Package ‘multilateral’

April 20, 2022

Type Package
Title Generalised Function to Calculate a Variety of Multilateral Price Index Methods
Version 1.0.0
Description A flexible, efficient implementation of multilateral price index calculations. Includes common methods focused on time product dummy regression and GEKS variations. Allows for extension of the methods through automatic window splicing. See Krsinich (2016) <doi:10.1515/jos-2016-0021>.
License GPL (>= 3)
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
URL https://github.com/MjStansfi/multilateral
Suggests testthat, knitr, rmarkdown, devtools, dplyr, ggplot2
VignetteBuilder knitr
Imports assertive, data.table, fastmatch, parallel, yaml, MatrixModels
NeedsCompilation no
Author Matthew Stansfield [aut, cre]
Maintainer Matthew Stansfield <m.stansfield85@gmail.com>
Repository CRAN
Date/Publication 2022-04-20 08:00:02 UTC

R topics documented:

gm_mean .......................................................... 2
multilateral ....................................................... 2
splice_update ..................................................... 4
synthetic_gfk .................................................... 4
turvey ............................................................. 5

Index 6
### gm_mean

**Description**

Calculate the geometric mean of a vector of numbers

**Usage**

```r
gm_mean(x, na.rm = TRUE)
```

**Arguments**

- `x`: an R numerical object
- `na.rm`: a logical value indicating whether NA values should be stripped before the computation proceeds.

**Value**

If all values in `x` are numeric class, a single numeric class value is returned.

**Examples**

```r
x <- c(0:10, 50)
gm_mean(x)
```

### multilateral

**Description**

A flexible implementation of multilateral price index calculation for scanner data. This function can be applied on any dataset where key attributes exist (depending on method). Those are in general terms a period, ID, price, and quantity. It will allow for extension of the method by the use of calculation over a window of time and splicing them together.

**Usage**

```r
multilateral(
    period,
    price,
    index_method,
    check_inputs_ind = TRUE,
    verbose = FALSE,
    ...
)```
## S3 method for class 'multilateral'
print(x, ...)

### Arguments
- **period**: vector of the periods corresponding to price observations.  
  NOTE: period must be of class Date or numeric.
- **price**: vector of prices
- **index_method**: The index method of choice
- **check_inputs_ind**: logical, whether to check inputs or not
- **verbose**: print additional information to console
- **...**: All other possible arguments, see details
- **x**: multilateral class object

### Details
The function takes vectors for each of the inputs. It is important to note that the `period` argument must be of numeric or Date class. This is because the order of the dates matters.

The function also has the capability to run in parallel, using the `num_cores` argument. Note that for smaller datasets using non-parallel code is often faster than using parallelisation due to the overhead associated with dividing the job across multiple cores.

... represents all other possible arguments the user can provide, they include: id, quantity, weight, features, splice_method, window_length, matched, chain_method, num_cores

The `index_method` can be one of 'TPD', 'TDH', 'GEKS-J', 'GEKS-F', 'GEKS-T', or 'GEKS-IT' you can view the configuration file found under inst/config/index_method_config for more information

The `splice_method` can be one of 'half', 'window', 'movement', 'geomean', or 'geomean_short' you can view the configuration file found under inst/config/splice_method_config for more information

### Value
A list object of length 3 containing:
- **index**: a data.frame of the final spliced price index based on the method specified
- **index_windows**: a data.frame containing each individual windows index before splicing
- **splice_detail**: a data.frame containing the breakdown of splice information

### Examples
```r
tpd_index <- multilateral(period = turvey$month,  
                          id = turvey$commodity,  
                          price = turvey$price,  
                          quantity = turvey$quantity,  
                          splice_method = "geomean",
```
splice_update

Splice update

Description

Calculate the splice factor

Usage

splice_update(old_window, new_window, splice_method)

Arguments

old_window vector of numeric values
new_window vector of numeric values
splice_method Method of splicing

Value

A numeric update factor of length 1, based on the splice_method provided.

synthetic_gfk

Synthetic scanner data for one consumer electronic product

Description

GfK have made this available as a public good for the international statistical community, to aid research into new price index methods.

Usage

synthetic_gfk

Format

data frame with 5509 rows and 15 variables

month_num Month number, 0-25
char1-11 Product characteristics
prodid_num Product identifier, created from unique characteristics
quantity The quantity of items sold in that month as an integer
value Sales total (NZD)
**Details**

The data is based on one product category from the scanner data used in production of the New Zealand Consumers Price Index.

The synthetic data has been heavily modified to remove identification potential, while still retaining some of the characteristics of scanner data which make traditional index methods inadequate - such as high product turnover and volatile price and quantities - which motivate the multilateral index methods that are currently being researched within the international statistical community.

**Source**

GfK New Zealand

---

**Artificial prices of seasonal products Data created by R. Turvey**

---

**Description**

The dataset is presented in the ILO CPI manual. In 1979 Turvey sent his artificial data set to statistical agencies around the world, asking them to use their normal techniques to construct monthly and annual average price indices. About 20 countries replied, and Turvey summarized the responses as follows: "It will be seen that the monthly indices display very large differences, e.g., a range of 129.12–169.50 in June, while the range of simple annual means is much smaller. It will also be seen that the indices vary as to the peak month or year."

**Usage**

turvey

**Format**

data frame with 176 rows and 4 variables

- **month** The time as a Date type
- **commodity** The seasonal product, as a factor (Apples, Grapes, Oranges, Peaches, Strawberries)
- **price** The price as numeric
- **quantity** The quantity of items sold in that month as an integer

**Source**

Index

* datasets
  synthetic_gfk, 4
  turvey, 5

gm_mean, 2
multilateral, 2
print.multilateral(multilateral), 2
splice_update, 4
synthetic_gfk, 4
turvey, 5