Package ‘eurostat’

February 11, 2020

Type Package

Title Tools for Eurostat Open Data

Date 2020-02-11

Version 3.6.1

Encoding UTF-8

Maintainer Leo Lahti <leo.lahti@iki.fi>

MailingList rOpenGov <ropengov-forum@googlegroups.com>

Description Tools to download data from the Eurostat database <http://ec.europa.eu/eurostat> together with search and manipulation utilities.

License BSD_2_clause + file LICENSE

Depends methods, R (>= 3.1.0)

Imports broom, classInt, countrycode, curl, dplyr, httr, jsonlite, lubridate, RColorBrewer, readr, RefManageR, sf, sp, stringi, stringr, tibble, tidyr, tidyselect

Suggests Cairo, ggplot2, knitr, mapproj, plotrix, rmarkdown, roxygen2, rsdmx, rvest, testthat, tmap, usethis

LazyData true

URL https://ropengov.github.io/eurostat

BugReports https://github.com/ropengov/eurostat/issues

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

RoxygenNote 7.0.2

Author Leo Lahti [aut, cre] (<https://orcid.org/0000-0001-5537-637X>),
Janne Huovari [aut],
Markus Kainu [aut],
Przemyslaw Biecek [aut],
Joona Lehtomaki [ctb],
Francois Briatte [ctb],
Oliver Reiter [ctb]

Date/Publication 2020-02-11 17:50:03 UTC

R topics documented:

eurostat-package *R Tools for Eurostat open data*

Description

Brief summary of the eurostat package

Details

Package:	eurostat
Type:	Package
Version:	See sessionInfo() or DESCRIPTION file
Date:	2014-2017
License:	BSD_2_clause + LICENSE

LazyLoad:	yes

R Tools for Eurostat Open Data

Author(s)

Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek `<ropengov-forum@googlegroups.com>`
`https://ropengov.github.io/eurostat`

References

See citation("eurostat") `https://ropengov.github.io/eurostat`

Examples

```
library(eurostat)
```

`add_nuts_level` *Add the statistical aggregation level to data frame*

Description

Eurostat regional statistics contain country, and various regional level information. In many cases, for example, when mapping, it is useful to filter out national level data from NUTS2 level regional data, for example.

Usage

```
add_nuts_level(dat, geo_labels = "geo")
```

Arguments

`dat`	A data frame or tibble returned by `get_eurostat`.
`geo_labels`	A geographical label, defaults to geo.

Value

a new numeric variable nuts_level with the numeric value of NUTS level 0 (country), 1 (greater region), 2 (region), 3 (small region).

Author(s)

Daniel Antal

Examples

```
{
  dat = data.frame (
       geo    = c("FR", "IE04", "DEB1C"),
       values = c(1000, 23, 12)
  )

  add_nuts_level(dat)
}
```

check_access_to_data *Check access to ec.europe.eu*

Description

Check if R has access to resources at http://ec.europa.eu

Usage

```
check_access_to_data()
```

Value

a logical.

Author(s)

Markus Kainu <markus.kainu@kapsi.fi>

Examples

```
 ## Not run:
  check_access_to_data()

## End(Not run)
```

clean_eurostat_cache *Clean Eurostat Cache*

Description

Delete all .rds files from the eurostat cache directory. See get_eurostat for more on cache.

Usage

```
clean_eurostat_cache(cache_dir = NULL)
```

Arguments

`cache_dir`	A path to cache directory. If `NULL` (default) tries to clean default temporary cache directory.

Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari and Markus Kainu `<ropengov-forum@googlegroups.com>` `http://github.com/ropengov/eurostat`

Examples

```
clean_eurostat_cache()
```

`cut_to_classes` *Cuts the Values Column into Classes and Polishes the Labels*

Description

Categorises a numeric vector into automatic or manually defined categories. and polishes the labels ready for used in mapping with `merge_with_geodata` function and ggplot2.

Usage

```
cut_to_classes(
  x,
  n = 5,
  style = "equal",
  manual = FALSE,
  manual_breaks = NULL,
  decimals = 0,
  nodata_label = "No data"
)
```

Arguments

`x`	A numeric vector, eg. `values` variable in data returned by `get_eurostat`
`n`	A numeric. number of classes/categories
`style`	Chosen style: one of "fixed", "sd", "equal", "pretty", "quantile", "kmeans", "hclust", "bclust", "fisher", or "jenks"
`manual`	Logical. If manual breaks are being used
`manual_breaks`	Numeric vector with manual threshold values
`decimals`	Number of decimals to include with labels
`nodata_label`	String. Text label for NA category.

Value

a factor.

Author(s)

Markus Kainu <markuskainu@gmail.com>

Examples

```
 ## Not run:
   #lp <- get_eurostat("nama_aux_lp")
   lp <- get_eurostat("nama_10_lp_ulc")
   lp$class <- cut_to_classes(lp$values, n=5, style="equal", decimals=1)

## End(Not run)
```

dic_order *Order of Variable Levels from Eurostat Dictionary.*

Description

Orders the factor levels.

Usage

```
dic_order(x, dic, type)
```

Arguments

x a variable (code or labelled) to get order for.

dic a name of the dictionary. Correspond a variable name in the data_frame from `get_eurostat`. Can be also data_frame from `get_eurostat_dic`.

type a type of the x. Could be `code` or `label`.

Details

Some variables, like classifications, have logical or conventional ordering. Eurostat data tables are nor neccessary ordered in this order. The function `dic_order` get the ordering from Eurostat classifications dictionaries. The function `label_eurostat` can also order factor levels of labels with argument `eu_order = TRUE`.

Value

A numeric vector of orders.

Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari and Markus Kainu <ropengov-forum@googlegroups.com>
http://github.com/ropengov/eurostat

`eurostat_geodata_60_2016` *Geospatial data of Europe from Gisco in 1:60 million scale from year 2016*

Description

Geospatial data of Europe from Gisco in 1:60 million scale from year 2016

Usage

```
eurostat_geodata_60_2016
```

Format

sf

id Country code in the Eurostat database

CNTRY_CODE Country code

NUTS_NAME NUTS name in local language

LEVL_CODE NUTS code

FID Country code

NUTS_ID NUTS code

geometry geospatial information

geo NUTS code

Source

`http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-unit`

`eurotime2date` *Date Conversion from Eurostat Time Format*

Description

Date conversion from Eurostat time format. A function to convert Eurostat time values to objects of class `Date` representing calendar dates.

Usage

```
eurotime2date(x, last = FALSE)
```

Arguments

x	a charter string with time information in Eurostat time format.
last	a logical. If FALsE (default) the date is the first date of the period (month, quarter or year). If TRUE the date is the last date of the period.

Value

an object of class Date.

Author(s)

Janne Huovari <janne.huovari@ptt.fi>

Examples

```
## Not run:
   na_q <- get_eurostat("namq_10_pc", time_format = "raw")
   na_q$time <- eurotime2date(x = na_q$time)

   un <- get_eurostat("une_rt_m", time_format = "raw")
   un$time <- eurotime2date(x = un$time)

   na_a <- get_eurostat("nama_10_pc", time_format = "raw")
   na_a$time <- eurotime2date(x = na_a$time)

   eur_d <- get_eurostat("ert_bil_eur_d", time_format = "raw")
   eur_d$time <- eurotime2date(x = eur_d$time)

## End(Not run)
```

eurotime2num *Conversion of Eurostat Time Format to Numeric*

Description

A conversion of a Eurostat time format to numeric.

Usage

```
eurotime2num(x)
```

Arguments

x	a charter string with time information in Eurostat time format.

Details

Bi-annual, quarterly and monthly data is presented as fraction of the year in beginning of the period. Conversion of daily data is not supported.

Value

see as.numeric.

Author(s)

Janne Huovari <janne.huovari@ptt.fi>

Examples

```
## Not run:
   na_q <- get_eurostat("namq_10_pc", time_format = "raw")
   na_q$time <- eurotime2num(x = na_q$time)

   un <- get_eurostat("une_rt_m", time_format = "raw")
   un$time <- eurotime2num(x = un$time)

   na_a <- get_eurostat("nama_10_pc", time_format = "raw")
   na_a$time <- eurotime2num(x = na_a$time)

## End(Not run)
```

---

eu_countries *Countries and Country Codes*

Description

Countries and country codes in EU, Euro area, EFTA and EU candidate countries.

Usage

```
eu_countries

ea_countries

efta_countries

eu_candidate_countries
```

Format

A data_frame:

code Country code in the Eurostat database

name Country name in English

label Country name in the Eurostat database

Source

http://ec.europa.eu/eurostat/statistics-explained/index.php/Tutorial:Country_codes_and_protocol_order, http://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Euro_area

get_bibentry *Create A Data Bibliography*

Description

Creates a bibliography from selected Eurostat data files, including last Eurostat update, URL access data, and optional keywords set by the user.

Usage

```
get_bibentry(code, keywords = NULL, format = "Biblatex")
```

Arguments

code	A Eurostat data code or a vector of Eurostat data codes as character or factor.
keywords	A list of keywords to be added to the entries. Defaults to NULL.
format	Default is 'Biblatex', alternatives are 'bibentry' or 'Bibtex' (not case sensitive.)

Value

a bibentry, Bibtex or Biblatex object.

Author(s)

Daniel Antal, Przemyslaw Biecek

Examples

```
 ## Not run:
   my_bibliography <- get_bibentry (
        code = c("tran_hv_frtra", "t2020_rk310","tec00001") ,
        keywords = list ( c("railways", "freight", "transport"),
                          c("railways", "passengers", "modal split") ),
        format = "Biblatex" )

   # readLines ( my_bibliograhy, "eurostat_data.bib")

## End(Not run)
```

get_eurostat *Read Eurostat Data*

Description

Download data sets from Eurostat https://ec.europa.eu/eurostat/.

Usage

```
get_eurostat(
  id,
  time_format = "date",
  filters = "none",
  type = "code",
  select_time = NULL,
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  stringsAsFactors = default.stringsAsFactors(),
  keepFlags = FALSE,
  ...
)
```

Arguments

`id` A code name for the dataset of interest. See `search_eurostat` or details for how to get code.

`time_format` a string giving a type of the conversion of the time column from the eurostat format. A "date" (default) convers to a `Date` with a first date of the period. A "date_last" convers to a `Date` with a last date of the period. A "num" convers to a numeric and "raw" does not do conversion. See `eurotime2date` and `eurotime2num`.

`filters` a "none" (default) to get a whole dataset or a named list of filters to get just part of the table. Names of list objects are Eurostat variable codes and values are vectors of observation codes. If `NULL` the whole dataset is returned via API. More on details. See more on filters and limitations per query via API from for `get_eurostat_json`.

`type` A type of variables, "code" (default) or "label".

`select_time` a character symbol for a time frequence or NULL, which is used by default as most datasets have just one time frequency. For datasets with multiple time frequencies, select the desired time format with: Y = annual, S = semi-annual, Q = quarterly, M = monthly. For all frequencies in same data frame `time_format = "raw"` should be used.

`cache` a logical whether to do caching. Default is `TRUE`. Affects only queries from the bulk download facility.

`update_cache` a logical whether to update cache. Can be set also with options(eurostat_update = TRUE)

`cache_dir` a path to a cache directory. The directory must exist. The `NULL` (default) uses and creates 'eurostat' directory in the temporary directory from `tempdir`. The directory can also be set with `option` eurostat_cache_dir.

`compress_file` a logical whether to compress the RDS-file in caching. Default is `TRUE`.

`stringsAsFactors`
if `TRUE` (the default) variables are converted to factors in original Eurostat order. If `FALSE` they are returned as a character.

`keepFlags` a logical whether the flags (e.g. "confidential", "provisional") should be kept in a separate column or if they can be removed. Default is `FALSE`. For flag values see: `http://ec.europa.eu/eurostat/data/database/information`. Also possible non-real zero "0n" is indicated in flags column. Flags are not available for eurostat API, so `keepFlags` can not be used with a `filters`.

`...` further argument for `get_eurostat_json`.

Details

Data sets are downloaded from the Eurostat bulk download facility or from The Eurostat Web Services JSON API. If only the table `id` is given, the whole table is downloaded from the bulk download facility. If also `filters` are defined the JSON API is used.

The bulk download facility is the fastest method to download whole datasets. It is also often the only way as the JSON API has limitation of maximum 50 sub-indicators at time and whole datasets usually exceeds that. Also, it seems that multi frequency datasets can only be retrived via bulk download facility and the `select_time` is not available for JSON API method.

If your connection is thru a proxy, you probably have to set proxy parameters to use JSON API, see `get_eurostat_json`.

By default datasets from the bulk download facility are cached as they are often rather large. Caching is not (currently) possible for datasets from JSON API. Cache files are stored in a temporary directory by default or in a named directory if cache_dir or option eurostat_cache_dir is defined. The cache can be emptied with `clean_eurostat_cache`.

The `id`, a code, for the dataset can be searched with the `search_eurostat` or from the Eurostat database `http://ec.europa.eu/eurostat/data/database`. The Eurostat database gives codes in the Data Navigation Tree after every dataset in parenthesis.

Value

a tibble. One column for each dimension in the data, the time column for a time dimension and the values column for numerical values. Eurostat data does not include all missing values and a treatment of missing values depend on source. In bulk download facility missing values are dropped if all dimensions are missing on particular time. In JSON API missing values are dropped only if all dimensions are missing on all times. The data from bulk download facility can be completed for example with `complete`.

Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari and Markus Kainu <ropengov-forum@googlegroups.com>
http://github.com/ropengov/eurostat

See Also

search_eurostat, label_eurostat

Examples

```
k <- get_eurostat("nama_10_lp_ulc")
## Not run:
k <- get_eurostat("nama_10_lp_ulc", time_format = "num")
k <- get_eurostat("nama_10_lp_ulc", update_cache = TRUE)
dir.create(file.path(tempdir(), "r_cache"))
k <- get_eurostat("nama_10_lp_ulc",
                  cache_dir = file.path(tempdir(), "r_cache"))
options(eurostat_update = TRUE)
k <- get_eurostat("nama_10_lp_ulc")
options(eurostat_update = FALSE)
options(eurostat_cache_dir = file.path(tempdir(), "r_cache"))
k <- get_eurostat("nama_10_lp_ulc")
k <- get_eurostat("nama_10_lp_ulc", cache = FALSE)
k <- get_eurostat("avia_gonc", select_time = "Y", cache = FALSE)

dd <- get_eurostat("nama_10_gdp",
                     filters = list(geo = "FI",
                                    na_item = "B1GQ",
                                    unit = "CLV_I10"))

## End(Not run)
```

get_eurostat_dic *Download Eurostat Dictionary*

Description

Download a Eurostat dictionary.

Usage

```
get_eurostat_dic(dictname, lang = "en")
```

Arguments

dictname	A character, dictionary for the variable to be downloaded.
lang	A character, language code. Options: "en" (default) / "fr" / "de".

Details

For given coded variable from Eurostat https://ec.europa.eu/eurostat/. The dictionaries link codes with human-readable labels. To translate codes to labels, use label_eurostat.

Value

tibble with two columns: code names and full names.

Author(s)

Przemyslaw Biecek and Leo Lahti <leo.lahti@iki.fi>. Thanks to Wietse Dol for contributions.

References

See citation("eurostat").

See Also

label_eurostat, get_eurostat, search_eurostat.

Examples

```
## Not run:
      tmp <- get_eurostat_dic("crop_pro")
      head(tmp)
      tmp <- get_eurostat_dic("crop_pro", lang = "fr")

## End(Not run)
```

---

get_eurostat_geospatial

Download Geospatial Data from GISGO

Description

Downloads either a simple features (sf), SpatialPolygonDataFrame or a data_frame preprocessed using broom::tidy().

Usage

```
get_eurostat_geospatial(
  output_class = "sf",
  resolution = "60",
  nuts_level = "all",
  year = "2016",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL
)
```

Arguments

`output_class` A string. Class of object returned, either `sf` `simple features`, `df` (`data_frame`) or `spdf` (`SpatialPolygonDataFrame`)

`resolution` Resolution of the geospatial data. One of "60" (1:60million), "20" (1:20million) "10" (1:10million) "03" (1:3million) or "01" (1:1million).

`nuts_level` Level of NUTS classification of the geospatial data. One of "0", "1", "2", "3" or "all" (mimics the original behaviour)

`year` NUTS release year. One of "2003", "2006", "2010", "2013" or "2016"

`cache` a logical whether to do caching. Default is `TRUE`. Affects only queries from the bulk download facility.

`update_cache` a locigal whether to update cache. Can be set also with options(eurostat_update = TRUE)

`cache_dir` a path to a cache directory. The directory have to exist. The `NULL` (default) uses and creates 'eurostat' directory in the temporary directory from `tempdir`. Directory can also be set with `option` eurostat_cache_dir.

Details

The data source URL is `http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units`.

Value

a sf, data_frame or SpatialPolygonDataFrame.

Author(s)

Markus Kainu <markuskainu@gmail.com>

Examples

```
sf <- get_eurostat_geospatial(output_class = "sf", resolution = "60", nuts_level = "all")
df <- get_eurostat_geospatial(output_class = "df", resolution = "20", nuts_level = "0")
spdf <- get_eurostat_geospatial(output_class = "spdf", resolution = "10", nuts_level = "3")
```

`get_eurostat_json` *Get Data from Eurostat API in JSON*

Description

Retrieve data from Eurostat API in JSON format.

Usage

```
get_eurostat_json(
  id,
  filters = NULL,
  type = c("code", "label", "both"),
  lang = c("en", "fr", "de"),
  stringsAsFactors = default.stringsAsFactors(),
  ...
)
```

Arguments

`id` A code name for the dataset of interested. See the table of contents of eurostat datasets for more details.

`filters` A named list of filters. Names of list objects are Eurostat variable codes and values are vectors of observation codes. If `NULL` (default) the whole dataset is returned. See details for more on filters and limitations per query.

`type` A type of variables, "code" (default), "label" or "both". The "both" will return a data_frame with named vectors, labels as values and codes as names.

`lang` A language used for metadata (en/fr/de).

`stringsAsFactors` if `TRUE` (the default) variables are converted to factors in original Eurostat order. If `FALSE` they are returned as a character.

`...` Other arguments passed on to `GET`. For example a proxy parameters, see details. .

Details

Data to retrieve from The Eurostat Web Services can be specified with filters. Normally, it is better to use JSON query through `get_eurostat`, than to use `get_eurostat_json` directly.

Queries are limited to 50 sub-indicators at a time. A time can be filtered with fixed "time" filter or with "sinceTimePeriod" and "lastTimePeriod" filters. A `sinceTimePeriod = 2000` returns observations from 2000 to a last available. A `lastTimePeriod = 10` returns a 10 last observations.

To use a proxy to connect, a `use_proxy` can be passed to `GET`. For example `get_eurostat_json(id,filters,config = httr::use_proxy(url,port,username,password))`.

Value

A dataset as a data_frame.

Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari and Markus Kainu <ropengov-forum@googlegroups.com>
http://github.com/ropengov/eurostat

Examples

```
 ## Not run:
   tmp <- get_eurostat_json("cdh_e_fos")
   nama_gdp_c has been discontinued since 2/2018 and this example has ceased working.
   yy <- get_eurostat_json(id = "nama_gdp_c", filters = list(geo=c("EU28", "FI"),
                                                      unit="EUR_HAB",
                                                      indic_na="B1GM"))

## End(Not run)
```

get_eurostat_raw *Download Data from Eurostat Database*

Description

Download data from the eurostat database.

Usage

```
get_eurostat_raw(id)
```

Arguments

id	A code name for the dataset of interested. See the table of contents of eurostat datasets for more details.

Details

Data is downloaded from http://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing and transformed into tabular format.

Value

A dataset in tibble format. First column contains comma separated codes of cases. Other columns usually corresponds to years and column names are years with preceding X. Data is in character format as it contains values together with eurostat flags for data.

Author(s)

Przemyslaw Biecek, Leo Lahti and Janne Huovari

References

see citation("eurostat")

See Also

get_eurostat.

Examples

```
   ## Not run:
     tmp <- eurostat:::get_eurostat_raw("educ_iste")
     head(tmp)

## End(Not run)
```

get_eurostat_toc *Download Table of Contents of Eurostat Data Sets*

Description

Download table of contents (TOC) of eurostat datasets.

Usage

```
get_eurostat_toc()
```

Details

The TOC is downloaded from http://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?sort=1&file=table_of_contents_en.txt. The values in column 'code' should be used to download a selected dataset.

Value

A tibble with eight columns

- titleThe name of dataset of theme
- codeThe codename of dataset of theme, will be used by the eurostat and get_eurostat_raw functions.
- typeIs it a dataset, folder or table.
- last.update.of.data, last.table.structure.change, data.start, data.endDates.

Author(s)

Przemyslaw Biecek and Leo Lahti <ropengov-forum@googlegroups.com>

References

See citation("eurostat").

See Also

get_eurostat, search_eurostat.

Examples

```
## Not run: tmp <- get_eurostat_toc(); head(tmp)
```

`harmonize_country_code` *Harmonize Country Code*

Description

The European Commission and the Eurostat generally uses ISO 3166-1 alpha-2 codes with two exceptions: EL (not GR) is used to represent Greece, and UK (not GB) is used to represent the United Kingdom. This function turns country codes into to ISO 3166-1 alpha-2.

Usage

```
harmonize_country_code(x)
```

Arguments

`x` A character or a factor vector of eurostat countycodes.

Value

a vector.

Author(s)

Janne Huovari <`janne.huovari@ptt.fi`>

Examples

```
 ## Not run:
   lp <- get_eurostat("nama_10_lp_ulc")
   lp$geo <- harmonize_country_code(lp$geo)

## End(Not run)
```

`harmonize_geo_code` *Harmonize NUTS region codes that changed with the* `NUTS2016` *definition*

Description

Eurostat mixes `NUTS2013` and `NUTS2016` geographic label codes in the `'geo'` column, which creates time-wise comparativity issues. This function checks if you data is affected by this problem and gives information on what to do.

Usage

```
harmonize_geo_code(dat)
```

Arguments

dat	A Eurostat data frame downloaded with get_eurostat

Value

An augmented data frame that explains potential problems and possible solutions.

Author(s)

Daniel Antal

Examples

```
 ## Not run:
   dat <- eurostat::tgs00026
   harmonize_geo_code(dat)

## End(Not run)
```

label_eurostat *Get Eurostat Codes*

Description

Get definitions for Eurostat codes from Eurostat dictionaries.

Usage

```
label_eurostat(
  x,
  dic = NULL,
  code = NULL,
  eu_order = FALSE,
  lang = "en",
  countrycode = NULL,
  countrycode_nomatch = NULL,
  custom_dic = NULL,
  fix_duplicated = FALSE
)

label_eurostat_vars(x, lang = "en")

label_eurostat_tables(x, lang = "en")
```

Arguments

`x` A character or a factor vector or a data_frame.

`dic` A string (vector) naming eurostat dictionary or dictionaries. If `NULL` (default) dictionry names taken from column names of the data_frame.

`code` For data_frames names of the column for which also code columns should be retained. The suffix "_code" is added to code column names.

`eu_order` Logical. Should Eurostat ordering used for label levels. Affects only factors.

`lang` A character, code for language. Available are "en" (default), "fr" and "de".

`countrycode` A `NULL` or a name of the coding scheme for the `countrycode` to label "geo" variable with countrycode-package. It can be used to convert to short and long country names in many different languages. If `NULL` (default) eurostat dictionary is used instead.

`countrycode_nomatch`
What to do when using the countrycode to label a "geo" and countrycode fails to find a match, for example other than country codes like EU28. the original code is used with a `NULL` (default), eurostat dictionary label is used with "eurostat", and `NA` is used with NA.

`custom_dic` a named vector or named list of named vectors to give an own dictionary for (part of) codes. Names of the vector should be codes and values labels. List can be used to spesify dictonaries and then list names should be dictionary codes.

`fix_duplicated` A logical. If TRUE, the code is added to the duplicated label values. If FALSE (default) error is given if labelling produce duplicates.

Details

A character or a factor vector of codes returns a corresponding vector of definitions. `label_eurostat` labels also data_frames from `get_eurostat`. For vectors a dictionary name have to be supplied. For data_frames dictonary names are taken from column names. "time" and "values" columns are returned as they were, so you can supply data_frame from `get_eurostat` and get data_frame with definitions instead of codes.

Some Eurostat dictionaries includes dublicated labels. By default dublicated labels cause an error, but they can be fixed automatically with `fix_duplicated = TRUE`.

Value

a vector or a data_frame.

Functions

- `label_eurostat_vars`: Get definitions for variable (column) names. For objects other than characters or factors definitions are get for names.
- `label_eurostat_tables`: Get definitions for table names

Author(s)

Janne Huovari <`janne.huovari@ptt.fi`>

Examples

```
 ## Not run:
   lp <- get_eurostat("nama_10_lp_ulc")
   lpl <- label_eurostat(lp)
   str(lpl)
   lpl_order <- label_eurostat(lp, eu_order = TRUE)
   lpl_code <- label_eurostat(lp, code = "unit")
   label_eurostat_vars(names(lp))
   label_eurostat_tables("nama_10_lp_ulc")
   label_eurostat(c("FI", "DE", "EU28"), dic = "geo")
   label_eurostat(c("FI", "DE", "EU28"), dic = "geo", custom_dic = c(DE = "Germany"))
   label_eurostat(c("FI", "DE", "EU28"), dic = "geo", countrycode = "country.name",
                  custom_dic = c(EU28 = "EU"))
   label_eurostat(c("FI", "DE", "EU28"), dic = "geo", countrycode = "country.name")
   # In Finnish
   label_eurostat(c("FI", "DE", "EU28"), dic = "geo", countrycode = "cldr.short.fi")

## End(Not run)
```

`nuts_correspondence` *Correspondence Table NUTS2013-NUTS2016*

Description

A tidy version of the Eurostat correspondence for NUTS1 and NUTS2 territorial units.

Usage

```
nuts_correspondence
```

Format

A data_frame:

code13 The geographical code of the territory in the NUTS2013 definition

code16 The geographical code of the territory in the NUTS2016 definition

name Name of the territorial unit in the Eurostat database

nuts_level Aggregation level, i.e. 0=national, 1,2,3 for smaller regions.

change Change with the region, or 'unchanged'

resolution How can the comparison made between NUTS2013 and NUTS2016 units made, if possible.

Source

https://ec.europa.eu/eurostat/web/nuts/history, https://ec.europa.eu/eurostat/documents/345175/629341/NUTS2013-NUTS2016.xlsx

recode_to_nuts_2013 *Recode geo labels and rename regions from NUTS2016 to NUTS2013*

Description

Eurostat mixes NUTS2013 and NUTS2016 geographic label codes in the `'geo'` column, which creates time-wise comparativity issues. This function recodes the observations where only the coding changed, and marks discontinued regions, and other regions which may or may not be somehow compared to the historic `'NUTS2013'` boundaries.

Usage

```
recode_to_nuts_2013(dat)
```

Arguments

`dat`	A Eurostat data frame downloaded with `get_eurostat`.

Value

An augmented and potentially relabelled data frame which contains all formerly `'NUTS2013'` definition geo labels in the `'NUTS2016'` vocabulary when only the code changed, but the boundary did not. It also contains some information on other geo labels that cannot be brought to the current `'NUTS2013'` definition. Furthermore, when the official name of the region changed, it will use the new name (if the otherwise the region boundary did not change.) If not called before, the function will use the helper function `harmonize_geo_code`

Author(s)

Daniel Antal

Examples

```
test_regional_codes <- data.frame (
  geo = c("FRB", "FRE", "UKN02", "IE022", "FR243", "FRB03"),
  time = c(rep(as.Date ("2014-01-01"), 5), as.Date("2015-01-01")),
  values = c(1:6),
  control = c("Changed from NUTS2 to NUTS1",
              "New region NUTS2016 only",
              "Discontinued region NUTS2013",
              "Boundary shift NUTS2013",
              "Recoded in NUTS2013",
              "Recoded in NUTS2016"
  ))

recode_to_nuts_2013(test_regional_codes)
```

recode_to_nuts_2016 *Recode geo labels and rename regions from NUTS2013 to NUTS2016*

Description

Eurostat mixes NUTS2013 and NUTS2016 geographic label codes in the `'geo'` column, which creates time-wise comparativity issues. This function recodes the observations where only the coding changed, and marks discontinued regions, and other regions which may or may not be somehow compared to current `'NUTS2016'` boundaries.

Usage

```
recode_to_nuts_2016(dat)
```

Arguments

`dat`	A Eurostat data frame downloaded with `get_eurostat`.

Value

An augmented and potentially relabelled data frame which contains all formerly `'NUTS2013'` definition geo labels in the `'NUTS2016'` vocabulary when only the code changed, but the boundary did not. It also contains some information on other geo labels that cannot be brought to the current `'NUTS2016'` definition. Furthermore, when the official name of the region changed, it will use the new name (if the otherwise the region boundary did not change.) If not called before, the function will use the helper function `harmonize_geo_code`

Author(s)

Daniel Antal

Examples

```
test_regional_codes <- data.frame (
  geo = c("FRB", "FRE", "UKN02", "IE022", "FR243", "FRB03"),
  time = c(rep(as.Date ("2014-01-01"), 5), as.Date("2015-01-01")),
  values = c(1:6),
  control = c("Changed from NUTS2 to NUTS1",
              "New region NUTS2016 only",
              "Discontinued region NUTS2013",
              "Boundary shift NUTS2013",
              "Recoded in NUTS2013",
              "Recoded in NUTS2016"
  ))

recode_to_nuts_2016(test_regional_codes)
```

`regional_changes_2016` *Changes in regional boundaries NUTS2013-NUTS2016*

Description

A comparison of regional boundaries, codes, and explanation for the change in a data frame, based on the Eurostat correspondence table.

Usage

```
regional_changes_2016
```

Format

A data_frame:

code13 The geographical code of the territory in the NUTS2013 definition

code16 The geographical code of the territory in the NUTS2016 definition

name Name of the territorial unit in the Eurostat database

nuts_level Aggregation level, i.e. 0=national, 1,2,3 for smaller regions.

change Change with the region, or 'unchanged'

Source

https://ec.europa.eu/eurostat/web/nuts/history, https://ec.europa.eu/eurostat/documents/345175/629341/NUTS2013-NUTS2016.xlsx

`search_eurostat` *Grep Datasets Titles from Eurostat*

Description

Lists names of dataset from eurostat with the particular pattern in the description.

Usage

```
search_eurostat(pattern, type = "dataset", fixed = TRUE)

grepEurostatTOC(pattern, type = "dataset")
```

Arguments

pattern	Character, datasets, folder or tables with this pattern in the description will be returned (depending on the 'type' argument)
type	Grep the Eurostat table of contents either for 'dataset' (default), 'folder', 'table' or "all" (for all types).
fixed	logical. If TRUE, pattern is a string to be matched as is. Change to FALSE if more complex regex matching is needed.

Details

Downloads list of all datasets available on eurostat and return list of names of datasets that contains particular pattern in the dataset description. E.g. all datasets related to education of teaching.

Value

A tibble with eight columns

- titleThe name of dataset of theme
- codeThe codename of dataset of theme, will be used by the get_eurostat and get_eurostat_raw functions.
- typeIs it a dataset, folder or table.
- last.update.of.data, last.table.structure.change, data.start, data.endDates.

Functions

- `grepEurostatTOC`: Old deprecated version

Author(s)

Przemyslaw Biecek and Leo Lahti <`ropengov-forum@googlegroups.com`>

References

See citation("eurostat")

See Also

`get_eurostat`, `get_eurostat_toc`

Examples

```
## Not run:
  tmp <- search_eurostat("education")
  head(tmp)
  # Use "fixed = TRUE" when pattern has characters that would need escaping.
  # Here, parentheses would normally need to be escaped in regex
  tmp <- search_eurostat("Live births (total) by NUTS 3 region", fixed = TRUE)

## End(Not run)
```

`tgs00026` *Auxiliary Data*

Description

Auxiliary Data Sets

Usage

```
tgs00026
```

Format

data_frame

Details

Retrieved with: `tgs00026 <-get_eurostat("tgs00026",time_format = "raw")`

Index