Retrieval and analysis of Eurostat open data with the eurostat package

Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek

Abstract The increasing availability of open statistical data resources is providing novel opportunities for research and citizen science. Efficient algorithmic tools are needed to realize the full potential of the new information resources. We introduce the **eurostat** R package that provides a collection of custom tools for Eurostat open data, including functions to query, download, manipulate, and visualize these data sets in a smooth, automated and reproducible manner. The online documentation provides detailed examples on the retrieval and analysis of these spatio-temporal data sets. The **eurostat** R package provides remarkable improvements over the previously available tools, and has been extensively tested by an active user community. This contributes to the growing ecosystem of R packages dedicated to reproducible research in computational social science and digital humanities.

Introduction

Eurostat, the statistical office of the European Union, provides a rich collection of data through its open data service[1], which includes thousands of data sets on European demography, economics, health, infrastructure, traffic and other topics. The statistics are often available with great geographical resolution and include time series spanning over several years or decades.

Availability of algorithmic tools to access and analyse open data collections can greatly benefit reproducible research (Gandrud, 2013; Boettiger et al., 2015), as complete analytical workflows spanning from raw data to final publications can be made fully replicable and transparent. Dedicated software packages help to simplify, standardize, and automate analysis workflows, greatly facilitating reproducibility, code sharing, and efficient data analytics. The algorithms need to be customized to specific data sources to accommodate variations in raw data formats, access details, and typical use cases so that the end users can avoid repetitive programming tasks and save time. A number of packages for governmental and other sources have been designed to meet these demands, including packages for the Food and Agricultural Organization (FAO) of the United Nations (**FAOSTAT**; Kao et al. (2015)), World Bank (**WDI**; Arel-Bundock (2013)), national statistics authorities (**pxweb**; Magnusson et al. (2014)), Open Street Map (**osmar**; Eugster and Schlesinger (2012)) and many other sources.

A dedicated R package for the Eurostat open data has been missing. The **eurostat** R package fills this gap. It combines the relevant parts from our earlier **statfi** (Lahti et al., 2013a) and **smarterpoland** (Biecek, 2015) packages and implements an expanded set of custom tools. Since its first CRAN release in 2014, the **eurostat** package has been actively developed by several contributors via Github based on frequent feedback from the user community. We are now reporting the first mature version that has been improved and tested by multiple users, and applied in several case studies by us and others[2]. The Eurostat has three services for programmatic data access: a bulk download, json/unicode, and SDMX web service; we provide methods for the first two. The bulk download provides single files, which is convenient and fast when major parts of data need to be retrieved. More light-weight json methods that allow data subsetting before the download may be preferred in more specific retrieval tasks; a disadvantage of this alternative json method is that the query size is limited to 50 categories.

The **datamart** (Weinert, 2014), **quandl** (McTaggart et al., 2015) and **pdfetch** (Reinhart, 2015) packages can be used to access certain versions of Eurostat data. Compared to these generic database packages, **eurostat** is particularly tailored for the Eurostat open data service. It depends on further R packages including **classInt** (Bivand, 2015), **dplyr** (Wickham and Francois, 2015a), **httr** (Wickham, 2016), **jsonlite** (Ooms, 2014), **knitr** (Xie, 2015), **ggplot2** (Wickham, 2009), **mapproj** (for R by Ray Brownrigg et al., 2015), **RColorBrewer** (Neuwirth, 2014), **readr** (Wickham and Francois, 2015b), **sp** (Pebesma and Bivand, 2005), **stringi** (Gagolewski and Tartanus, 2015), and **stringr** (Wickham, 2015b). The **eurostat** package is part of the rOpenGov collection (Lahti et al., 2013b) that provides reproducible research tools for computational social science and digital humanities.

In summary, **eurostat** package provides custom algorithms for Eurostat open data. It supports key features such as cache, date formatting, and tidy data (Wickham, 2014). The data sets are provided as **tibble** data frames (Wickham et al., 2016) to support standard tools for data subsetting and reshaping. Here, we provide an overview of the core functionality in the current CRAN release version (2.1.1). Further documentation and reproducible source code of this article are available via Github[3].

[1] http://ec.europa.eu/eurostat/data/database

[2] See e.g. http://blog.revolutionanalytics.com/2015/04/financial-times-tracks-unemployment-with-r.html

[3] https://github.com/rOpenGov/eurostat

Search and download commands

To install and load the CRAN release version, just type in R:

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

The database table of contents is available on-line[4], or can be downloaded in R with `get_eurostat_toc()`. A more focused search is provided by the `search_eurostat()` function:

```
> query <- search_eurostat("road accidents", type = "table")
```

This seeks data on road accidents. The `type` argument limits the search on a selected data set type, one of three hierarchical levels including *'table'*, which resides in *'dataset'*, which is in turn stored in a *'folder'*. Values in the `code` column of the `search_eurostat()` function output provide data identifiers for subsequent download commands. Alternatively, these identifiers can be browsed at the Eurostat open data service; check the codes in the Data Navigation Tree listed after each dataset in parentheses. Let us look at the data identifier and title for the first entry of the query data:

```
> query$code[[1]]
[1] "tsdtr420"

> query$title[[1]]
[1] "People killed in road accidents"
```

Let us next retrieve the data set with this identifier:

```
> dat <- get_eurostat(id = "tsdtr420", time_format = "num")
```

We have here used a numeric time format, which is more convient for annual time series than the default date format. The above function call returns a table of transport statistics (Table 1). This can be filtered before the download with the `filters` argument, where the list names and values are Eurostat variable and observation codes, respectively. To retrieve filtered transport statistics for specific countries, use:

```
> t1 <- get_eurostat("tsdtr420",
+   filters = list(geo = c("UK", "SK", "FR", "PL", "ES", "PT")))
```

	sex	geo	time	values
1	T	AT	1999.00	1079.00
2	T	BE	1999.00	1397.00
3	T	BG	1999.00	
4	T	CH	1999.00	
5	T	CY	1999.00	
6	T	CZ	1999.00	1455.00

Table 1: First lines of output from the `get_eurostat()` function with the road accident data set identifier 'tsdtr420'.

	sex	geo	time	values
1	Total	Austria	1999.00	1079.00
2	Total	Belgium	1999.00	1397.00
3	Total	Bulgaria	1999.00	
4	Total	Switzerland	1999.00	
5	Total	Cyprus	1999.00	
6	Total	Czech Republic	1999.00	1455.00

Table 2: The output from `get_eurostat()` (Table 1), now converted into human-readable labels with `label_eurostat(dat)`.

[4] http://ec.europa.eu/eurostat/data/database

A subsequent visualization reveals a decreasing trend of road accidents over time (Figure 1):

```
> ggplot(t1, aes(x = time, y = values, color=geo, group=geo, shape=geo)) +
+   geom_point(size=4) + geom_line() + theme_bw() +
+   ggtitle("Road accidents")+ xlab("Year") + ylab("Victims (n)") +
+   theme(legend.position="none") +
+   ggrepel::geom_label_repel(data=t1 %>% group_by(geo) %>% na.omit() %>%
+     filter(time %in% c(min(time), max(time))), aes(fill=geo,label=geo),color="white")
```

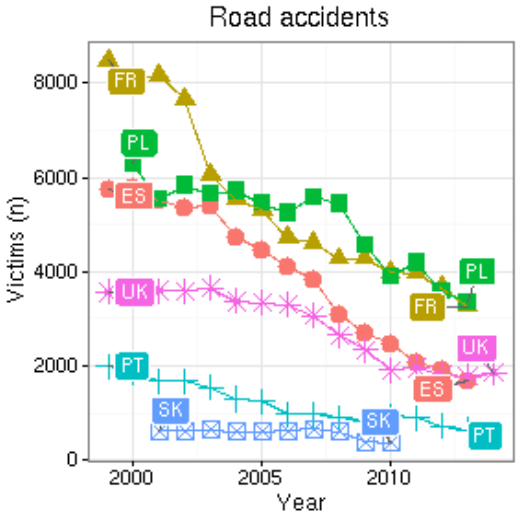

Figure 1: Timeline indicating the number of people killed in road accidents in various countries based on Eurostat open data retrieved with the **eurostat** package.

Utilities

Many entries in Table 1 are not readily interpretable, but a simple call `label_eurostat(dat)` can be used to convert the original identifier codes into human-readable labels (Table 2) based on translations in the Eurostat database. Labels are available in English, France and Germany.

The Eurostat database includes a variety of demographic and health indicators. We see, for instance, that overweight varies remarkably across different age groups (Figure 2A). Sometimes the data sets require more complicated pre-processing. Let's consider, for instance, the distribution of renewable energy sources in different European countries. In order to summarise such data one needs to first aggregate a multitude of possible energy sources into a smaller number of coherent groups. Then one can use standard R tools to process the data, chop country names, filter countries depending on production levels, normalize the within country production. After a series of transformations (see Appendix for the source code) we can finally plot the data to discover that countries vary a lot in terms of renewable energy sources (Figure 2B). Three-dimensional data sets such as this can be conveniently visualized as triangular maps by using the **plotrix** (Lemon, 2006) package.

The data sets are stored in cache by default to avoid repeated downloads of identical data and to speed up the analysis. Storing an exact copy of the retrieved raw data on the hard disk will also support reproducibility when the source database is constantly updated.

Geospatial information

Map visualizations

The indicators in the Eurostat open data service are typically available as annual time series grouped by country, and sometimes at more refined temporal or geographic levels. Eurostat provides complementary geospatial data on the corresponding administrative statistical units to support visualizations at the appropriate geographic resolution. The geospatial data sets are available as standard shapefiles[5].

[5]http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units

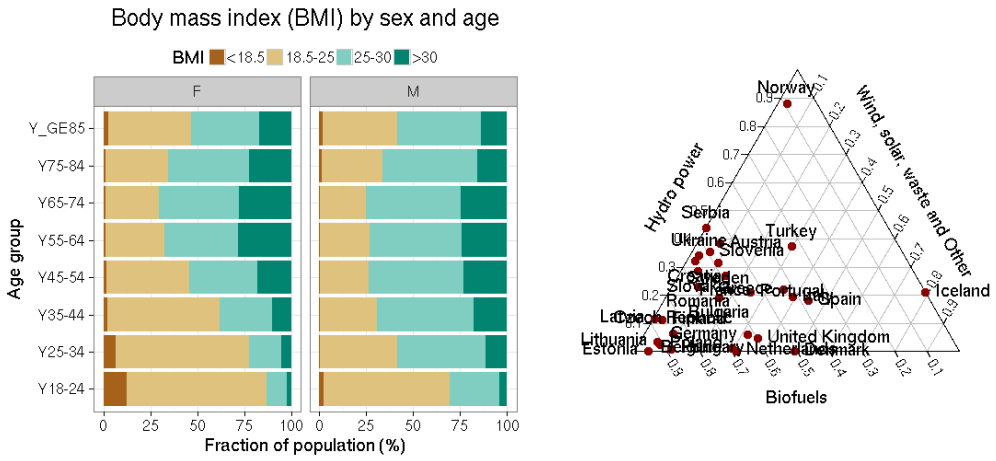

Figure 2: **A** The body-mass index (BMI) in different age groups in Poland (Eurostat table `hlth_ehis_de1`). **B** Production of renewable energy in various countries in 2013 (Eurostat table `ten00081`). See the Appendix for the source code.

Let us look at disposable income of private households (data identifier tgs00026[6]). This is provided at the geographic NUTS2 regions, the intermediate territorial units in the Eurostat regional classifications, roughly corresponding to provinces or states in each country[7] (Figure 3). The map is generated with:

```
> # Loading required libraries
> library(eurostat)
> library(dplyr)
> library(ggplot2)

> # Downloading and manipulating the tabular data
> get_eurostat("tgs00026", time_format = "raw") %>%
+   # subsetting to year 2005 and NUTS-3 level
+   dplyr::filter(time == 2005, nchar(as.character(geo)) == 4) %>%

+   # Classify the values the variable
+   dplyr::mutate(cat = cut_to_classes(values)) %>%

+   # Merge Eurostat data with geodata from Cisco
+   merge_eurostat_geodata(data=.,geocolumn="geo",resolution = "60", output_class ="df") %>%

+   # Plot map
+   ggplot(data=., aes(long,lat,group=group)) +
+   geom_polygon(aes(fill = cat),colour=alpha("white", 1/2),size=.2) +
+   scale_fill_manual(values=RColorBrewer::brewer.pal(n = 5, name = "Oranges")) +
+   labs(title="Disposable household income") +
+   coord_map(project="orthographic", xlim=c(-22,34), ylim=c(35,70)) + theme_minimal() +
+   guides(fill = guide_legend(title = "EUR per Year",title.position = "top", title.hjust=0))
```

This demonstrates how the Eurostat statistics and geospatial data, retrieved with the eurostat package, can be combined with other R utilities including **grid** (R Core Team, 2015), **maptools** (Bivand and Lewin-Koh, 2015), **rgdal** (Bivand et al., 2015), **rgeos** (Bivand and Rundel, 2015), **scales** (Wickham, 2015a), and **stringr** (Wickham, 2015b).

Default country groupings

To facilitate the analysis and visualization of standard European country groups, we have included ready-made country code lists. The list of EFTA countries (Table 3), for instance, is retrieved with:

[6] http://ec.europa.eu/eurostat/en/web/products-datasets/-/TGS00026
[7] http://ec.europa.eu/eurostat/web/nuts/overview

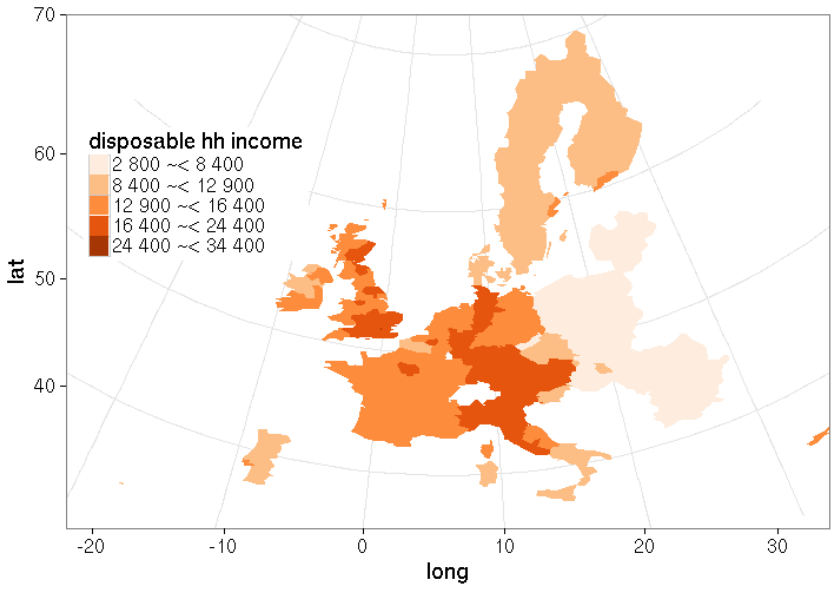

Figure 3: Disposable income of private households across NUTS2-level national regions in European countries visualized based on geospatial data available from Eurostat.

```
> data(efta_countries)
```

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

Table 3: The EFTA country listing from the eurostat R package.

Similar lists are available for Euro area (ea_countries), EU (eu_countries) and the EU candidate countries (eu_candidate_countries). These auxiliary data sets facilitate smooth selection of specific country groups for a closer analysis. The full name and a two-letter identifier are provided for each country according to the Eurostat database. The country codes follow the ISO 3166-1 alpha-2 standard, except that GB and GR are replaced by UK (United Kingdom) and EL (Greece) in the Eurostat database, respectively. Linking these country codes with external data sets can be facilitated by conversions between different country coding standards with the **countrycode** package (Arel-Bundock, 2014).

Summary

By combining programmatic access to data with custom analysis and visualization tools it is possible to facilitate a seamless automation of the complete data analytical workflow from raw data to statistical summaries and final publication. The **eurostat** R package provides convenient tools to access open data from Eurostat. It exemplifies automated and transparent data retrieval from institutional data repositories, featuring options such as search, subsetting and cache. Moreover, it provides several functions to facilitate the analysis and visualization of the Eurostat data. Possible future extensions and improvements include implementation of specific data representation formats that could be used to harmonize the data representation with other related data sources and to facilitate subsequent tool development. In particular, we should take further advantage of the existing spatiotemporal data structures that are available in R, such as those provided by the **spacetime** package (Pebesma, 2012), and construct wrapper functions to speed up routine operations such as the visualization of the temporal and geospatial data sets. The source code and installation instructions for the latest

development version of the eurostat package and this manuscript[8] are available via Github. The source code can be freely used, modified and distributed under the BSD-2-clause (modified FreeBSD) license. Issues, bug reports, pull requests, and other feedback are welcome.

Acknowledgements

We are grateful to Eurostat for maintaining the open data service and the rOpenGov[9] for supporting R package development. This work has been partially funded by Academy of Finland (decision 293316). We also wish to thank all package contributors.

Bibliography

V. Arel-Bundock. *WDI: World Development Indicators (World Bank)*, 2013. URL http://CRAN.R-project.org/package=WDI. R package version 2.4. [p1]

V. Arel-Bundock. *countrycode: Convert Country Names and Country Codes*, 2014. URL http://CRAN.R-project.org/package=countrycode. R package version 0.18. [p5]

P. Biecek. *SmarterPoland: Tools for Accessing Various Datasets Developed by the Foundation SmarterPoland.pl*, 2015. URL http://CRAN.R-project.org/package=SmarterPoland. R package version 1.5. [p1]

R. Bivand. *classInt: Choose Univariate Class Intervals*, 2015. URL https://CRAN.R-project.org/package=classInt. R package version 0.1-23. [p1]

R. Bivand and N. Lewin-Koh. *maptools: Tools for Reading and Handling Spatial Objects*, 2015. URL http://CRAN.R-project.org/package=maptools. R package version 0.8-37. [p4]

R. Bivand and C. Rundel. *rgeos: Interface to Geometry Engine - Open Source (GEOS)*, 2015. URL http://CRAN.R-project.org/package=rgeos. R package version 0.3-14. [p4]

R. Bivand, T. Keitt, and B. Rowlingson. *rgdal: Bindings for the Geospatial Data Abstraction Library*, 2015. URL http://CRAN.R-project.org/package=rgdal. R package version 1.0-7. [p4]

C. Boettiger, S. Chamberlain, E. Hart, and K. Ram. Building software, building community: Lessons from the ropensci project. *Journal of Open Research Software*, 3(1), November 2015. [p1]

M. J. A. Eugster and T. Schlesinger. Openstreetmap and r. *R Journal*, 5(1):53–63, June 2012. [p1]

D. M. P. for R by Ray Brownrigg, T. P. Minka, and transition to Plan 9 codebase by Roger Bivand. *mapproj: Map Projections*, 2015. URL http://CRAN.R-project.org/package=mapproj. R package version 1.2-4. [p1]

M. Gagolewski and B. Tartanus. *R package stringi: Character string processing facilities*, 2015. URL http://stringi.rexamine.com/. [p1]

C. Gandrud. *Reproducible Research with R and R Studio*. Chapman & Hall/CRC, July 2013. [p1]

M. C. J. Kao, M. Gesmann, and F. Gheri. *FAOSTAT: Download Data from the FAOSTAT Database of the Food and Agricultural Organization (FAO) of the United Nations*, 2015. URL http://CRAN.R-project.org/package=FAOSTAT. R package version 2.0. [p1]

L. Lahti, J. Parkkinen, and J. Lehtomäki. statfi r package, 2013a. [p1]

L. Lahti, J. Parkkinen, J. Lehtomäki, and M. Kainu. ropengov: open source ecosystem for computational social sciences and digital humanities. Presentation at ICML/MLOSS workshop (Int'l Conf. on Machine Learning - Open Source Software workshop)., December 2013b. URL http://ropengov.github.io. [p1]

J. Lemon. Plotrix: a package in the red light district of r. *R-News*, 6(4):8–12, 2006. [p3]

M. Magnusson, L. Lahti, and L. Hansson. pxweb: R tools for px-web api, 2014. URL http://CRAN.R-project.org/package=pxweb. R package version 0.5.57. [p1]

R. McTaggart, G. Daroczi, and C. Leung. *Quandl: API Wrapper for Quandl.com*, 2015. URL http://CRAN.R-project.org/package=Quandl. R package version 2.7.0. [p1]

[8] https://github.com/rOpenGov/eurostat
[9] https://github.ropengov.io

E. Neuwirth. *RColorBrewer: ColorBrewer Palettes*, 2014. URL https://CRAN.R-project.org/package=RColorBrewer. R package version 1.1-2. [p1]

J. Ooms. The jsonlite package: A practical and consistent mapping between json data and r objects. *arXiv:1403.2805 [stat.CO]*, 2014. URL http://arxiv.org/abs/1403.2805. [p1]

E. Pebesma. spacetime: Spatio-temporal data in r. *Journal of Statistical Software*, 51(7):1–30, 2012. URL http://www.jstatsoft.org/v51/i07/. [p5]

E. Pebesma and R. Bivand. Classes and methods for spatial data in r. *R News*, 5(2), 2005. [p1]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL https://www.R-project.org/. [p4]

A. Reinhart. *pdfetch: Fetch Economic and Financial Time Series Data from Public Sources*, 2015. URL http://CRAN.R-project.org/package=pdfetch. R package version 0.1.7. [p1]

K. Weinert. *datamart: Unified access to your data sources*, 2014. URL http://CRAN.R-project.org/package=datamart. R package version 0.5.2. [p1]

H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009. ISBN 978-0-387-98140-6. URL http://had.co.nz/ggplot2/book. [p1]

H. Wickham. Tidy data. *Journal of Statistical Software*, 59(10), 2014. [p1]

H. Wickham. *scales: Scale Functions for Visualization*, 2015a. URL http://CRAN.R-project.org/package=scales. R package version 0.3.0. [p4]

H. Wickham. *stringr: Simple, Consistent Wrappers for Common String Operations*, 2015b. URL http://CRAN.R-project.org/package=stringr. R package version 1.0.0. [p1, 4]

H. Wickham. *httr: Tools for Working with URLs and HTTP*, 2016. URL https://CRAN.R-project.org/package=httr. R package version 1.2.1. [p1]

H. Wickham and R. Francois. *dplyr: A Grammar of Data Manipulation*, 2015a. URL http://CRAN.R-project.org/package=dplyr. R package version 0.4.3. [p1]

H. Wickham and R. Francois. *readr: Read Tabular Data*, 2015b. URL https://CRAN.R-project.org/package=readr. R package version 0.2.2. [p1]

H. Wickham, R. Francois, and K. Müller. *tibble: Simple Data Frames*, 2016. URL https://CRAN.R-project.org/package=tibble. R package version 1.1. [p1]

Y. Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, 2nd edition edition, 2015. [p1]

Leo Lahti
Department of Mathematics and Statistics
PO Box 20014 University of Turku
Finland
leo.lahti@iki.fi

Janne Huovari
Pellervo Economic Research PTT
Eerikinkatu 28 A 00180 Helsinki
Finland
janne.huovari@ptt.fi

Markus Kainu
Research Department, The Social Insurance Institution of Finland
PO Box 450, 00101 Helsinki
Finland
markus.kainu@kela.fi

Przemyslaw Biecek
Faculty of Mathematics, Informatics, and Mechanics
University of Warsaw
Banacha 2, 02-097 Warsaw
Poland
P.Biecek@mimuw.edu.pl

Appendix

Source code for the obesity example (Figure 2A):

```
> library(dplyr)
> tmp1 <- get_eurostat("hlth_ehis_de1", time_format = "raw")
> tmp1 %>%
+  dplyr::filter( isced97 == "TOTAL" ,
+          sex != "T",
+          age != "TOTAL", geo == "PL") %>%
+  mutate(BMI = factor(bmi,
+                      levels=c("LT18P5","18P5-25","25-30","GE30"),
+                      labels=c("<18.5", "18.5-25", "25-30",">30"))) %>%
+  arrange(BMI) %>%
+  ggplot(aes(y=values, x=age, fill=BMI)) + geom_bar(stat="identity") +
+  facet_wrap(~sex) + coord_flip() +
+  theme(legend.position="top") +
+  ggtitle("Body mass index (BMI) by sex and age") +
+  xlab("% of population") + scale_fill_brewer(type = "div")
```

Source code for the renewable energy example (Figure 2B):

```
# All sources of renewable energy are to be grouped into three sets
> dict <- c("Solid biofuels (excluding charcoal)" = "Biofuels",
+          "Biogasoline" = "Biofuels",
+          "Other liquid biofuels" = "Biofuels",
+          "Biodiesels" = "Biofuels",
+          "Biogas" = "Biofuels",
+          "Hydro power" = "Hydro power",
+          "Tide, Wave and Ocean" = "Hydro power",
+          "Solar thermal" = "Wind, solar, waste and Other",
+          "Geothermal Energy" = "Wind, solar, waste and Other",
+          "Solar photovoltaic" = "Wind, solar, waste and Other",
+          "Municipal waste (renewable)" = "Wind, solar, waste and Other",
+          "Wind power" = "Wind, solar, waste and Other",
+          "Bio jet kerosene" = "Wind, solar, waste and Other")
# Some cleaning of the data is required
> energy3 <- get_eurostat("ten00081") %>%
+  label_eurostat(dat) %>%
+  filter(time == "2013-01-01",
+         product != "Renewable energies") %>%
+  mutate(nproduct = dict[as.character(product)], # just three categories
+         geo = gsub(geo, pattern=" \\(.*", replacement="")) %>%
+  select(nproduct, geo, values) %>%
+  group_by(nproduct, geo) %>%
+  summarise(svalue = sum(values)) %>%
+  group_by(geo) %>%
+  mutate(tvalue = sum(svalue),
+         svalue = svalue/sum(svalue)) %>%
+  filter(tvalue > 1000,
+         !grepl(geo, pattern="^Euro")) %>% # only large countrie
+  spread(nproduct, svalue)
# Triangle plot
> library(plotrix)
> par(cex=0.75)
> plotrix::triax.plot(as.matrix(energy3[, c(3,5,4)]),
+                     show.grid = TRUE,
+                     label.points = TRUE, point.labels = energy3$geo,
+                     pch = 19)
```