Access Eurostat data with eurostat::**cheat sheet**

Search and download

Data in the Eurostat database is stored in tables.
Each table has an identifier, a short table_code, and
a description (e.g. tps00165 - Death due to transport accidents, by sex).

Key eurostat functions allow to find the table_code,
download the eurostat table and polish labels in the table.

Find the table code

The **search_eurostat(pattern, ...)** function scans the directory of Eurostat tables and returns codes and descriptions of tables that match pattern.

```
library("eurostat")
query <- search_eurostat(pattern = "fertility rate",
                         type = "table", fixed = FALSE)
query[,1:2]
## title                                code
## <chr>                                <chr>
## Total fertility rate by NUTS 2 region tgs00100
## Total fertility rate                 tps00199
## Total fertility rate by NUTS 2 region tgs00100
```

Download the table

The **get_eurostat(id, time_format = "date", filters = "none", type = "code", cache = TRUE, ...)** function downloads the requested table from the Eurostat bulk download facility or from The Eurostat Web Services JSON API (if filters are defined). Downloaded data is cached (if cache=TRUE). Additional arguments define how to read the time column (time_format) and if table dimensions shall be kept as codes or converted to labels (type).

```
ct <- c("AT","BE","BG","CH","CY","CZ","DE","DK","EE","EL","ES",
        "FI","FR","HR","HU","IE","IS","IT","LI","LT","LU","LV",
        "MT","NL","NO","PL","PT","RO","SE","SI","SK","UK")
dat <- get_eurostat(id="tps00199", time_format="num",
                    filters = list(geo = ct))
dat[1:2,]
## indic_de geo    time values
## TOTFERRT AT     2006   1.41
## TOTFERRT AT     2007   1.38
```

Add labels

The **label_eurostat(x, lang = "en", ...)** gets definitions for Eurostat codes and replace them with labels in given language ("en", "fr" or "de")

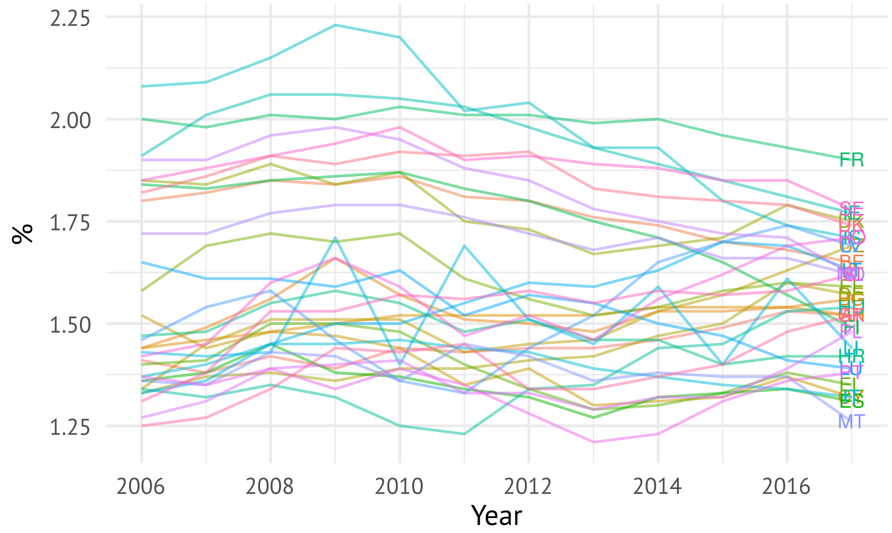
```
dat <- label_eurostat(dat)
dat[1:3,]
## indic_de             geo      time  values
## <fct>                <fct>    <dbl>  <dbl>
## Total fertility rate Andorra  2006    1.24
## Total fertility rate Albania  2006    1.67
## Total fertility rate Armenia  2006    1.34
```

eurostat and plots

The **get_eurostat()** function returns tibbles in the long format. Packages dplyr and tidyr are well suited to transform these objects. The **ggplot2** -package is well suited to plot these objects.

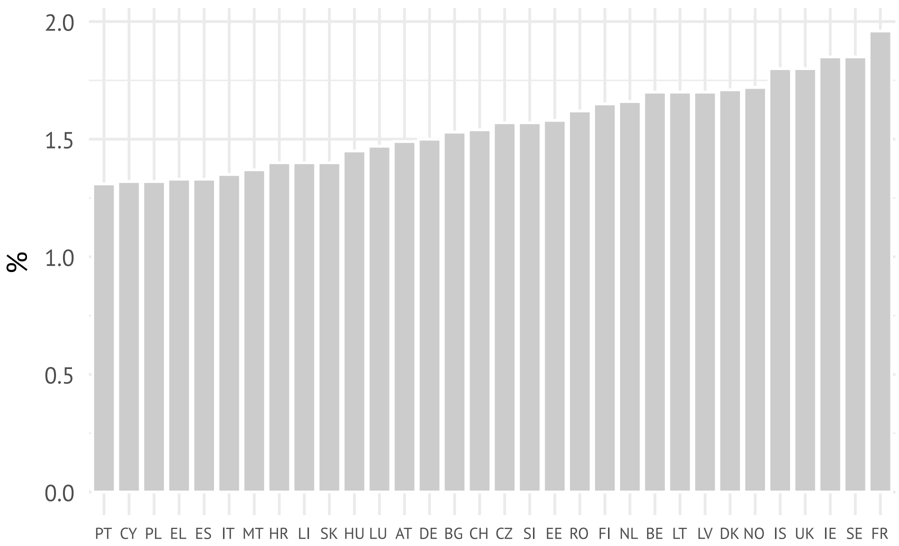
```
ggplot(dat,
       aes(x = time, y = values, color = geo, label = geo)) +
  geom_line(alpha = .5) +
  geom_text(data = dat %>% group_by(geo) %>%
              filter(time == max(time)),
            size = 2.6) +
  theme(legend.position = "none") +
  labs(title = "Total fertility rate, 2006-2017",
       x = "Year", y = "%")
```

Total fertility rate, 2006-2017



```
dat_2015 <- dat %>%
  filter(time == "2015-01-01")
ggplot(dat_2015, aes(x = reorder(geo, values), y = values)) +
  geom_col(color = "white", fill = "grey80") +
  theme(axis.text.x = element_text(size = 6)) +
  labs(title = "Total fertility rate, 2015",
       y = "%", x = NULL)
```

Total fertility rate, 2015



eurostat and maps

There are two function to work with geospatial data from GISCO. The **get_eurostat_geospatial()** returns spatial data as sf-object. Object can me merged with data.frames using **dplyr::*_join()**-functions. The **cut_to_classes()** is a wrapper for cut() - function and is used for categorizing data for maps with tidy labels.

```
mapdata <- get_eurostat_geospatial(nuts_level = 0) %>%
  left_join(dat_2015) %>%
  mutate(cat = cut_to_classes(values, n=4, decimals=1))
head(select(mapdata,geo,values,cat))
## geo values        cat                       geometry
## BG    1.53 1.5 ~< 1.7 MULTIPOLYGON (((22.99717 43...
## CH    1.54 1.5 ~< 1.7 MULTIPOLYGON (((8.61383 47....
## CY    1.32 1.3 ~< 1.5 MULTIPOLYGON (((33.75237 34...
## AL    1.59 1.5 ~< 1.7 MULTIPOLYGON (((19.831 42.4...
## CZ    1.57 1.5 ~< 1.7 MULTIPOLYGON (((14.49122 51...
## BE    1.70 1.5 ~< 1.7 MULTIPOLYGON (((5.10218 51....
```

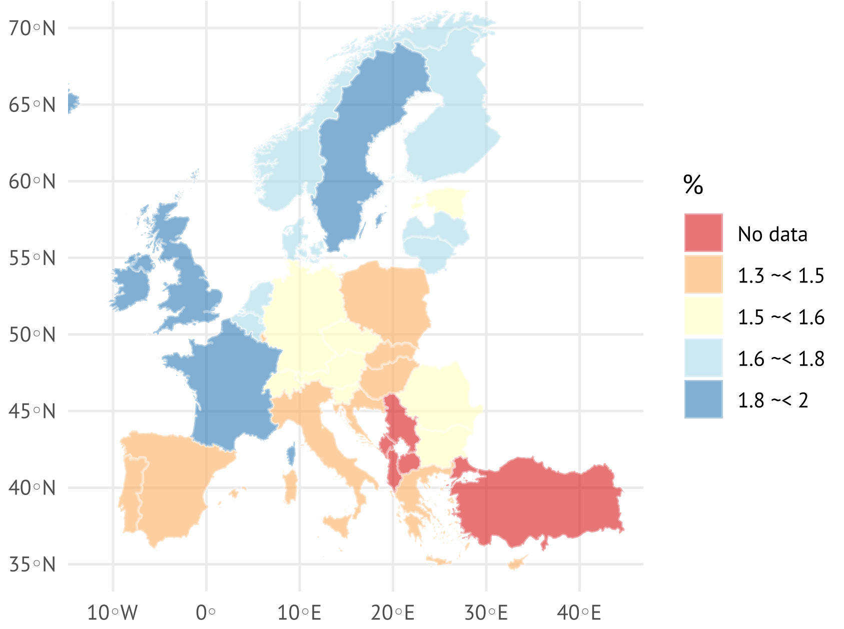
Plot a Map

The **sf**-object returned are ready to be plotted with **ggplot::geom_sf()**-function.

```
ggplot(mapdata, aes(fill = cat)) +
  scale_fill_brewer(palette = "RdYlBu") +
  geom_sf(color = alpha("white",1/3), alpha = .6) +
  xlim(c(-12,44)) + ylim(c(35,70)) +
  labs(title = "Total fertility rate, 2015",
       subtitle = "Avg. number of life births per woman",
       fill = "%")
```

Total fertility rate, 2015

Avg. number of life births per woman



This onepager presents the eurostat package 2014-2019
Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek
package version 3.3.55 URL: https://github.com/rOpenGov/eurostat

See also: Lahti et al. Retrieval and Analysis of Eurostat Open Data with the eurostat Package. The R Journal 2017.