Eurostat R tools

2016-08-16

This R package provides tools to access Eurostat database as part of the rOpenGov project.

For contact information and source code, see the github page

Installation

Release version:

```
install.packages("eurostat")
```

Development version:

```
library(devtools)
install_github("ropengov/eurostat")
```

Overall, the eurostat package includes the following functions:

```
clean_eurostat_cache    Clean Eurostat Cache
dic_order               Order of Variable Levels from Eurostat
                        Dictionary.
eu_countries            Countries and Country Codes
eurostat-package        R Tools for Eurostat open data
eurotime2date           Date Conversion from Eurostat Time Format
eurotime2num            Conversion of Eurostat Time Format to Numeric
get_eurostat            Read Eurostat Data
get_eurostat_dic        Download Eurostat Dictionary
get_eurostat_json       Get Data from Eurostat API in JSON
get_eurostat_raw        Download Data from Eurostat Database
get_eurostat_toc        Download Table of Contents of Eurostat Data
                        Sets
harmonize_country_code
                        Harmonize Country Code
label_eurostat          Get Eurostat Codes
search_eurostat         Grep Datasets Titles from Eurostat
```

Finding data

Function `get_eurostat_toc()` downloads a table of contents of eurostat datasets. The values in column 'code' should be used to download a selected dataset.

```
# Load the package
library(eurostat)
library(rvest)

# Get Eurostat data listing
```

```
toc <- get_eurostat_toc()

# Check the first items
library(knitr)
kable(head(toc))
```

title	code	type	last.update.of.data	last.table.structure
Database by themes	data	folder		
General and regional statistics	general	folder		
European and national indicators for short-term analysis	euroind	folder		
Business and consumer surveys (source: DG ECFIN)	ei_bcs	folder		
Consumer surveys (source: DG ECFIN)	ei_bcs_cs	folder		
Consumers - monthly data	ei_bsco_m	dataset	28.07.2016	28.07.2016

With `search_eurostat()` you can search the table of contents for particular patterns, e.g. all datasets related to *passenger transport*. The kable function to produces nice markdown output. Note that with the `type` argument of this function you could restrict the search to for instance datasets or tables.

```
# info about passengers
kable(head(search_eurostat("passenger transport")))
```

	title
5688	Volume of passenger transport relative to GDP
5689	Modal split of passenger transport
5742	Railway transport - Total annual passenger transport (1 000 pass., million pkm)
5746	International railway passenger transport from the reporting country to the country of disembarkation (1 000 passen
5747	International railway passenger transport from the country of embarkation to the reporting country (1 000 passenge
6097	Air passenger transport by reporting country

Codes for the dataset can be searched also from the Eurostat database. The Eurostat database gives codes in the Data Navigation Tree after every dataset in parenthesis.

Downloading data

The package supports two of the Eurostats download methods: the bulk download facility and the Web Services' JSON API. 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 a time and whole datasets usually exceeds that. To download only a small section of the dataset the JSON API is faster, as it allows to make a data selection before downloading.

A user does not usually have to bother with methods, as both are used via main function `get_eurostat()`. 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.

Here an example of indicator Modal split of passenger transport. This is the percentage share of each mode of transport in total inland transport, expressed in passenger-kilometres (pkm) based on transport by passenger cars, buses and coaches, and trains. All data should be based on movements on national territory, regardless of the nationality of the vehicle. However, the data collection is not harmonized at the EU level.

Pick and print the id of the data set to download:

```
id <- search_eurostat("Modal split of passenger transport",
                      type = "table")$code[1]
print(id)
```

[1] "tsdtr210"

Get the whole corresponding table. As the table is annual data, it is more convient to use a numeric time variable than use the default date format:

```
dat <- get_eurostat(id, time_format = "num")
```

Investigate the structure of the downloaded data set:

```
str(dat)
```

```
## 'data.frame':    2326 obs. of  5 variables:
##  $ unit   : Factor w/ 1 level "PC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ vehicle: Factor w/ 3 levels "BUS_TOT","CAR",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ geo    : Factor w/ 35 levels "AT","BE","CH",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ time   : num  1990 1990 1990 1990 1990 1990 1990 1990 1990 1990 ...
##  $ values : num  11 10.6 3.7 9.1 11.3 32.4 14.9 13.5 6 24.8 ...
```

```
kable(head(dat))
```

	unit	vehicle	geo	time	values
1	PC	BUS_TOT	AT	1990	11.0
2	PC	BUS_TOT	BE	1990	10.6
4	PC	BUS_TOT	CH	1990	3.7
7	PC	BUS_TOT	DE	1990	9.1
8	PC	BUS_TOT	DK	1990	11.3
10	PC	BUS_TOT	EL	1990	32.4

Or you can get only a part of the dataset by defining `filters` argument. It should be named list, where names corresponds to variable names (lower case) and values are vectors of codes corresponding desidered series (upper case). For time variable, in addition to a `time`, also a `sinceTimePeriod` and a `lastTimePeriod` can be used.

```
dat2 <- get_eurostat(id, filters = list(geo = c("EU28", "FI"), lastTimePeriod=1), time_format = "num")
kable(dat2)
```

unit	vehicle	geo	time	values
PC	BUS_TOT	EU28	2014	9.1
PC	BUS_TOT	FI	2014	9.8
PC	CAR	EU28	2014	83.4
PC	CAR	FI	2014	85.2
PC	TRN	EU28	2014	7.6
PC	TRN	FI	2014	5.0

Replacing codes with labels

By default variables are returned as Eurostat codes, but to get human-readable labels instead, use a `type = "label"` argument.

```
dat12 <- get_eurostat(id, filters = list(geo = c("EU28", "FI"),
                                         lastTimePeriod = 1),
                      type = "label", time_format = "num")
kable(head(dat12))
```

unit	vehicle	geo	time	values
Percentage	Motor coaches, buses and trolley buses	European Union (28 countries)	2014	9.1
Percentage	Motor coaches, buses and trolley buses	Finland	2014	9.8
Percentage	Passenger cars	European Union (28 countries)	2014	83.4
Percentage	Passenger cars	Finland	2014	85.2
Percentage	Trains	European Union (28 countries)	2014	7.6
Percentage	Trains	Finland	2014	5.0

Eurostat codes can be replaced also after downloadind with human-readable labels using a function `label_eurostat()`. It replaces the eurostat codes based on definitions from Eurostat dictionaries.

```
datl <- label_eurostat(dat)
kable(head(datl))
```

	unit	vehicle	geo	time	valu
1	Percentage	Motor coaches, buses and trolley buses	Austria	1990	1
2	Percentage	Motor coaches, buses and trolley buses	Belgium	1990	1
4	Percentage	Motor coaches, buses and trolley buses	Switzerland	1990	
7	Percentage	Motor coaches, buses and trolley buses	Germany (until 1990 former territory of the FRG)	1990	
8	Percentage	Motor coaches, buses and trolley buses	Denmark	1990	1
10	Percentage	Motor coaches, buses and trolley buses	Greece	1990	3

The `label_eurostat()` allows also conversion of individual variable vectors or variable names.

```
label_eurostat_vars(names(datl))
```

Vehicle information has 3 levels. You can check them now with:

```
levels(datl$vehicle)
```

Selecting and modifying data

EFTA, Eurozone, EU and EU candidate countries

To facilititate fast plotting of standard European geographic areas, the package provides ready-made lists of the country codes used in the eurostat database for EFTA (efta_countries), Euro area (ea_countries), EU (eu_countries) and EU candidate countries (candidate_countries). This helps to select specific groups

of countries for closer investigation. For conversions with other standard country coding systems, see the countrycode R package. To retrieve the country code list for EFTA, for instance, use:

```
data(efta_countries)
kable(efta_countries)
```

code	name
IS	Iceland
LI	Liechtenstein
NO	Norway
CH	Switzerland

EU data from 2012 in all vehicles:

```
dat_eu12 <- subset(datl, geo == "European Union (28 countries)" & time == 2012)
kable(dat_eu12, row.names = FALSE)
```

unit	vehicle	geo	time	values
Percentage	Motor coaches, buses and trolley buses	European Union (28 countries)	2012	9.3
Percentage	Passenger cars	European Union (28 countries)	2012	83.0
Percentage	Trains	European Union (28 countries)	2012	7.7

EU data from 2000 - 2012 with vehicle types as variables:

Reshaping the data is best done with `spread()` in `tidyr`.

```
library("tidyr")
dat_eu_0012 <- subset(dat, geo == "EU28" & time %in% 2000:2012)
dat_eu_0012_wide <- spread(dat_eu_0012, vehicle, values)
kable(subset(dat_eu_0012_wide, select = -geo), row.names = FALSE)
```

unit	time	BUS_TOT	CAR	TRN
PC	2000	10.4	82.4	7.2
PC	2001	10.2	82.7	7.1
PC	2002	9.9	83.3	6.8
PC	2003	9.9	83.5	6.7
PC	2004	9.8	83.4	6.8
PC	2005	9.9	83.2	6.9
PC	2006	9.7	83.2	7.1
PC	2007	9.8	83.1	7.2
PC	2008	9.7	83.1	7.3
PC	2009	9.2	83.7	7.1
PC	2010	9.2	83.6	7.2
PC	2011	9.2	83.4	7.3
PC	2012	9.3	83.0	7.7

Train passengers for selected EU countries in 2000 - 2012

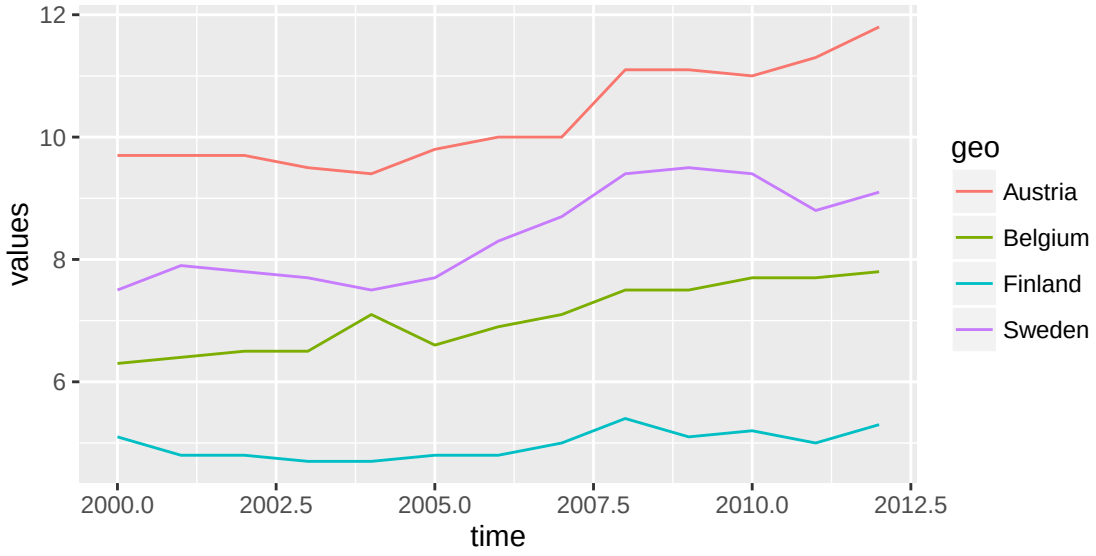
```
dat_trains <- subset(datl, geo %in% c("Austria", "Belgium", "Finland", "Sweden")
                     & time %in% 2000:2012
                     & vehicle == "Trains")

dat_trains_wide <- spread(dat_trains, geo, values)
kable(subset(dat_trains_wide, select = -vehicle), row.names = FALSE)
```

unit	time	Austria	Belgium	Finland	Sweden
Percentage	2000	9.7	6.3	5.1	7.5
Percentage	2001	9.7	6.4	4.8	7.9
Percentage	2002	9.7	6.5	4.8	7.8
Percentage	2003	9.5	6.5	4.7	7.7
Percentage	2004	9.4	7.1	4.7	7.5
Percentage	2005	9.8	6.6	4.8	7.7
Percentage	2006	10.0	6.9	4.8	8.3
Percentage	2007	10.0	7.1	5.0	8.7
Percentage	2008	11.1	7.5	5.4	9.4
Percentage	2009	11.1	7.5	5.1	9.5
Percentage	2010	11.0	7.7	5.2	9.4
Percentage	2011	11.3	7.7	5.0	8.8
Percentage	2012	11.8	7.8	5.3	9.1

Visualization

Visualizing train passenger data with `ggplot2`:

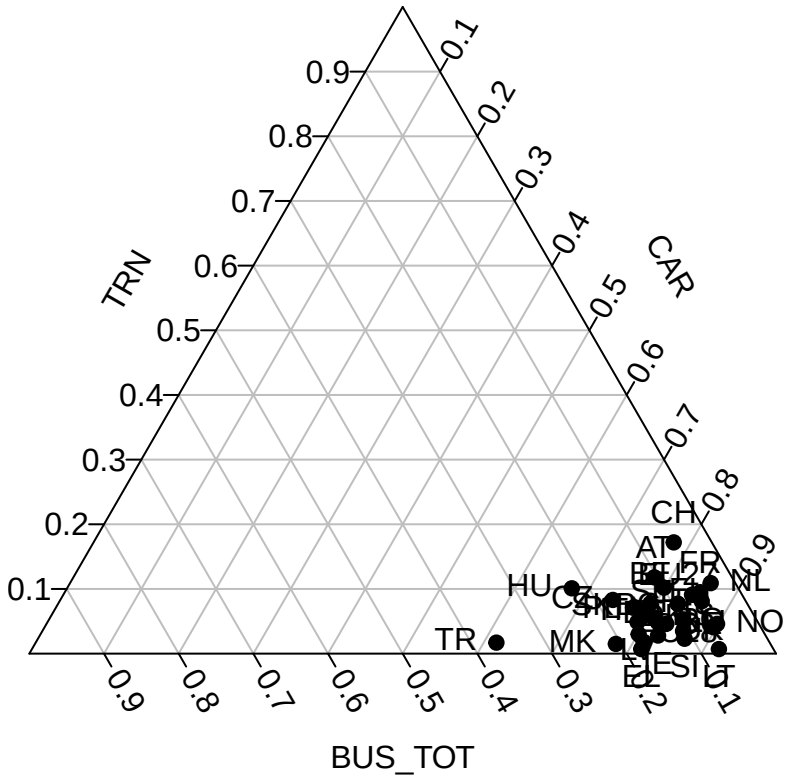
```
library(ggplot2)
p <- ggplot(dat_trains, aes(x = time, y = values, colour = geo))
p <- p + geom_line()
print(p)
```



Triangle plot

Triangle plot on passenger transport distributions with 2012 data for all countries with data.

```
library(tidyr)

transports <- spread(subset(dat, time == 2012, select = c(geo, vehicle, values)), vehicle, values)

transports <- na.omit(transports)

# triangle plot
library(plotrix)
triax.plot(transports[, -1], show.grid = TRUE,
           label.points = TRUE, point.labels = transports$geo,
           pch = 19)
```



Citing the package

Citing the Data Kindly cite Eurostat.

Citing the R tools This work can be freely used, modified and distributed under the BSD-2-clause (modified FreeBSD) license:

```
citation("eurostat")
```

```
##
## Kindly cite the eurostat R package as follows:
##
```

```
##   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek
##   2014-2016. eurostat R package URL:
##   https://github.com/rOpenGov/eurostat
##
## A BibTeX entry for LaTeX users is
##
##   @Misc{,
##     title = {eurostat R package},
##     author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
##     year = {2014-2016},
##     url = {https://github.com/rOpenGov/eurostat},
##   }
```

Acknowledgements

We are grateful to all contributors and Eurostat open data portal! This rOpenGov R package is based on earlier CRAN packages statfi and smarterpoland. The datamart and reurostat packages seem to develop related Eurostat tools but at the time of writing this tutorial this package seems to be in an experimental stage. The quandl package may also provides access to some versions of eurostat data sets.

Session info

This tutorial was created with

```
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] plotrix_3.6-2      ggplot2_2.1.0      tidyr_0.5.1
## [4] rvest_0.3.2        xml2_1.0.0         eurostat_1.2.23
## [7] rmarkdown_0.9.6.14 knitr_1.13
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.5      magrittr_1.5     munsell_0.4.3    colorspace_1.2-6
##  [5] R6_2.1.2         plyr_1.8.4       stringr_1.0.0    httr_1.2.1
##  [9] highr_0.6        tools_3.3.1      grid_3.3.1       gtable_0.2.0
```

```
## [13] htmltools_0.3.5  yaml_2.1.13      digest_0.6.9     assertthat_0.1
## [17] tibble_1.1       formatR_1.4      curl_0.9.7       evaluate_0.9
## [21] labeling_0.3     stringi_1.1.1    scales_0.4.0     jsonlite_1.0
```