Package ‘eurodata’

May 6, 2023

Type Package

Title Fast and Easy Eurostat Data Import and Search

Description Interface to Eurostat’s API (SDMX 2.1) with fast data.table-based import of data, labels, and metadata. On top of the core functionality, data search and data description/comparison functions are also provided.
Use <https://github.com/alekrutkowski/eurodata_codegen> — a point-and-click app for rapid and
easy generation of richly-commented R code — to import a Eurostat dataset or its subset (based on the eurodata::importData() function).

Version 1.7.0

Date 2023-05-05

License GPL-2

URL https://github.com/alekrutkowski/eurodata/

Encoding UTF-8

Depends R (>= 3.5.0)

Imports utils, Rcpp, magrittr, data.table, R.utils, xtable, memoise, stringr, xml2

Collate RcppExports.R const.R helpers.R core_import.R core_search.R extras.R

LinkingTo Rcpp

RoxygenNote 7.2.3

NeedsCompilation yes

Author Aleksander Rutkowski [aut, cre]

Maintainer Aleksander Rutkowski <alek.rutkowski@gmail.com>

Repository CRAN

Date/Publication 2023-05-06 10:00:02 UTC

R topics documented:

`as.EurostatDataList` *Coerce a data.frame to a EurostatDataList*

Description

Some manipulations of the `EurostatDataList` data.frame (imported with `importDataList`) e.g. filtering with package **dplyr** may remove the S3 class tag `EurostatDataList`. This function coerces it back to `EurostatDataList` after checking that the critical columns (`PCode`, `Dataset name`,`Link`) are present. This is useful if a user wants to print and browse this filtered data.frame as a specially formatted HTML table.

Usage

```
as.EurostatDataList(x, SearchCriteria = "", ...)
```

Arguments

`x`	A (most likely filtered subset of) `EurostatDataList` data.frame returned by `importDataList`.
`SearchCriteria`	A string describing the search criteria used for filtering/subsetting.
`...`	Additional arguments to be passed to or from methods (currently not used).

Value

A data.frame of S3 class `EurostatDataList`.

browseDataList *Search Eurostat datasets and see the result as a table in a browser*

Description

Search Eurostat datasets and see the result as a table in a browser

Usage

```
browseDataList(subs)
```

Arguments

subs An expression to be passed to `subset`. The column names of the table of datasets can be used – those with spaces should be backtick (‘) quoted. See the examples below. The names of the available columns are:

- `` `Data subgroup, level 0` ``
- `` `Data subgroup, level 1` ``
- `` `Data subgroup, level 2` ``
- `` `Data subgroup, level 3` ``
- `` `Data subgroup, level 4` ``
- `` `Data subgroup, level 5` ``
- `` `Data subgroup, level 6` ``
- `` `Data subgroup, level 7` ``
- `` `Dataset name` ``
- `` `Code` ``
- `` `Type` ``
- `` `Last update of data` ``
- `` `Last table structure change` ``
- `` `Data start` ``
- `` `Data end` ``
- `` `Link` ``

Value

- Side effect (via `print`) – a table opened in a browser via `browseURL`.
- Value – a list with:
  - criteria – a string, search criteria,
  - time – the time of the query,
  - df – a data.frame, imported via `importDataList` and filtered based on the conditions specified in `subs`.
  - html – a string, with the HTML code that generated the table in a browser.

Examples

```
## Not run:
browseDataList(grepl('servic',`Dataset name`))
browseDataList(grepl('bop',Code) & !grepl('its',Code))

## End(Not run)
## Not run:
browseDataList(grepl('GDP',`Dataset name`) &
grepl('main',`Dataset name`) &
   grepl('international',`Dataset name`) &
   !grepl('quarterly',`Dataset name`))
browseDataList(grepl('bop',Code) & grepl('its',Code))

## End(Not run)
```

compare — *Compare specific Eurostat datasets on the basis of information from Metabase*

Description

Compare specific Eurostat datasets on the basis of information from Metabase

Usage

```
compare(..., import_labels = TRUE, import_dim_labels = TRUE)
```

Arguments

`...` Two or more Eurostat dataset code names, e.g. "nama_10_gdp" or "bop_its6_det", as strings.

`import_labels` Boolean: should labels for the codes inside dimensions be imported. Default: `TRUE`.

`import_dim_labels` Boolean: should the dimensions (e.g. `geo`, `indic_is`, or `nace_r2`) be labelled with a descriptive name (via `importDimLabel`). Default: `TRUE`.

Value

A data.table with columns `Dim_name`, `Dim_name_label` (if `import_dim_labels=TRUE`), `Dim_val`, `Dim_val_label` (if `import_labels=TRUE`), and logical columns corresponding to the dataset names in `...` indicating in which dataset a given dimension and dimension value appears and in which it does not.

Examples

```
## Not run:
compare('nama_10_gdp', 'nama_10_pe')

## End(Not run)
```

`describe` *Describe a given Eurostat dataset on the basis of information from Metabase*

Description

Describe a given Eurostat dataset on the basis of information from Metabase

Usage

```
describe(
  EurostatDatasetCode,
  import_labels = !wide,
  wide = FALSE,
  import_dim_labels = TRUE
)
```

Arguments

`EurostatDatasetCode`
: A string with Eurostat dataset code name, e.g. `"nama_10_gdp"` or `"bop_its6_det"`. See e.g.: `https://ec.europa.eu/eurostat/databrowser/explore/all/all_themes` where, once you follow one of the "branches" of the "tree" of datasets, the dataset codes are in tiny grey font in square brackets just under the full names of the datasets (the names are in navy blue and preceded by a cube icon).

`import_labels`
: Boolean: should labels for the codes inside dimensions be imported. Default: if `wide` is `FALSE` then `import_labels` is TRUE and vice versa.

`wide`
: Boolean: should each dimension be compressed to one row and all values within each dimension to a single, comma-separated string. Default: `FALSE`.

`import_dim_labels`
: Boolean: should the dimensions (e.g. `geo`, `indic_is`, or `nace_r2`) be labelled with a descriptive name (via `importDimLabel`). Default: `TRUE`.

Value

A data.table with columns `Dim_name`, `Dim_name_label` (if `import_dim_labels=TRUE`), either `Dim_val` (if `wide=FALSE`) or `Dim_values` (if `wide=TRUE`), `Dim_val_label` (if `import_labels=TRUE`), and a column with a name = `EurostatDatasetCode` with all its values = `TRUE`.

Examples

```
## Not run:
describe('nama_10_gdp')

## End(Not run)
```

---

find — *Search Eurostat datasets and see the result as text*

Description

A tool for a quick ad-hoc search.

Usage

```
find(...)
```

Arguments

`...`	A series of unquoted words to be searched either in Eurostat dataset codes or in dataset full names. All words not preceded by minus (-) will be linked with logical AND; all words preceded by a minus entail exclusion (logical NOT), a bit like in Google search. It is possible to search also with phrases that include spaces – in such a case the phrases should be quoted. Partial word/phrase match is applied. See the examples below.

Value

- Side effect (via `print`) – a text report file opened via `file.show`.
- Value – a list with:
  - criteria – a string, search criteria,
  - time – the time of the query,
  - df – a data.frame, imported via `importDataList` and filtered based on the conditions specified in `...`,
  - report – a string, with the text report.

Examples

```
## Not run:
find(bop, its)
find(bop,-ybk,its)
find(nama_)
find(nama,10,64)
find('economic indic')

## End(Not run)
```

`importData` *Download and import a Eurostat dataset*

Description

Download and import a Eurostat dataset

Usage

```
importData(EurostatDatasetCode, filters = NULL)
```

Arguments

`EurostatDatasetCode`
: A string (upper/lower-case difference is not relevant) with Eurostat dataset code name, e.g. `nama_10_gdp` or `bop_its6_det`. See `https://ec.europa.eu/eurostat/databrowser/explore/all/all_themes` to find a dataset code – the dataset codes are in tiny font in square brackets.

`filters`
: Optional: a list of atomic vectors. The names of the elements of the list should correspond to the names of the dimensions of the dataset (defined in `EurostatDatasetCode`), e.g. `geo`, `nace_r2`, `indic_esb` etc. The elements of each vector in that list should correspond to each respective dimension's values available in the dataset. Only these dimension values will be downloaded. For `TIME_PERIOD` it's enough to provide 1 or 2 values – the lowest one will be used as a start of the data period and the highest as the end of the data period downloaded. Use `filters` if you need only a few dimension values as it will be faster than downloading the full dataset.

Value

A Eurostat dataset as a 'flat' data.frame. A 'flat' dataset has all numeric values in one column, with each row representing one of the available combinations of all dimensions (e.g. if dimensions are: countries, years, sectors, and indicators, there can be a row for value added in retail in Germany in 2013).

Examples

```
## Not run:
# Full dataset import:
importData('nama_10_gdp')
# Import only a subset of a dataset:
importData('bop_its6_det',
           filters = list(geo=c('AT','BG'),
                          TIME_PERIOD=2014:2020,
                          bop_item='SC'))

## End(Not run)
```

importDataList *Import and reshape Eurostat inventory of datasets*

Description

Import and reshape Eurostat inventory of datasets

Usage

```
importDataList()
```

Value

The imported data.frame reflects the hierarchical structure of datasets (see the columns `Data subgroup, level 0`, `Data subgroup, level 1`, `Data subgroup, level 2`, etc.). It is tagged with S3 class `EurostatDataList`.

Examples

```
## Not run:
importDataList()

## End(Not run)
```

importDimLabel *Import Eurostat label (description) of a given dimension code*

Description

Import the appropriate description file for the selected Eurostat dimension, e.g. for `"geo"` it is `"Geopolitical entity (reporting)"`, for `"nace_r2"` it is `"Classification of economic activities - NACE Rev. 2"`, for `"indic_sb"` it is `"Economical indicator for structural business statistics"` etc. Click on "Code lists" just under "Apply download operations on" at `https://ec.europa.eu/eurostat/databrowser/bulk?lang=en` for the list of all codes. Each description is imported from inside the XML file (via the path: *m:Structure / m:Structures / s:Codelists / s:Codelist / c:Name xml:lang="en"*) from the respective URL, e.g. for `"geo"` it is `https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/codelist/ESTAT/GEO`.

Usage

```
importDimLabel(EurostatDimCode)
```

Arguments

`EurostatDimCode`
A string – the code name of the Eurostat dimension, e.g. `"geo"` or `"nace_r2"` or `"indic_sb"`, etc.

Value

A character vector of length 1: the label/description of EurostatDimCode.

Examples

```
## Not run:
importDimLabel('nace_r2')

## End(Not run)
```

---

importLabels *Import Eurostat code list: labels (descriptions) for a given dimension code*

Description

Import the appropriate 'code list' from for the selected Eurostat dimension, e.g. geo (countries or other geographic entities), nace_r2 (sectors), indic_sb (indicators), etc.

Usage

```
importLabels(EurostatDimCode)
```

Arguments

EurostatDimCode
A string – the code name of the Eurostat dimension, e.g. geo or nace_r2 or indic_sb.

Value

A data.frame with 2 columns: codes (with a name determined by EurostatDimCode) and corresponding labels (named with suffix _labels).

Examples

```
## Not run:
importLabels('nace_r2')

## End(Not run)
```

importMetabase *Import Eurostat “Metabase”*

Description

The Eurostat “Metabase” shows which datasets contain which dimensions (where a dimension is e.g. `geo` or `nace_r2` or `indic_sb`) and, within each dataset and dimension, which codes (e.g. which countries for the `geo` dimension).

Usage

```
importMetabase()
```

Value

The imported data.frame which reflects the hierarchical structure described above. It is a ‘flat’ data.frame with 3 columns, where each row corresponds to the combination of:

- `Code` – Eurostat dataset code names, e.g. `"nama_10_a64"`
- `Dim_name` – Eurostat dimension code names, e.g. `"nace_r2"`
- `Dim_val` – Eurostat dimension code values, e.g. `"EU28"` if `Dim_name` is `"geo"`; not to be confused with the actual numeric values in the actual datasets

Examples

```
## Not run:
importMetabase()

## End(Not run)
```

Index