Package: cbsREPS (via r-universe)

May 18, 2026

Type Package

Title Hedonic and Multilateral Index Methods for Real Estate Price Statistics

Version 0.1.0

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 for real estate and other domains 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 GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Depends R (>= 4.4.0)

Imports dplyr, stats, assertthat, KFAS, stringr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

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 https://cran.r-universe.dev

Date/Publication 2025-04-25 15:10:01 UTC

RemoteUrl https://github.com/cran/cbsREPS

RemoteRef HEAD

RemoteSha 73aafb89ad83aacffc39a03319e015fccb789432

Contents

`calculate_fisher` *Calculate direct index according to the Fisher hedonic double imputation method*

Description

By the parameters 'dependent_variable', 'continue_variable' and 'categorical_variables' as regression model is compiled. With the model, a direct series of index figures is estimated by use of hedonic regression.

Usage

```
calculate_fisher(
  dataset,
  period_variable,
  dependent_variable,
  continuous_variables,
  categorical_variables,
  reference_period = NULL,
  number_of_observations = FALSE
)
```

Arguments

`dataset`	table with data (does not need to be a selection of relevant variables)
`period_variable`	variable in the table with periods
`dependent_variable`	usually the sale price
`continuous_variables`	vector with quality determining numeric variables (no dummies)
`categorical_variables`	vector with quality determining categorical variables (also dummies)
`reference_period`	period or group of periods that will be set to 100 (numeric/string)
`number_of_observations`	number of observations per period (default = TRUE)

Details

N.B.: the independent variables must be entered transformed (and ready) in the parameters. Hence, not: log(floor_area), but transform the variable in advance and then provide log_floor_area. This does not count for the dependent variable. This should be entered untransformed

Within the data, it is not neccesary to filter the data on relevant variables or complete records. This is taken care of in the function.

Value

table with index, imputation averages, number of observations and confidence intervals per period

Author(s)

Farley Ishaak

`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_laspeyres` *Calculate direct index according to the Laspeyres hedonic double imputation method*

Description

By the parameters 'dependent_variable', 'continue_variable' and 'categorical_variables' as regression model is compiled. With the model, a direct series of index figures is estimated by use of hedonic regression.

Usage

```
calculate_laspeyres(
  dataset,
  period_variable,
  dependent_variable,
  continuous_variables,
  categorical_variables,
  reference_period = NULL,
  index = TRUE,
  number_of_observations = FALSE,
  imputation = FALSE
)
```

Arguments

`dataset` table with data (does not need to be a selection of relevant variables)

`period_variable` variable in the table with periods

`dependent_variable` usually the sale price

`continuous_variables` vector with quality determining numeric variables (no dummies)

`categorical_variables` vector with quality determining categorical variables (also dummies)

`reference_period` period or group of periods that will be set to 100 (numeric/string)

`index` caprice index

`number_of_observations`
: number of observations per period (default = TRUE)

`imputation`
: display the underlying average imputation values? (default = FALSE)

Details

N.B.: the independent variables must be entered transformed (and ready) in the parameters. Hence, not: log(floor_area), but transform the variable in advance and then provide log_floor_area. This does not count for the dependent variable. This should be entered untransformed/

Within the data, it is not necessary to filter the data on relevant variables or complete records. This is taken care of in the function.

Value

table with index, imputation averages, number of observations and confidence intervals per period

Author(s)

Farley Ishaak

`calculate_paasche` *Calculate direct index according to the Paasche hedonic double imputation method*

Description

By the parameters 'dependent_variable', 'continue_variable' and 'categorical_variables' as regression model is compiled. With the model, a direct series of index figures is estimated by use of hedonic regression.

Usage

```
calculate_paasche(
  dataset,
  period_variable,
  dependent_variable,
  continuous_variables,
  categorical_variables,
  reference_period = NULL,
  index = TRUE,
  number_of_observations = FALSE,
  imputation = FALSE
)
```

Arguments

- `dataset` table with data (does not need to be a selection of relevant variables)
- `period_variable` variable in the table with periods
- `dependent_variable` usually the sale price
- `continuous_variables` vector with quality determining numeric variables (no dummies)
- `categorical_variables` vector with quality determining categorical variables (also dummies)
- `reference_period` period or group of periods that will be set to 100 (numeric/string)
- `index` caprice index
- `number_of_observations` number of observations per period (default = TRUE)
- `imputation` display the underlying average imputation values? (default = FALSE)

Details

N.B.: the independent variables must be entered transformed (and ready) in the parameters. Hence, not: log(floor_area), but transform the variable in advance and then provide log_floor_area. This does not count for the dependent variable. This should be entered untransformed

Within the data, it is not necessary to filter the data on relevant variables or complete records. This is taken care of in the function.

Value

table with index, imputation averages, number of observations and confidence intervals per period

Author(s)

Farley Ishaak

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

Description

Central hub function to calculate index figures using different methods.

Usage

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

Arguments

`method`	one of: "fisher", "laspeyres", "paasche", "hmts"
`dataset`	data frame with input data
`period_variable`	name of the variable indicating time periods
`dependent_variable`	usually the price
`continuous_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`	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) return detailed outputs? Default = FALSE
`index`	(Laspeyres/Paasche only) include index column? Default = TRUE
`imputation`	(Laspeyres/Paasche only) include imputation values? Default = FALSE

Value

A data.frame (or list for when method is HMTS with resting_points = TRUE)

Author(s)

Vivek Gajadhar

Examples

```
# Laspeyres index
Tbl_Laspeyres <- calculate_price_index(
  method = "laspeyres",
  dataset = data_constraxion,
  period_variable = "period",
  dependent_variable = "price",
  continuous_variables = "floor_area",
  categorical_variables = "neighbourhood_code",
  reference_period = 2015,
  number_of_observations = TRUE,
  imputation = FALSE
)
head(Tbl_Laspeyres)

# Paasche index
Tbl_Paasche <- calculate_price_index(
  method = "paasche",
  dataset = data_constraxion,
  period_variable = "period",
  dependent_variable = "price",
  continuous_variables = "floor_area",
  categorical_variables = "neighbourhood_code",
  reference_period = 2015,
  number_of_observations = TRUE,
  imputation = FALSE
)
head(Tbl_Paasche)

# Fisher index (geometric mean of Laspeyres and Paasche)
Tbl_Fisher <- calculate_price_index(
  method = "fisher",
  dataset = data_constraxion,
  period_variable = "period",
  dependent_variable = "price",
  continuous_variables = "floor_area",
  categorical_variables = "neighbourhood_code",
  reference_period = 2015,
  number_of_observations = TRUE
)
head(Tbl_Fisher)
```

`custom_update_function` *Default update function*

Description

This function is used in the function: calculate_trend_line_KFAS()

Usage

```
custom_update_function(params, model)
```

Arguments

`params`	startvalues
`model`	state space modelnumber

Value

Newmodel

Author(s)

Vivek Gajadhar

`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)
```

Index