Package: REPS (via r-universe)

October 8, 2026

Type Package

Title Hedonic and Multilateral Index Methods for Real Estate Price Statistics

Version 1.1.1

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_hedonic_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

Roxygen list(markdown = TRUE)

Depends R (>= 4.4.0)

Imports dplyr, stats, KFAS, stringr, lmtest, parallel

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
Config/roxygen2/version 8.0.0
Config/pak/sysreqs libicu-dev
Repository https://vivekag7.r-universe.dev
Date/Publication 2026-07-08 12:16:47 UTC
RemoteUrl https://github.com/vivekag7/reps
RemoteRef HEAD
RemoteSha 19842eecf24b2c7851e530eedac82fce1961ca89

Contents

calculate_hedonic_index 2
calculate_price_index 4
calculate_regression_diagnostics 5
hedonic_data 6
plot_price_index 7
plot_regression_diagnostics 7

Index **9**

`calculate_hedonic_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. Can also calculate chained indices using the Annual Overlap Method.

Usage

```
calculate_hedonic_index(
  dataset,
  method,
  period_variable,
  dependent_variable,
  numerical_variables = NULL,
  categorical_variables = NULL,
  reference_period = NULL,
  number_of_observations = TRUE,
  chained = FALSE,
  index_contribution = FALSE,
  parallel = FALSE,
  ...
)
```

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.
`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)
`chained`	Logical. If TRUE, calculates a chained index using the Annual Overlap Method. Default is FALSE.
`index_contribution`	Logical. If TRUE, calculates an index-contribution table for one method. Default is FALSE.
`parallel`	Logical. If TRUE, independent calculations are parallelized where useful. Default is FALSE.
`...`	Additional method-specific arguments passed to the underlying functions:

- `periods_in_year`: (Required for Repricing) Number of periods per year (e.g. 12 for months, 4 for quarters)
- `number_preliminary_periods`: (Optional for HMTS) Number of preliminary periods. Default = 3
- `production_since`: (Optional for HMTS) Start period for production simulation. Default = NULL
- `resting_points`: (Optional for HMTS) Whether to return detailed outputs. Default = FALSE
- `imputation`: (Optional for Laspeyres/Paasche) Include imputation values? Default = FALSE
- `window_length`: (Optional for Rolling Time Dummy) Window size in number of periods. Default = 5
- `unit_variable`: (Optional for index contribution) Unit column to exclude as groups. Default = NULL
- `index_contribution_period`: (Optional for index contribution) Period to analyze. Default = latest period

Value

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

Author(s)

Vivek Gajadhar

Examples

```
data("hedonic_data")

Tbl_indices <- REPS::calculate_hedonic_index(
  method = c("fisher"),
  dataset = hedonic_data,
  period_variable = "period",
  dependent_variable = "price",
  numerical_variables = c("floor_area", "dist_trainstation"),
  categorical_variables = c("neighbourhood_code", "dummy_large_city"),
  reference_period = "2015",
  number_of_observations = FALSE,
  periods_in_year = 4,
  number_preliminary_periods = 1,
  window_length = 4,
  production_since = NULL,
  resting_points = FALSE,
  imputation = FALSE
)
```

---

calculate_price_index *Calculate Price Index (Deprecated)*

Description

This function has been renamed. Please use `calculate_hedonic_index` instead.

Usage

```
calculate_price_index(...)
```

Arguments

`...`	Arguments passed to `calculate_hedonic_index`.

`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,
  parallel = FALSE
)
```

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)
`parallel`	Logical; whether independent period-level diagnostics are parallelized.

Value

A data.frame with diagnostics by period

Author(s)

Mohammad Kardal, Vivek Gajadhar

Examples

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

---

hedonic_data *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

```
hedonic_data
```

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(hedonic_data)
head(hedonic_data)
```

plot_price_index *Plot index output from calculate_hedonic_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_hedonic_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 = hedonic_data,
    period_variable = "period",
    dependent_variable = "price",
    numerical_variables = c("floor_area", "dist_trainstation"),
    categorical_variables = c("dummy_large_city", "neighbourhood_code")
  )
)
```

Index

∗ **datasets**
hedonic_data, 6

calculate_hedonic_index, 2, *4*
calculate_price_index, 4
calculate_regression_diagnostics, 5

hedonic_data, 6

plot_price_index, 7
plot_regression_diagnostics, 7