Package 'REPS'

July 30, 2025

Type Package

Title Hedonic and Multilateral Index Methods for Real Estate Price Statistics

Version 1.0.0

Maintainer Vivek Gajadhar <v.gajadhar@cbs.nl>

Description Compute price indices using various Hedonic and multilateral methods, including Laspeyres, Paasche, Fisher, and HMTS (Hedonic Multilateral Time series re-estimation with splicing). The central function calculate_price_index() offers a unified interface for running these methods on structured datasets. This package is designed to support index construction workflows across a wide range of domains — including but not limited to real estate — where quality-adjusted price comparisons over time are essential.
The development of this package was funded by Eurostat and Statistics Netherlands (CBS), and carried out by Statistics Netherlands.
The HMTS method implemented here is described in Ishaak, Ouwehand and Remøy (2024) <doi:10.1177/0282423X241246617>. For broader methodological context, see Eurostat (2013, ISBN:978-92-79-25984-5, <doi:10.2785/34007>).

License EUPL-1.2

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 4.4.0)

Imports dplyr, stats, KFAS, stringr, lmtest

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/vivekag7/REPS

BugReports https://github.com/vivekag7/REPS/issues

NeedsCompilation no

Author Farley Ishaak [aut],
Pim Ouwehand [aut],
David Pietersz [aut],
Liu Nuo Su [aut],
Cynthia Cao [aut],
Mohammed Kardal [aut],
Odens van der Zwan [aut],
Vivek Gajadhar [aut, cre]

Repository CRAN

Date/Publication 2025-07-30 08:10:02 UTC

Contents

`calculate_geometric_average` *Calculate the geometric average of a series of values*

Description

The equation for the calculation is:: exp(mean(log(series_values)))

Usage

```
calculate_geometric_average(values)
```

Arguments

`values`	series with numeric values

Value

geometric average

Author(s)

Farley Ishaak

`calculate_price_index` *Calculate index based on specified method (Fisher, Laspeyres, Paasche, HMTS, Time Dummy, Rolling Time Dummy)*

Description

Central hub function to calculate index figures using different methods.

Usage

```
calculate_price_index(
  dataset,
  method,
  period_variable,
  dependent_variable,
  numerical_variables = NULL,
  categorical_variables = NULL,
  reference_period = NULL,
  number_of_observations = TRUE,
  periods_in_year = 4,
  production_since = NULL,
  number_preliminary_periods = 3,
  resting_points = FALSE,
  imputation = FALSE,
  window_length = 5
)
```

Arguments

`dataset`	Data frame with input data
`method`	One of: "fisher", "laspeyres", "paasche", "hmts", "timedummy", "rolling_timedummy", "repricing"
`period_variable`	A string with the name of the column containing time periods. Values must follow a consistent format such as "2020Q1" (quarterly), "2020M01" (monthly), "202001" (YYYYMM), "2020W01" (weekly), or "2020" (yearly). Mixed or irregular formats (e.g., "Q1_2020", "Jan2020") are not supported.
`dependent_variable`	Usually the price
`numerical_variables`	Vector with numeric quality-determining variables
`categorical_variables`	Vector with categorical variables (also dummies)
`reference_period`	Period or group of periods that will be set to 100

`number_of_observations`
: Logical, whether to show number of observations (default = TRUE)

`periods_in_year`
: (HMTS only) Number of periods per year (e.g. 12 for months)

`production_since`
: (HMTS only) Start period for production simulation

`number_preliminary_periods`
: (HMTS only) Number of preliminary periods

`resting_points`
: (HMTS only) Whether to return detailed outputs (default = FALSE)

`imputation`
: (Laspeyres/Paasche only) Include imputation values? Default = FALSE

`window_length`
: (Rolling Time Dummy only) Window size in number of periods

Value

A data.frame (or list for HMTS with resting_points = TRUE; or named list if multiple methods are used)

Author(s)

Vivek Gajadhar

Examples

```
# Example: Time Dummy index
Tbl_TD <- calculate_price_index(
  method = "timedummy",
  dataset = data_constraxion,
  period_variable = "period",
  dependent_variable = "price",
  numerical_variables = "floor_area",
  categorical_variables = "neighbourhood_code",
  reference_period = "2015",
  number_of_observations = FALSE
)
head(Tbl_TD)

# Example: Multiple methods (Fisher, Paasche, Laspeyres)
multi_result <- calculate_price_index(
  method = c("fisher", "paasche", "laspeyres"),
  dataset = data_constraxion,
  period_variable = "period",
  dependent_variable = "price",
  numerical_variables = "floor_area",
  categorical_variables = "neighbourhood_code",
  reference_period = "2015",
  number_of_observations = FALSE
)

head(multi_result$fisher)
head(multi_result$paasche)
head(multi_result$laspeyres)
```

`calculate_regression_diagnostics` *Calculate regression diagnostics by period*

Description

For each period in the data, fits a log-linear model and computes diagnostics:

- Normality test (Shapiro-Wilk)
- Adjusted R-squared
- Breusch-Pagan test for heteroscedasticity
- Durbin-Watson test for autocorrelation

Usage

```
calculate_regression_diagnostics(
  dataset,
  period_variable,
  dependent_variable,
  numerical_variables = NULL,
  categorical_variables = NULL
)
```

Arguments

`dataset`	A data.frame with input data
`period_variable`	Name of the period variable (string)
`dependent_variable`	Name of the dependent variable (string)
`numerical_variables`	Vector of numerical independent variables (default = NULL)
`categorical_variables`	Vector of categorical independent variables (default = NULL)

Value

A data.frame with diagnostics by period

Author(s)

Mohammad Kardal, Vivek Gajadhar

Examples

```
diagnostics <- calculate_regression_diagnostics(
  dataset = data_constraxion,
  period_variable = "period",
  dependent_variable = "price",
  numerical_variables = c("floor_area", "dist_trainstation"),
  categorical_variables = c("dummy_large_city", "neighbourhood_code")
)
head(diagnostics)
```

---

data_constraxion *A real estate example dataframe*

Description

A subset of data from a fictitious real estate data frame containing transaction prices and some categorical and numerical characteristics of each dwelling.

Usage

```
data_constraxion
```

Format

A data frame with 7,800 rows and 6 columns:

period A (string) vector indicating a time period

price A (string) vector indicating the transaction price of the dwelling

floor_area A real-valued vector of (the logarithm of) the floor area of the dwelling

dist_trainstation A real-valued vector of (the logarithm of) the distance of the dwelling to the nearest train station

neighbourhood_code A categorical code/string referring to the neighbourhood the dwelling belongs to

dummy_large_city A vector indicating whether the dwelling belongs to a large city or not

Source

A fictitious dataset for illustration purposes

Examples

```
data(data_constraxion)
head(data_constraxion)
```

plot_price_index *Plot index output from calculate_price_index*

Description

Static price index plot using base R graphics with grid lines and external legend.

Usage

```
plot_price_index(index_output, title = NULL)
```

Arguments

index_output	A data.frame or named list of data.frames (from calculate_price_index())
title	Optional plot title

Details

Supports both single index data.frame and named list of multiple methods. X-axis shows only first period of each year with rotated labels to avoid clutter.

Value

None. Draws plots in the active graphics device.

Author(s)

Vivek Gajadhar

plot_regression_diagnostics *Plot diagnostics output from calculate_regression_diagnostics as a multi-panel grid (base R)*

Description

Creates a static 3x2 grid of base R plots showing regression diagnostics:

- Normality (Shapiro-Wilk)
- Linearity (Adjusted R-squared)
- Heteroscedasticity (Breusch-Pagan)
- Autocorrelation (Durbin-Watson)
- Autocorrelation (p-value DW)

Usage

```
plot_regression_diagnostics(diagnostics, title = "Regression Diagnostics")
```

Arguments

diagnostics	A data.frame as returned by calculate_regression_diagnostics()
title	Optional overall title for the entire plot grid (default: "Regression Diagnostics")

Value

None. Produces plots in the active graphics device.

Author(s)

Vivek Gajadhar

Examples

```
plot_regression_diagnostics(
  calculate_regression_diagnostics(
    dataset = data_constraxion,
    period_variable = "period",
    dependent_variable = "price",
    numerical_variables = c("floor_area", "dist_trainstation"),
    categorical_variables = c("dummy_large_city", "neighbourhood_code")
  )
)
```

Index