = "2018-12", end = "2019-01", method = "f")
elasticity(coffee, start = "2018-12", end = "2019-01", method = "sv")
```

---

elasticity_fig *Presenting elasticities of substitution for time interval*

Description

The function provides a data frame or a figure presenting elasticities of substitution calculated for time interval.

Usage

```
elasticity_fig(
  data,
  start,
  end,
  method = c("lm"),
  fixedbase = TRUE,
  figure = TRUE,
  date_breaks = "1 month",
  names = c(),
  left = -10,
  right = 10,
```

```
  precision = 1e-06
)
```

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".
method	A vector indicating index formulas for which the CES index will be equated to calculate the elasticity. Acceptable options are lm, f, t, w and sv or their combinations.
fixedbase	A logical parameter indicating whether the procedure is to work for subsequent months from the considered time interval (fixedbase=FALSE). Otherwise the period 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 values of elasticity of substitution.
date_breaks	A string giving the distance between breaks on the X axis like "1 month" (default value) or "4 months".
names	A character string indicating names of indices used for elasticity approximation (see the method parameter).
left	The beginning of an interval for estimation of each elasticity of substitution (its default value is -10)
right	The end of an interval for estimation of each elasticity of substitution (its default value is 10)
precision	The precision of estimation (a 'stop' condition for the procedure). A default value of this parameter is 0.000001.

Value

The function provides a data frame or a figure presenting elasticities of substitution calculated for time interval (see the figure parameter). The elasticities of substitution can be calculated for subsequent months or for a fixed base month (see the start parameter) and rest of months from the given time interval (it depends on the fixedbase parameter). The above-mentioned parameters for compared months are calculated by using the elasticity function.

References

de Haan, J., Balk, B.M., Hansen, C.B. (2010). *Retrospective Approximations of Superlative Price Indexes for Years Where Expenditure Data Is Unavailable.* In: Biggeri, L., Ferrari, G. (eds) Price Indexes in Time and Space. Contributions to Statistics. Physica-Verlag HD.

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

Examples

```
elasticity_fig (milk,start="2018-12",end="2019-04",figure=TRUE,
method=c("lm","f"),names=c("LM","Fisher"))
elasticity_fig (milk,start="2018-12",end="2019-06",figure=FALSE)
```

`expenditures` *Providing expenditures of sold products*

Description

The function returns expenditures of sold products with given IDs.

Usage

```
expenditures(data, period, set = c(), ID = 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'), `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 expenditures of sold products (see also `data_matching`). If the `set` is empty, the function returns quantities of all products being available in `period`.
`ID`	A logical parameter indicating whether a data frame with prodIDs and quantities should be returned.

Value

The function analyzes the user's data frame and returns expenditures of products with given `ID` and being sold in the time period indicated by the `period` parameter. Please note that the function returns the expenditure values for sorted prodIDs and in the absence of a given prodID in the data set, the function returns nothing (it does not return zero). If the ID parameter is set to TRUE then the function returns a data frame with columns: `by` (IDs of products) and `expend` (expenditures of products).

Examples

```
expenditures(milk, period="2019-06")
expenditures(milk, period="2019-12", set=c(400032, 82919), ID=TRUE)
```

final_index *A general function to compute a final price index*

Description

This function returns a value (or values) of the selected final price index for the selected type of aggregation of partial results.

Usage

```
final_index(
  data = data.frame(),
  start = c(),
  end = c(),
  formula = c(),
  window = c(),
  splice = c(),
  base = c(),
  sigma = c(),
  r = c(),
  outlets = FALSE,
  groups = FALSE,
  by = c(),
  aggr = "fisher",
  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 retID (as numeric, factor or character) is also essential if the aggregation over outlets is considered. A column with grouping variable (as numeric, factor or character - indicated by the by parameter) is essential if the aggregation over product subgroups is considered.
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, AG Mean or GEKS-LM indices (as numeric).
r	The non-zero parameter used in the quadratic mean of order r quantity / price index or in the GEKS-QM index (as numeric).
outlets	A logical parameter indicating whether the aggregation over outlets (defined by retID column) should be done.
groups	A logical parameter indicating whether the aggregation over product subgroups (indicated by 'by' parameter) should be done.
by	A character string which indicates a column name for creating product subgroups.
aggr	The formula used for aggregating partial index results (available values are: "arithmetic", "geometric", "laspeyres", "paasche", "fisher", "tornqvist").
interval	A logical value indicating whether the function is to provide price indices 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 final price index for the selected type of aggregation of partial results. 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.

Examples

```
final_index(coffee, start = "2018-12", end = "2019-12",
         formula = "fisher", groups = TRUE, outlets = FALSE,
         aggr = "tornqvist", by = "description")
final_index(milk, start = "2018-12", end = "2019-12",
         formula = "fisher", groups = TRUE, outlets = TRUE,
         aggr = "laspeyres", by = "description",
         interval = 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_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 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_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 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_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 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")
```

`geksaqi` *Calculating the multilateral GEKS-AQI price index*

Description

This function returns a value of the multilateral GEKS-AQI price index (to be more precise: the GEKS index based on the asynchronous quality adjusted price index formula).

Usage

```
geksaqi(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-AQI price index (to be more precise: the GEKS index based on the asynchronous quality adjusted price index 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`geksaqi_fbew` *Extending the multilateral GEKS-AQI price index by using the FBEW method.*

Description

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

Usage

```
geksaqi_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-AQI price index (the GEKS index based on the asynchronous quality adjusted price index 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`geksaqi_fbmw` *Extending the multilateral GEKS-AQI price index by using the FBMW method.*

Description

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

Usage

```
geksaqi_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-AQI price index (the GEKS index based on the asynchronous quality adjusted price index 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`geksaqi_splice` *Extending the multilateral GEKS-AQI price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-AQI 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

```
geksaqi_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-AQI price index (the GEKS index based on the asynchronous quality adjusted price index 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

```
geksaqi_splice(milk, start="2018-12", end="2020-02",splice="half")
```

geksaqu *Calculating the multilateral GEKS-AQU price index*

Description

This function returns a value of the multilateral GEKS-AQU price index (to be more precise: the GEKS index based on the asynchronous quality adjusted unit value formula).

Usage

```
geksaqu(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-AQU price index (to be more precise: the GEKS index based on the asynchronous quality adjusted unit value 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`geksaqu_fbew` *Extending the multilateral GEKS-AQU price index by using the FBEW method.*

Description

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

Usage

```
geksaqu_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-AQU price index (the GEKS index based on the asynchronous quality adjusted unit value 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`geksaqu_fbmw` *Extending the multilateral GEKS-AQU price index by using the FBMW method.*

Description

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

Usage

```
geksaqu_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-AQU price index (the GEKS index based on the asynchronous quality adjusted unit value 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`geksaqu_splice` *Extending the multilateral GEKS-AQU price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-AQU 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

```
geksaqu_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-AQU price index (the GEKS index based on the asynchronous quality adjusted unit value 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

```
geksaqu_splice(milk, start="2018-12", end="2020-02",splice="half")
```

geksgaqi *Calculating the multilateral GEKS-GAQI price index*

Description

This function returns a value of the multilateral GEKS-GAQI price index (to be more precise: the GEKS index based on the geometric asynchronous quality adjusted price index formula).

Usage

```
geksgaqi(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-GAQI price index (to be more precise: the GEKS index based on the geometric asynchronous quality adjusted price index 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_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 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

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

`geksgaqi_fbew` *Extending the multilateral GEKS-GAQI price index by using the FBEW method.*

Description

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

Usage

```
geksgaqi_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-GAQI price index (the GEKS index based on the geometric asynchronous quality adjusted price index 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_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` 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

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

`geksgaqi_fbmw` *Extending the multilateral GEKS-GAQI price index by using the FBMW method.*

Description

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

Usage

```
geksgaqi_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-GAQI price index (the GEKS index based on the geometric asynchronous quality adjusted price index 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_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` 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

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

`geksgaqi_splice` *Extending the multilateral GEKS-GAQI price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-GAQI 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

```
geksgaqi_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-GAQI price index (the GEKS index based on the geometric asynchronous quality adjusted price index 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Examples

```
geksgaqi_splice(milk, start="2018-12", end="2020-01",window=10)
```

`geksgl` *Calculating the multilateral GEKS-GL price index*

Description

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

Usage

```
geksgl(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-GL price index (to be more precise: the GEKS index based on the geometric 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

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

`geksgl_fbew` *Extending the multilateral GEKS-GL price index by using the FBEW method.*

Description

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

Usage

```
geksgl_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-GL price index (the GEKS index based on the geometric 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

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

`geksgl_fbmw` *Extending the multilateral GEKS-GL price index by using the FBMW method.*

Description

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

Usage

```
geksgl_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-GL price index (the GEKS index based on the geometric 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

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

`geksgl_splice` *Extending the multilateral GEKS-GL price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-GL 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

```
geksgl_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-GL price index (the GEKS index based on the geometric 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

```
geksgl_splice(milk, start="2018-12", end="2020-02",splice="half")
```

`geksiqm` *Calculating the multilateral GEKS-IQM price index*

Description

This function returns a value of the multilateral GEKS-IQM price index (to be more precise: the GEKS index based on the implicit quadratic mean of order r price index IQMp).

Usage

```
geksiqm(data, start, end, r = 2, 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".
`r`	The real and non-zero parameter used in the implicit quadratic mean of order r price index.
`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-IQM price index (to be more precise: the GEKS index based on the the implicit quadratic mean of order r price index IQMp) 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_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` 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.

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

Examples

```
geksiqm(milk, start="2019-01", end="2019-08",window=10)
geksiqm(milk, start="2018-12", end="2019-12", r=1.6)
```

`geksiqm_fbew` *Extending the multilateral GEKS-IQM price index by using the FBEW method.*

Description

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

Usage

```
geksiqm_fbew(data, start, end, r)
```

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".
`r`	The real and non-zero parameter used in the implicit quadratic mean of order r price index.

Value

This function returns a value of the multilateral GEKS-IQM 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_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` 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

```
geksiqm_fbew(milk, start="2018-12", end="2019-08", r=1.2)
```

`geksiqm_fbmw` *Extending the multilateral GEKS-IQM price index by using the FBMW method.*

Description

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

Usage

```
geksiqm_fbmw(data, start, end, r)
```

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".
`r`	The real and non-zero parameter used in the implicit quadratic mean of order r price index.

Value

This function returns a value of the multilateral GEKS-IQM 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_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` 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

```
geksiqm_fbmw(milk, start="2019-12", end="2020-04", r=1.6)
```

`geksiqm_splice` *Extending the multilateral GEKS-IQM price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-IQM 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

```
geksiqm_splice(
  data,
  start,
  end,
  r = 2,
  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".

r	The real and non-zero parameter used in the implicit quadratic mean of order r price index.
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-IQM 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_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 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

```
geksiqm_splice(milk, start="2018-12", end="2020-02", r=0.8, splice="half")
```

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_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 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_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` 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_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` 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_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 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")
```

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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

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

`gekslm` *Calculating the multilateral GEKS-LM price index*

Description

This function returns a value of the multilateral GEKS-LM price index (to be more precise: the GEKS index based on the Lloyd-Moulton price index).

Usage

```
gekslm(data, start, end, sigma = 0.7, 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".
`sigma`	The elasticity of substitution (a parameter used in the Lloyd-Moulton index formula).
`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-LM price index (to be more precise: the GEKS index based on the Lloyd-Moulton 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). To get information about both price index values and corresponding dates, please see functions: `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` 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.

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

Examples

```
gekslm(milk, start="2019-01", end="2019-08",window=10)
gekslm(milk, start="2018-12", end="2019-12", sigma=0.5)
```

`gekslm_fbew` *Extending the multilateral GEKS-LM price index by using the FBEW method.*

Description

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

Usage

```
gekslm_fbew(data, start, end, sigma)
```

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".
`sigma`	The elasticity of substitution (a parameter used in the Lloyd-Moulton index formula)..

Value

This function returns a value of the multilateral GEKS-LM 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_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` 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

```
gekslm_fbew(milk, start="2018-12", end="2019-08", sigma=1.2)
```

`gekslm_fbmw` *Extending the multilateral GEKS-LM price index by using the FBMW method.*

Description

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

Usage

```
gekslm_fbmw(data, start, end, sigma)
```

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".
sigma	The elasticity of substitution (a parameter used in the Lloyd-Moulton index formula).

Value

This function returns a value of the multilateral GEKS-LM 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_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 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

```
geksqm_fbmw(milk, start="2019-12", end="2020-04", r=1.6)
```

---

gekslm_splice *Extending the multilateral GEKS-LM price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-LM 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

```
gekslm_splice(
  data,
  start,
  end,
```

```
  sigma = 0.7,
  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".
sigma	The elasticity of substitution (a parameter used in the Lloyd-Moulton index formula).
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-LM 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_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 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

```
gekslm_splice(milk, start="2018-12", end="2020-02", sigma=0.8, splice="half")
```

`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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

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

`geksqm` *Calculating the multilateral GEKS-QM price index*

Description

This function returns a value of the multilateral GEKS-QM price index (to be more precise: the GEKS index based on the quadratic mean of order r price index QMp).

Usage

```
geksqm(data, start, end, r = 2, 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".
`r`	The real and non-zero parameter used in the implicit quadratic mean of order r price index.
`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-QM price index (to be more precise: the GEKS index based on the the quadratic mean of order r price index QMp) 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_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` 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.

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

Examples

```
geksqm(milk, start="2019-01", end="2019-08",window=10)
geksqm(milk, start="2018-12", end="2019-12", r=1.6)
```

`geksqm_fbew` *Extending the multilateral GEKS-QM price index by using the FBEW method.*

Description

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

Usage

```
geksqm_fbew(data, start, end, r)
```

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".
`r`	The real and non-zero parameter used in the implicit quadratic mean of order r price index.

Value

This function returns a value of the multilateral GEKS-QM 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_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` 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

```
geksqm_fbew(milk, start="2018-12", end="2019-08", r=1.2)
```

`geksqm_fbmw` *Extending the multilateral GEKS-QM price index by using the FBMW method.*

Description

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

Usage

```
geksqm_fbmw(data, start, end, r)
```

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".
r	The real and non-zero parameter used in the implicit quadratic mean of order r price index.

Value

This function returns a value of the multilateral GEKS-QM 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_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 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

```
geksqm_fbmw(milk, start="2019-12", end="2020-04", r=1.6)
```

---

geksqm_splice *Extending the multilateral GEKS-QM price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral GEKS-QM 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

```
geksqm_splice(
  data,
  start,
  end,
```

```
  r = 2,
  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".
r	The real and non-zero parameter used in the implicit quadratic mean of order r price index.
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-QM 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_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 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

```
geksqm_splice(milk, start="2018-12", end="2020-02", r=0.8, splice="half")
```

`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_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` 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_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` 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_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` 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_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` 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")
```

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_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 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_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 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_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` 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")
```

`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, psigma, 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.

References

Sulewski, P., Białek, J. (2022). *Probability Distribution Modelling of Scanner Prices and Relative Prices*. Statistika – Statistics and Economy Journal, Vol. 3/2022, 282-298, Czech Statistical Office, Prague.

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)
```

---

generate_CES *Generating an artificial scanner dataset in the CES model*

Description

This function provides artificial scanner datasets where prices are lognormally distributed and quantities are obtained under a CES utility.

Usage

```
generate_CES(
  pmi = c(),
  psigma = c(),
  prec = 2,
  elasticity = 0.7,
  S = 1000,
  alfa = c(),
  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.
prec	A numeric value indicating precision, i.e. the number of decimal places, for generating prices.
elasticity	The elasticity of substitution. The default value is 0.7.
S	Sum of spending. The default value is 1000.
alfa	A numeric vector indicating positive weights that reflect the consumer preferences.By default, this vector is randomized based on a uniform distribution.
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 are lognormally distributed, quantities are calculated under the assumption that consumers have CES (Constant Elasticity of Substitution) preferences and their spending on all products is S. The characteristics for the lognormal price distribution are set by pmi and psigma 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.

References

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

Examples

```
#Generating an artificial dataset (the elasticity of substitution is 1.25)
df<-generate_CES(pmi=c(1.02,1.03),psigma=c(0.04,0.03),
elasticity=1.25,start="2020-01",n=100,days=TRUE)
#Verifying the elasticity of substitution
elasticity(df, start="2020-01",end="2020-02")
```

---

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_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 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_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 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_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 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_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 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_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 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_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 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_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 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_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 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_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 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")
```

`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_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` 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_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` 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)
```

IQMp *Calculating the implicit quadratic mean of order r price index*

Description

This function returns a value (or vector of values) of the implicit quadratic mean of order r price index.

Usage

```
IQMp(data, start, end, r = 2, 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".
r	The real and non-zero parameter.
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 implicit quadratic mean of order r price index - see CPI Manual (2004), Section 17.37, formula 17.32 (page 321).

References

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

Examples

```
IQMp(sugar, start="2019-01", end="2020-01")
IQMp(sugar, start="2019-01", end="2020-01", r=1.3, 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_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 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_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` 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_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 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_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` 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
my.grid=list(eta=c(0.01,0.02,0.05),subsample=c(0.5,0.8))
data_train<-dplyr::filter(dataCOICOP,dataCOICOP$time<=as.Date("2021-10-01"))
data_test<-dplyr::filter(dataCOICOP,dataCOICOP$time==as.Date("2021-11-01"))
ML<-model_classification(data_train,data_test,class="coicop6",grid=my.grid,
indicators=c("description","codeIN", "grammage"),key_words=c("uht"),rounds=60)
#Saving the model
## Not run: save_model(ML, dir="My_model")
#Loading the model
## Not run: ML_fromPC<-load_model("My_model")
#classes 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_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 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_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` 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 time dependent matched_index() function*

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,
  base = "start",
  type = "prodID",
  fixedbase = TRUE,
  figure = TRUE,
  date_breaks = "1 month"
)
```

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 beginning of a time interval (as character) limited to the year and month, e.g. "2019-03".
end	The end of a time interval (as character) limited to the year and month, e.g. "2019-04".
base	The base period (as character) for product comparisons. Its possible values are: "start" and "end".
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 period defined by base 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.
date_breaks	A string giving the distance between breaks on the X axis like "1 month" (default value) or "4 months".

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")
```

`mbennet` *Calculating the multilateral Bennet price and quantity indicators*

Description

This function returns the multilateral Bennet price and quantity indicators and optionally also the price and quantity contributions of individual products.

Usage

```
mbennet(
  data,
  start,
  end,
  wstart = start,
  matched = FALSE,
  window = 13,
  interval = FALSE,
  contributions = FALSE,
  prec = 2
)
```

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".
`wstart`	The first period of the time window (as character) limited to the year and month, e.g. "2019-12".
`matched`	A logical parameter indicating whether the matched sample approach is to be used (if yes, the parameter has the value TRUE).
`window`	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
`interval`	A logical parameter indicating whether calculations are to be made for the whole time interval (TRUE) or no (FALSE).

contributions	A logical parameter indicating whether contributions of individual products are to be displayed. If it is TRUE, then contributions are calculated for the the base period start and the current period end.
prec	A numeric vector indicating precision, i.e. the number of decimal places for presenting results.

Value

This function returns the multilateral Bennet price and quantity indicators and optionally also the price and quantity contributions of individual products.

References

Bennet, T. L., (1920). *The Theory of Measurement of Changes in Cost of Living*. Journal of the Royal Statistical Society, 83, 455-462.

Fox, K.J., (2006). *A Method for Transitive and Additive Multilateral Comparisons: A Transitive Bennet Indicator*. Journal of Economics, 87(1), 73-87.

Białek, J. (2024). *The use of the Bennet indicators and their transitive versions for scanner data analysis*. Statistics in Transition new series, 25(3), 155-173.

Examples

```
mbennet(milk, "2018-12", "2019-12", matched=TRUE, contributions=TRUE)
mbennet(coffee, start="2018-12", end="2019-03", interval=TRUE)
```

---

milk *A 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 4386 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 68 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)

`mmontgomery` *Calculating the multilateral Montgomery price and quantity indicators*

Description

This function returns the multilateral Montgomery price and quantity indicators and optionally also the price and quantity contributions of individual products.

Usage

```
mmontgomery(
  data,
  start,
  end,
  wstart = start,
  matched = FALSE,
  window = 13,
  interval = FALSE,
  contributions = FALSE,
  prec = 2
)
```

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".
`wstart`	The first period of the time window (as character) limited to the year and month, e.g. "2019-12".
`matched`	A logical parameter indicating whether the matched sample approach is to be used (if yes, the parameter has the value TRUE).
`window`	The length of the time window (as positive integer: typically multilateral methods are based on the 13-month time window).
`interval`	A logical parameter indicating whether calculations are to be made for the whole time interval (TRUE) or no (FALSE).
`contributions`	A logical parameter indicating whether contributions of individual products are to be displayed. If it is `TRUE`, then contributions are calculated for the the base period `start` and the current period `end`.
`prec`	A numeric vector indicating precision, i.e. the number of decimal places for presenting results.

Value

This function returns the multilateral Montgomery price and quantity indicators and optionally also the price and quantity contributions of individual products.

References

Montgomery, J. K., (1929). *Is There a Theoretically Correct Price Index of a Group of Commodities?* Rome, International Institute of Agriculture

Fox, K.J., (2006). *A Method for Transitive and Additive Multilateral Comparisons: A Transitive Bennet Indicator*. Journal of Economics, 87(1), 73-87.

Białek, J., Pawelec, N. (2024). *The use of transitive Montgomery Indicators for scanner data analysis*. Argumenta Oeconomica, 2(53).

Examples

```
mmontgomery(milk, "2018-12", "2019-12", matched=TRUE, contributions=TRUE)
mmontgomery(coffee, start="2018-12", end="2019-03", interval=TRUE)
```

---

model_classification *Building the machine learning model for product classification*

Description

This function provides a trained machine learning model to classify products into classes or any other groups defined by the user. 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(),
  class = c(),
  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 class column. 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).
class	A character string which indicates the column with classes (groups) of products (e.g. COICOPs).
indicators	A vector of column names to be considered in building a machine learning model. Important: the indicated variables can be numeric but also categorical (factor or character types are acceptable).
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 class-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 classes (or any other 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; `classes` - a dataframe with categorized classes; `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(0.5,0.8))
data_train<-dplyr::filter(dataCOICOP,dataCOICOP$time<=as.Date("2021-10-01"))
data_test<-dplyr::filter(dataCOICOP,dataCOICOP$time==as.Date("2021-11-01"))
ML<-model_classification(data_train,data_test,class="coicop6",grid=my.grid,
indicators=c("description","codeIN","grammage"),key_words=c("uht"),rounds=60)
ML$best_parameters
ML$indicators
ML$figure_training
ML$figure_importance
```

---

montgomery *Calculating the Montgomery price and quantity indicators*

Description

This function returns the Montgomery price and quantity indicators and optionally also the price and quantity contributions of individual products.

Usage

```
montgomery(
  data,
  start,
  end,
  interval = FALSE,
  matched = FALSE,
  contributions = FALSE,
  prec = 2
)
```

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 parameter indicating whether calculations are to be made for the whole time interval (TRUE) or no (FALSE).
matched	A logical parameter indicating whether the matched sample approach is to be used (if yes, the parameter has the value TRUE).
contributions	A logical parameter indicating whether contributions of individual products are to be displayed. If it is TRUE, then contributions are calculated for the the base period start and the current period end.
prec	A numeric vector indicating precision, i.e. the number of decimal places for presenting results.

Value

This function returns the Montgomery price and quantity indicators and optionally also the price and quantity contributions of individual products.

References

Montgomery, J. K., (1929). *Is There a Theoretically Correct Price Index of a Group of Commodities?* Rome, International Institute of Agriculture

Examples

```
montgomery(milk, "2018-12", "2019-12", matched=TRUE, contributions=TRUE)
montgomery(coffee, start="2018-12", end="2019-03", interval=TRUE)
```

---

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_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 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_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 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, date_breaks = "1 month", 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.
date_breaks	A string giving the distance between breaks on the X axis like "1 month" (default value) or "4 months".
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

dataAGGR

dataMATCH

dataCOICOP

data_DOWN_UP_SIZED

milk

sugar

coffee

dataU

generate

tindex

Functions for data processing

data_check

data_preparing

data_aggregating

data_reducing

data_unit

data_norm

data_selecting

data_classifying

model_classification

save_model

load_model

data_matching

data_filtering

shrinkflation

Functions providing dataset characteristics

available

matched

matched_index

matched_fig

prices

quantities

sales

sales_groups

sales_groups2

pqcor

pqcor_fig

dissimilarity_fig

elasticity

elasticity_fig

Functions for bilateral unweighted price index calculation

bmw

carli

cswd

dutot

jevons

harmonic

dikhanov

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

chdikhanov

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

wgeks

geksl

wgeksl

geksgl

wgeksgl

geksaqu

wgeksaqu

geksaqi

wgeksaqi

geksgaqi

wgeksgaqi

geksj

geksw

gk

QU

tpd

SPQ

Functions for extending multilateral price indices by using splicing methods

ccdi_splice

geks_splice

wgeks_splice

geksj_splice

geksw_splice

geksl_splice

wgeksl_splice

geksgl_splice

wgeksgl_splice

geksaqu_splice

wgeksaqu_splice

geksaqi_splice

wgeksaqi_splice

geksgaqi_splice

wgeksgaqi_splice

gk_splice

tpd_splice

Functions for extending multilateral price indices by using the FBEW method

ccdi_fbew

geks_fbew

wgeks_fbew

geksj_fbew

geksw_fbew

geksl_fbew

wgeksl_fbew

geksgl_fbew

wgeksgl_fbew

geksaqu_fbew

wgeksaqu_fbew

geksaqi_fbew

wgeksaqi_fbew

geksgaqi_fbew

wgeksgaqi_fbew

gk_fbew

tpd_fbew

Functions for extending multilateral price indices by using the FBMW method

ccdi_fbmw

geks_fbmw

wgeks_fbmw

geksj_fbmw

geksw_fbmw

geksl_fbmw

wgeksl_fbmw

geksgl_fbmw

wgeksgl_fbmw

geksaqu_fbmw

wgeksaqu_fbmw

geksaqi_fbmw

wgeksaqi_fbmw

geksgaqi_fbmw

wgeksgaqi_fbmw

gk_fbmw

tpd_fbmw

General functions for price index calculations

price_indices

final_index

Functions for comparisons of price indices

compare_indices_df

compare_indices_list

compare_indices_jk

compare_distances

compare_to_target

---

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(), ID = 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 prices of sold products (see also data_matching). If the set is empty, the function returns prices of all products being available in period.
ID	A logical parameter indicating whether a data frame with prodIDs and prices (unit values) should be returned.

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. Please note, that the function returns the price values for sorted prodIDs and in the absence of a given prodID in the data set, the function returns nothing (it does not return zero). If the ID parameter is set to TRUE then the function returns a data frame with columns: by (IDs of products) and uv (unit values of products).

Examples

```
prices(milk, period="2019-06")
prices(milk, period="2019-12", set=c(400032, 82919), ID=TRUE)
```

`price_indices` *A 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,
  formula = c(),
  window = c(),
  splice = c(),
  base = c(),
  sigma = c(),
  r = c(),
  interval = FALSE,
  names = c()
)
```

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`	A vector of character strings indicating price index formulas that are to be calculated. To see available options please use the link: `PriceIndices`.
`window`	A vector of integers. Each element of the vector defines the length of the time window of the corresponding multilateral index.
`splice`	A vector of character strings. Each element of the vector indicates the splicing method is to be used for the corresponding multilateral index. Available values of vector elements are: "movement", "window","half","mean" and their additional variants: "window_published", "half_published" and "mean_published".

base	The vector of prior periods used in the Young- or Lowe-type price indices or hybrid/geohybrid index. Each element of the vector (as character) must be limited to the year and month, e.g. "2020-01".
sigma	The vector of elasticity of substitution parameters used in the Lloyd-Moulton, AG Mean or GEKS-LM indices (as numeric).
r	The vector of non-zero parameters used in the quadratic mean of order r quantity / price index or in the GEKS-QM index (as numeric).
interval	A logical value indicating whether the function is to provide price indices 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).
names	A vector of strings indicating names of indices which are to be used in the resulting data frame.

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 function).

Examples

```
price_indices(milk,
         start="2018-12",end="2019-12",
         formula=c("geks","ccdi","hybrid","fisher",
         "QMp","young","geksl_fbew"),
         window=c(13,13),
         base=c("2019-03","2019-03"),
         r=c(3),interval=TRUE)
price_indices(milk,
         start="2018-12",end="2019-12",
         formula=c("geks","ccdi","hybrid","fisher",
         "QMp","young","geksl_fbew"),
         window=c(13,13),
         base=c("2019-03","2019-03"),
         r=c(3),interval=FALSE)
```

---

products *Detecting and summarising available, matched, new and disappearing products.*

Description

This function detects and summarises available, matched, new as well as disappearing products on the basis of their prodIDs.

Usage

```
products(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') 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".

Value

This function detects and summarises available, matched, new and disappearing products on the basis of their prodIDs. It compares products from the base period (start) with products from the current period (end). It returns a list containing the following objects: details with prodIDs of available, matched, new and disappearing products, statistics with basic statistics for them and figure with a pie chart describing a contribution of matched, new and disappearing products in a set of available products.

Examples

```
list<-products(milk, "2018-12","2019-12")
list$details
list$statistics
list$figure
```

---

products_fig *Function for graphical comparison of available, matched, new as well as disappearing products.*

Description

This function returns a figure with plots of volume (or contributions) of available, matched, new as well as disappearing products.

Usage

```
products_fig(
  data,
  start,
  end,
  show = c("available", "matched", "new", "disappearing"),
  fixed_base = TRUE,
  contributions = TRUE,
  date_breaks = "1 month"
)
```

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') 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".
show	A character vector indicating which groups of products are to be taken into consideration. Available options are available, matched, new and disappearing.
fixed_base	A logical parameter indicating whether each month is to be compared to the base period (TRUE) or to the previous month (then it is set to FALSE).
contributions	A logical parameter indicating whether contributions or volumes counted for available, matched, new and disappearing products are to be displayed.
date_breaks	A string giving the distance between breaks on the X axis like "1 month" (default value) or "4 months".

Value

This function returns a figure with plots of volume (or contributions) of available, matched, new as well as disappearing products. The User may control which groups of products are to be taken into consideration (see the show parameter). Available options are available, matched, new and disappearing.

Examples

```
products_fig(milk, "2018-12","2019-04",
fixed_base=TRUE, contributions=FALSE,
show=c("new","disappearing","matched","available"))
```

QMp *Calculating the quadratic mean of order r price index*

Description

This function returns a value (or vector of values) of the quadratic mean of order r price index.

Usage

```
QMp(data, start, end, r = 2, 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".
r	The real and non-zero parameter.
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 quadratic mean of order r price index - see CPI Manual (2004), Section 17.40, formula 17.35 (page 321).

References

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

Examples

```
QMp(sugar, start="2019-01", end="2020-01")
QMp(sugar, start="2019-01", end="2020-01", r=1.3, interval=TRUE)
```

QMq *Calculating the quadratic mean of order r quantity index*

Description

This function returns a value (or vector of values) of the quadratic mean of order r quantity index.

Usage

```
QMq(data, start, end, r = 2, 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".
r	The real and non-zero parameter.

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 quadratic mean of order r quantity index - see CPI Manual (2004), Section 17.35, formula 17.30 (page 321).

References

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

Examples

```
QMq(sugar, start="2019-01", end="2020-01")
QMq(sugar, start="2019-01", end="2020-01", r=1.3, 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(), ID = 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'), `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`.
`ID`	A logical parameter indicating whether a data frame with prodIDs and quantities should be returned.

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. Please note that the function returns the quantity values for sorted prodIDs and in the absence of a given prodID in the data set, the function returns nothing (it does not return zero). If the ID parameter is set to TRUE then the function returns a data frame with columns: `by` (IDs of products) and `q` (quantities of products).

Examples

```
quantities(milk, period="2019-06")
quantities(milk, period="2019-12", set=c(400032, 82919), ID=TRUE)
```

`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 (see also `expenditures` function which returns the expenditure values for sorted prodIDs).

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_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` 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
my.grid=list(eta=c(0.01,0.02,0.05),subsample=c(0.5,0.8))
data_train<-dplyr::filter(dataCOICOP,dataCOICOP$time<=as.Date("2021-10-01"))
data_test<-dplyr::filter(dataCOICOP,dataCOICOP$time==as.Date("2021-11-01"))
ML<-model_classification(data_train,data_test,class="coicop6",grid=my.grid,
indicators=c("description","codeIN", "grammage"),key_words=c("uht"),rounds=60)
#Saving the model
## Not run: save_model(ML, dir="My_model")
```

---

shrinkflation *Detecting and summarising downsized and upsized products.*

Description

This function detects and summarises downsized and upsized products.

Usage

```
shrinkflation(
  data,
  start,
  end,
  type = "shrinkflation",
  min_p_change = 0,
  max_p_change = Inf,
  min_s_change = 0,
  max_s_change = Inf,
  prec = 3,
  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') and `prodID` (as numeric, factor or character), `prices` (with standardised prices!) and `quantities` (as numeric), `grammage` (as numeric), `unit` (as character) and `description` (as character). Important: prices must be standardized beforehand, that is, they must refer to the sales unit (the `data_norm` function can be used for this).
`start`	The base period (as character) limited to the year and month, e.g. "2024-01".
`end`	The research period (as character) limited to the year and month, e.g. "2024-02".

type	A parameter specifying what phenomenon is to be included in the resulting elements of the returned list (i.e. in returned products_detected, df_detected and df_reduced). The available values are: shrinkflation, shrinkdeflation, sharkflation, unshrinkdeflation, unshrinkflation and sharkdeflation (default value is: shrinkflation).
min_p_change	Lower limit for unit price change, i.e.: a product is considered if the percentage change in its unit price is greater than the value of this parameter. The default value is zero, possibly positive values can be considered (in percentage).
max_p_change	Upper limit for unit price change, i.e.: a product is considered if the percentage change in its unit price is less than the value of this parameter. The default value is Inf, possibly positive values can be considered (in percentage).
min_s_change	Lower limit for size change, i.e.: a product is considered if the percentage change in its size is greater than the value of this parameter. The default value is zero, possibly positive values can be considered (in percentage).
max_s_change	Upper limit for size change, i.e.: a product is considered if the percentage change in its size is less than the value of this parameter. The default value is Inf, possibly positive values can be considered (in percentage).
prec	Number of decimal places for the presented summary results.
interval	A parameter that specifies whether the search for downsized products should consider the entire time interval, or only the compared months specified by the start and end parameters.

Value

This function detects and summarises downsized and upsized products. The function detects phenomena such as: shrinkflation, shrinkdeflation, sharkflation, unshrinkdeflation, unshrinkflation, sharkdeflation (see the type parameter). It returns a list containing the following objects: df_changes - data frame with detailed information on downsized and upsized products with the whole history of size changes, df_type - data frame with recognized type of products, df_overview - a table with basic summary of all detected products grouped by the type parameter, products_detected with prodIDs of products indicated by the 'type' parameter, df_detected being a subset of the data frame with only detected products, df_reduced which is the difference of the input data frame and the data frame containing the detected products, and df_summary which provides basic statistics for all detected downsized and upsized products (including their share in the total number of products and mean price and size changes).

References

Białek, J., Bobel, A., Oprych-Franków D. (2004). *Immeasurability of shrinkflation in the CPI? Automatic downsizing detection using scanner data*. 18th Meeting of the Ottawa Group, Ottawa.

Examples

```
#Data matching over time
df<-data_matching(data=data_DOWN_UP_SIZED, start="2024-01", end="2024-02",
codeIN=TRUE,codeOUT=TRUE,description=TRUE,
onlydescription=FALSE,precision=0.9,interval=FALSE)
# Extraction of information about grammage (if needed)
```

```
df<-data_unit(df,units=c("g|ml|kg|l"),multiplication="x")
# Price standardization
df<-data_norm(df, rules=list(c("ml","l",1000),c("g","kg",1000)))
# Downsized and upsized products detection
result<-shrinkflation(data=df, start="2024-01","2024-02",
prec=3, interval=FALSE, type="shrinkflation")
result$df_changes
result$df_type
result$df_overview
result$products_detected
result$df_detected
result$df_reduced
result$df_summary
```

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_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` 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_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` 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 *A 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_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` 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_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 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_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 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_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 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_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 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")
```

---

unit_value_index *Calculating the unit value index*

Description

This function returns a value (or vector of values) of the unit value index

Usage

```
unit_value_index(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 unit value index. The value index is calculated as the unit value at time start divided by the unit value at time start.

References

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

Examples

```
unit_value_index(sugar, start="2019-01", end="2020-01")
unit_value_index(sugar, start="2019-01", end="2020-01", interval=TRUE)
```

---

utpd *Calculating the unweighted multilateral TPD price index*

Description

This function returns a value of the unweighted multilateral TPD (Time Product Dummy) price index.

Usage

```
utpd(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 unweighted 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 the estimation procedure runs the Ordinary Least Squares (OLS) method instead of the Weighted Least Squares (WLS) method like in the case of the TPD index. To get information about both price index values and corresponding dates, please see functions: `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` 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

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

`utpd_fbew` *Extending the unweighted multilateral TPD price index by using the FBEW method.*

Description

This function returns a value of the unweighted multilateral TPD price index (Time Product Dummy index) extended by using the FBEW (Fixed Base Monthly Expanding Window) method.

Usage

```
utpd_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 unweighted 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_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` 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

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

`utpd_fbmw` *Extending the unweighted multilateral TPD price index by using the FBMW method.*

Description

This function returns a value of the unweighted multilateral TPD price index (Time Product Dummy index) extended by using the FBMW (Fixed Base Moving Window) method.

Usage

```
utpd_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 unweighted 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_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` 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

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

---

utpd_splice *Extending the multilateral unweighted TPD price index by using window splicing methods.*

Description

This function returns a value (or values) of the unweighted 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

```
utpd_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 unweighted 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_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` 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

```
utpd_splice(milk, start="2018-12", end="2020-02",splice="half")
```

`value_index` *Calculating the value index*

Description

This function returns a value (or vector of values) of the value index

Usage

```
value_index(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 value index. The value index is calculated as sum of expenditures from period `end` divided by sum of expenditures from period `start`.

References

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

Examples

```
value_index(sugar, start="2019-01", end="2020-01")
value_index(sugar, start="2019-01", end="2020-01", 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_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 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_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 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)
```

wgeks *Calculating the multilateral weighted WGEKS price index*

Description

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

Usage

```
wgeks(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 price index (to be more precise: the weighted 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_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 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

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

wgeksaqi *Calculating the multilateral weighted WGEKS-AQI price index*

Description

This function returns a value of the multilateral weighted WGEKS-AQI price index (to be more precise: the weighted GEKS index based on the asynchronous quality adjusted price index formula).

Usage

```
wgeksaqi(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-AQI price index (to be more precise: the weighted GEKS index based on the asynchronous quality adjusted price index 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`wgeksaqi_fbew` *Extending the multilateral weighted GEKS-AQI price index by using the FBEW method.*

Description

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

Usage

```
wgeksaqi_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-AQI price index (the weighted GEKS index based on the asynchronous quality adjusted price index 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`wgeksaqi_fbmw` *Extending the multilateral weighted GEKS-AQI price index by using the FBMW method.*

Description

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

Usage

```
wgeksaqi_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-AQI price index (the GEKS index based on the asynchronous quality adjusted price index 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`wgeksaqi_splice` *Extending the multilateral weighted GEKS-AQI price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral weighted GEKS-AQI 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

```
wgeksaqi_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-AQI price index (the weighted GEKS index based on the asynchronous quality adjusted price index 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

```
wgeksaqi_splice(milk, start="2018-12", end="2020-02",splice="half")
```

`wgeksaqu` *Calculating the multilateral weighted WGEKS-AQU price index*

Description

This function returns a value of the multilateral weighted WGEKS-AQU price index (to be more precise: the weighted GEKS index based on the asynchronous quality adjusted unit value formula).

Usage

```
wgeksaqu(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-AQU price index (to be more precise: the weighted GEKS index based on the asynchronous quality adjusted unit value 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

`wgeksaqu_fbew` *Extending the multilateral weighted GEKS-AQU price index by using the FBEW method.*

Description

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

Usage

```
wgeksaqu_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-AQU price index (the weighted GEKS index based on the asynchronous quality adjusted unit value 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

---

wgeksaqu_fbmw *Extending the multilateral weighted GEKS-AQU price index by using the FBMW method.*

Description

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

Usage

```
wgeksaqu_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-AQU price index (the GEKS index based on the asynchronous quality adjusted unit value 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

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

---

wgeksaqu_splice *Extending the multilateral weighted GEKS-AQU price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral weighted GEKS-AQU 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

```
wgeksaqu_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-AQU price index (the weighted GEKS index based on the asynchronous quality adjusted unit value 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2023). *Quality adjusted GEKS-type indices for price comparisons based on scanner data.* Statistics in Transition – new series, 24(3), 151-169.

Examples

```
wgeksaqu_splice(milk, start="2018-12", end="2020-02",splice="half")
```

`wgeksgaqi` *Calculating the multilateral weighted WGEKS-GAQI price index*

Description

This function returns a value of the multilateral weighted WGEKS-GAQI price index (to be more precise: the weighted GEKS index based on the geometric asynchronous quality adjusted price index formula).

Usage

```
wgeksgaqi(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-GAQI price index (to be more precise: the weighted GEKS index based on the geometric asynchronous quality adjusted price index 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_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 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

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

---

wgeksgaqi_fbew *Extending the multilateral weighted GEKS-GAQI price index by using the FBEW method.*

Description

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

Usage

```
wgeksgaqi_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-GAQI price index (the weighted GEKS index based on the geometric asynchronous quality adjusted price index 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_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` 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

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

---

wgeksgaqi_fbmw — *Extending the multilateral weighted GEKS-GAQI price index by using the FBMW method.*

Description

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

Usage

```
wgeksgaqi_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-GAQI price index (the GEKS index based on the geometric asynchronous quality adjusted price index 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_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 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

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

---

wgeksgaqi_splice *Extending the multilateral weighted GEKS-GAQI price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral weighted GEKS-GAQI 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

```
wgeksgaqi_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-GAQI price index (the weighted GEKS index based on the geometric asynchronous quality adjusted price index 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_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` 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

```
wgeksgaqi_splice(milk, start="2018-12", end="2020-02",splice="half")
```

`wgeksgl` *Calculating the multilateral weighted WGEKS-GL price index*

Description

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

Usage

```
wgeksgl(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-GL price index (to be more precise: the weighted GEKS index based on the geometric 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

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

`wgeksgl_fbew` *Extending the multilateral weighted GEKS-GL price index by using the FBEW method.*

Description

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

Usage

```
wgeksgl_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-GL price index (the weighted GEKS index based on the geometric 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

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

`wgeksgl_fbmw` *Extending the multilateral weighted GEKS-GL price index by using the FBMW method.*

Description

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

Usage

```
wgeksgl_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-GL price index (the GEKS index based on the geometric 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

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

---

wgeksgl_splice *Extending the multilateral weighted GEKS-GL price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral weighted GEKS-GL 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

```
wgeksgl_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-GL price index (the weighted GEKS index based on the geometric 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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

```
wgeksgl_splice(milk, start="2018-12", end="2020-02",splice="half")
```

`wgeksl` *Calculating the multilateral weighted 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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

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_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 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

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_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` 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.

Białek, J. (2022). *The general class of multilateral indices and its two special cases.* Paper presented at the 17th Meeting of the Ottawa Group on Price Indices, Rome, Italy.

Białek, J. (2022). *Improving quality of the scanner CPI: proposition of new multilateral methods*, Quality & Quantity, https://doi.org/10.1007/s11135-022-01506-6.

Examples

```
wgeksl_splice(milk, start="2018-12", end="2020-02",splice="half")
```

`wgeks_fbew` *Extending the multilateral weighted GEKS price index by using the FBEW method.*

Description

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

Usage

```
wgeks_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 price index (the weighted GEKS index based on the Fisher 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_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 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

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

wgeks_fbmw *Extending the multilateral weighted GEKS price index by using the FBMW method.*

Description

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

Usage

```
wgeks_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 price index (the weighted GEKS index based on the Fisher 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_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 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

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

wgeks_splice *Extending the multilateral weighted GEKS price index by using window splicing methods.*

Description

This function returns a value (or values) of the multilateral weighted 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

```
wgeks_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 price index (the weighted GEKS index based on the Fisher 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_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` 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

```
wgeks_splice(milk, start="2018-12", end="2020-02",splice="half")
```

---

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_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` 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