

eurostat tutorial for R

Leo Lahti, Markus Kainu, Janne Huovari, Przemyslaw Biecek

2016-01-18

Eurostat R tools

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](#)

Available tools

- [Installation](#)
- Finding data
- Downloading data
- Replacing codes with labels
- Selecting and modifying data
- [Visualization](#)
- Triangle plot
- Citing the package
- [Acknowledgements](#)
- Session info

Installation

Release version:

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

Development version:

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

Overall, the eurostat package includes the following functions:

```
library(eurostat)  
kable(as.data.frame(ls("package:eurostat")))
```

ls(“package:eurostat”)

```
candidate_countries
clean_eurostat_cache
dic_order
ea_countries
efta_countries
eu_countries
eurotime2date
eurotime2num
get_eurostat
get_eurostat_dic
getEurostatDictionary
get_eurostat_toc
getEurostatTOC
grepEurostatTOC
label_eurostat
label_eurostat_tables
label_eurostat_vars
search_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	07.01.2016	07.01.2016

With `search_eurostat()` you can search the table of contents for particular patterns, e.g. all datasets related to *passenger transport*. The `kable` function 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
5562	Volume of passenger transport relative to GDP
5563	Modal split of passenger transport
5602	Railway transport - Total annual passenger transport (1 000 pass., million pkm)
5606	International railway passenger transport from the reporting country to the country of disembarkation (1 000 passenger transport)
5607	International railway passenger transport from the country of embarkation to the reporting country (1 000 passenger transport)
5956	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

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 corresponding table. As the table is annual data, it is more convenient 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': 2520 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","BG",...: 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 NA 3.7 NA NA 9.1 11.3 NA 32.4 ...
```

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

unit	vehicle	geo	time	values
PC	BUS_TOT	AT	1990	11.0
PC	BUS_TOT	BE	1990	10.6
PC	BUS_TOT	BG	1990	NA
PC	BUS_TOT	CH	1990	3.7
PC	BUS_TOT	CY	1990	NA
PC	BUS_TOT	CZ	1990	NA

Replacing codes with labels

Eurostat variable IDs can be replaced with human-readable labels. Function `label_eurostat()` replaces the eurostat IDs based on definitions from Eurostat dictionaries.

```
dat1 <- label_eurostat(dat)
kable(head(dat1))
```

unit	vehicle	geo	time	values
Percentage	Motor coaches, buses and trolley buses	Austria	1990	11.0
Percentage	Motor coaches, buses and trolley buses	Belgium	1990	10.6
Percentage	Motor coaches, buses and trolley buses	Bulgaria	1990	NA
Percentage	Motor coaches, buses and trolley buses	Switzerland	1990	3.7
Percentage	Motor coaches, buses and trolley buses	Cyprus	1990	NA
Percentage	Motor coaches, buses and trolley buses	Czech Republic	1990	NA

Vehicle information has 3 levels. They are:

```
levels(dat1$vehicle)
```

```
## [1] "Motor coaches, buses and trolley buses"
## [2] "Passenger cars"
## [3] "Trains"
```

You can get also labels for variable names

```
label_eurostat_vars(names(dat1))
```

```
## Warning in label_eurostat(x, dic = "dimlst", lang = lang): All labels for
## dimlst were not found.
```

```
## [1] "Unit of measure"
## [2] "Vehicles"
## [3] "Geopolitical entity (reporting)"
## [4] "Period of time (a=annual, q=quarterly, m=monthly, d=daily, c=cumulated from January)"
## [5] NA
```

Selecting and modifying data

EFTA, Eurozone, EU and EU candidate countries

To facilitate 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)
print(efta_countries)
```

```
##   code      name
## 1  IS      Iceland
## 2  LI Liechtenstein
## 3  NO      Norway
## 4  CH      Switzerland
```

EU data from 2012 in all vehicles:

```
dat_eu12 <- subset(dat1, 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	7.6

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	7.6

Train passengers for selected EU countries in 2000 - 2012

```
dat_trains <- subset(dat1, geo %in% c("Austria", "Belgium", "Finland", "Sweden")
                     & time %in% 2000:2012
                     & vehicle == "Trains")

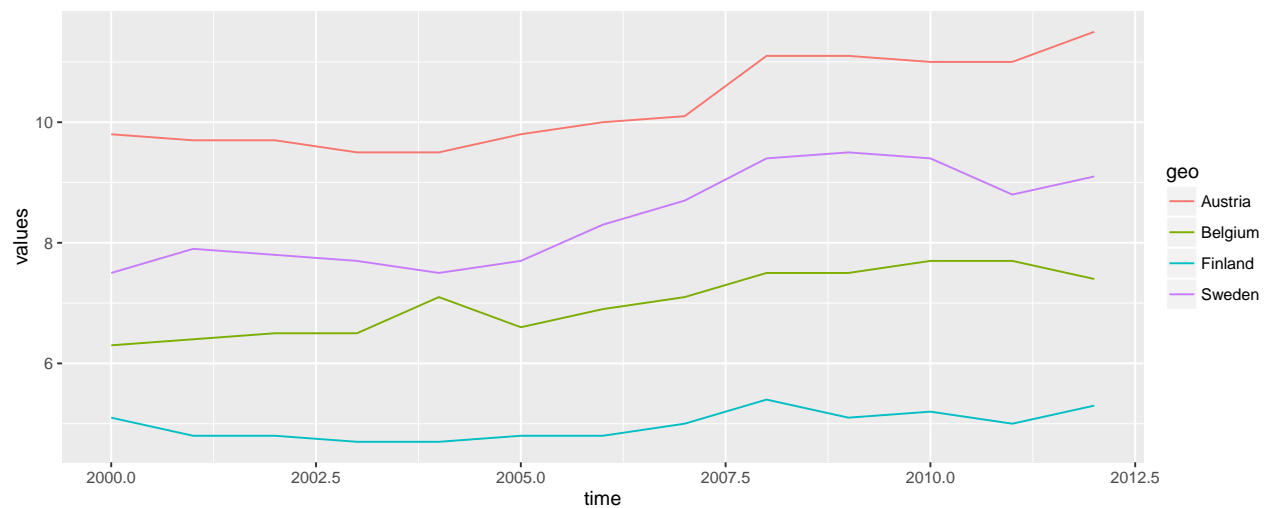
dat_trains_wide <- spread(dat_trains, geo, values)
kable(subset(dat_trains_wide, select = -vehicle), row.names = FALSE)
```

unit	time	Austria	Belgium	Finland	Sweden
Percentage	2000	9.8	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	9.5	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	10.1	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	11.0	7.7	5.0	8.8
Percentage	2012	11.5	7.4	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)
```



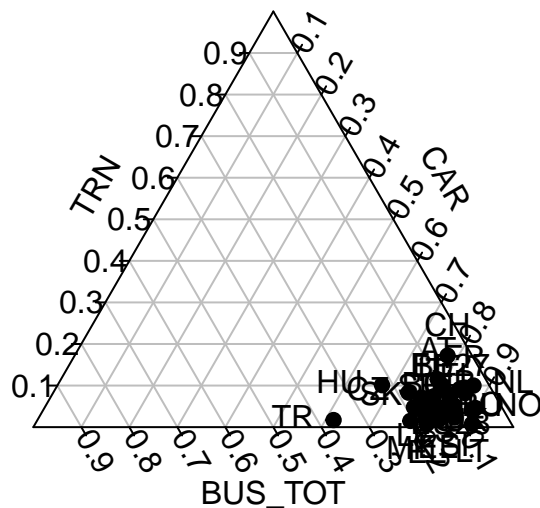
Triangle plot on passenger transport distributions with 2012 data for all countries with data.

```
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)
```



For further examples, see also the [blog post on the eurostat R package](#).

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},
##   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 3.2.2 (2015-08-14)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 15.10
##
## 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_3.6-1  ggplot2_2.0.0  tidyr_0.3.1   rvest_0.3.1
## [5] xml2_0.1.2     eurostat_1.2.11 rmarkdown_0.9.4 knitr_1.12
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.3      magrittr_1.5      munsell_0.4.2    colorspace_1.2-6
##  [5] R6_2.1.1         plyr_1.8.3        stringr_1.0.0    httr_1.0.0
##  [9] highr_0.5.1      dplyr_0.4.3       tcltk_3.2.2      tools_3.2.2
## [13] parallel_3.2.2   grid_3.2.2        gtable_0.1.2     DBI_0.3.1
## [17] htmltools_0.3    yaml_2.1.13       digest_0.6.9     assertthat_0.1
## [21] formatR_1.2.1    evaluate_0.8       labeling_0.3      stringi_1.0-1
## [25] scales_0.3.0
```