

Package ‘IndexNumberTools’

July 21, 2025

Type Package

Title Working with Index Numbers

Version 1.1

Description A set of utilities for manipulating index numbers series including chain-linking, re-referencing, and computing growth rates.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports methods, stats

Depends R (>= 4.1.0)

URL <https://mserrano-ine.github.io/IndexNumberTools/>,
<https://github.com/mserrano-ine/IndexNumberTools>

Suggests knitr, rmarkdown, dplyr

VignetteBuilder knitr

NeedsCompilation no

Author Miguel Serrano [aut, cre]

Maintainer Miguel Serrano <miguel.serrano.martin@ine.es>

Repository CRAN

Date/Publication 2025-03-03 12:10:18 UTC

Contents

aggr_rep_lag	2
apply_to_columns	3
change_ref_year	3
compute_gr	4
gdp_current	4
gdp_volume	5

get_chain_linked	5
get_pyp	6
get_q_index	7
get_v_index	7
Index	9

aggr_rep_lag	<i>Aggregate, repeat and lag</i>
--------------	----------------------------------

Description

Helper function to repeat the aggregate annual value of a series on each period, and possibly lag it.

Usage

```
aggr_rep_lag(x, fun = mean, k = 0)
```

Arguments

- x (ts) Any time series
- fun (function) Aggregation function, mean by default
- k (int) Units to lag.

Details

Applies `aggregate.ts` to the series to get the annual values and then repeats those values for every subyear period.

The `k` parameter is passed to `stats::lag`.

Value

description

Examples

```
aggr_rep_lag(gdp_volume) |> plot()
```

apply_to_columns	<i>Apply method to multivariate</i>
------------------	-------------------------------------

Description

This function applies a function for univariate series ("ts") to a multivariate series ("mts").

Usage

```
apply_to_columns(x, f, ...)
```

Arguments

x	(mts) A multivariate time series.
f	(function) A function that takes an univariate series as input.
...	Arguments for f.

change_ref_year	<i>Change reference year</i>
-----------------	------------------------------

Description

Changes the reference year of a chain-linked series (with annual overlap).

Usage

```
change_ref_year(x, new_ref_year)
```

Arguments

x	(ts) A chain-linked series with annual overlap.
new_ref_year	(num) New reference year.

Value

The re-referenced index series.

Examples

```
change_ref_year(gdp_volume, 2015)
plot(gdp_volume)
lines(change_ref_year(gdp_volume, 2015))
```

compute_gr	<i>Compute the growth-rate series</i>
------------	---------------------------------------

Description

Function that computes the growth-rate series of a given time series.

Usage

```
compute_gr(x, s)
```

Arguments

x	(ts) A time series.
s	(int) Lag at which the growth-rate is computed.

Value

Series of growth-rates.

Examples

```
compute_gr(gdp_current, 4)
```

gdp_current	<i>Spanish GDP (Current prices)</i>
-------------	-------------------------------------

Description

Spanish GDP from 1995 Q1 to 2024 Q4.

Format

A univariate time series object.

Source

<https://ine.es/jaxiT3/Tabla.htm?t=67823&L=1> Spanish National Statistics Institute.

gdp_volume	Spanish GDP (Chain-linked volume)
------------	-----------------------------------

Description

Quantity chain-linked indices of the Spanish GDP from 1995 Q1 to 2024 Q4 with reference year 2020.

Format

A univariate time series object

Source

<https://ine.es/jaxiT3/Tabla.htm?t=67824&L=1> Spanish National Statistics Institute.

get_chain_linked	Get chain-linked indices
------------------	--------------------------

Description

Computes chain-linked index series from a pyp series.

Usage

```
get_chain_linked(x, ref_year, x_a = NULL)
```

Arguments

- x (ts) A pyp series.
- ref_year (num) Reference year for the chain-linked series.
- x_a (ts) Annual pyp series. If not given, it's computed by taking the average of each year.

Details

The chain-linked series `x_chain` is computed with the annual overlap method. Suppose the `x` series runs from $(y_0, p_0 = 0)$ to (y_1, p_1) , where p_i is a subyear period. Then the chain-linked series at (y_2, p_2) is given by the cumulative product of the annual series from y_0 to y_2-1 times `x` at (y_2, p_2) .

Value

The chain-linked series.

Examples

```
gdp_pyp <- get_pyp(gdp_volume)
get_chain_linked(gdp_pyp, 2015)
```

get_pyp*Get pyp indices*

Description

Computes the pyp index series from a chain-linked series.

Usage

```
get_pyp(x, x_a = NULL)
```

Arguments

x	(ts) Chain-linked series with annual overlap.
x_a	(ts) Annual chain-linked series. If not given, it's computed by taking the average of each year.

Details

The time series should start at (y,1) where y is any year.

Value

The pyp series.

Examples

```
get_pyp(gdp_volume)
```

get_q_index	<i>Get quantity index</i>
-------------	---------------------------

Description

Returns the series of quantity indices in previous year prices from a current prices and

Usage

```
get_q_index(current, constant)
```

Arguments

current	(ts) Values at current prices.
constant	(ts) Values at previous year prices.

Value

Series of quantity indices for previous year prices.

Examples

```
gdp_pyp <- get_pyp(gdp_volume)
gdp_constant <- gdp_current / gdp_pyp * 100
get_q_index(gdp_current, gdp_constant)
```

get_v_index	<i>Get value index</i>
-------------	------------------------

Description

Returns the (not chain-linked) series of value indices from a series of current prices.

Usage

```
get_v_index(current)
```

Arguments

current	(ts) Series of current prices series.
---------	---------------------------------------

Details

The value of the resulting series x at (y,s) , where y is the year and s is the subyear period, is $\text{current}(y,s)/\text{current}(y)$

Value

(ts) Series of value indices.

Examples

```
get_v_index(gdp_current)
```

Index

aggr_rep_lag, [2](#)
apply_to_columns, [3](#)

change_ref_year, [3](#)
compute_gr, [4](#)

gdp_current, [4](#)
gdp_volume, [5](#)
get_chain_linked, [5](#)
get_pyp, [6](#)
get_q_index, [7](#)
get_v_index, [7](#)