Reproducible and dynamic access to OECD data

2015-09-21

Introduction

The `OECD` package allows the user to download data from the OECD's API in a dynamic and reproducible way.

The package can be installed with the following code:

```r
library(devtools)
install_github("expersso/OECD")
library(OECD)
```

How to use the package

Unless you know the exact code of the series you're looking for, the best way to start is by downloading a dataframe with all the available datasets and their descriptions, and then run searches on it. The search string can be a regular expression, and is case-insensitive.

```r
data <- get_datasets()
search_dataset(string = "unemployment", data = data)
```

In the following, we'll explore the `DUR_D` data set, which contains data on the duration of unemployment.

```r
dataset <- "DUR_D"
```

Before downloading the series we are interested in, it is often prudent to look at the data structure, to see what type of breakdowns the data set offers:

```r
dstruc <- get_data_structure(dataset)
str(dstruc, max.level = 1)
```

The `get_data_structure` function returns a list of dataframes with human-readable values for variable names and values. The first data frame contains the variable names and shows the dimensions of a dataset:

```r
dstruc$VAR_DESC
```

It is often easiest not to specify any filters at this point, but rather download the entire dataset and then filter it with native `R` functions. However, sometimes the dataset is very large, so filtering it before download will cut down on download time. To illustrate, let's find out the available filters for the variables `SEX` and `AGE`:

```r
dstruc$SEX
dstruc$AGE
```

Let's say we're only interested in the duration of unemployment of men aged 20 to 24 in Germany and France. We provide these filters in the form of a list to the `filter` argument of the `get_dataset` function:

```
filter_list <- list(c("DEU", "FRA"), "MW", "2024")
df <- get_dataset(dataset = dataset, filter = filter_list)
head(df)
```

Let's say we're only interested in long-term unemployment. We can then first look at the variable `DURATION` to find the different levels, then go back to our list of variable descriptions to learn what they mean:

```
unique(df$DURATION)
dstruc$DURATION
```

We could of course merge the two data structures, but I rarely find that useful in the long run.

Plotting the results

We can now subset to only those unemployed for a year or more, and finally produce a plot.

```
library(dplyr)
library(ggplot2)

df %>%
  filter(DURATION == "UN5") %>%
  mutate(obsTime = as.numeric(obsTime)) %>%
  qplot(data = ., x = obsTime, y = obsValue, color = COUNTRY, geom = "line") +
  labs(x = NULL, y = "Persons, thousands", color = NULL,
       title = "Number of long-term unemployed men, aged 20-24")
```

If we want more in-depth information about a dataset (e.g. methodology, exact definitions of variables, etc), `browse_metadata` opens up a web browser with the metadata for the requested series.

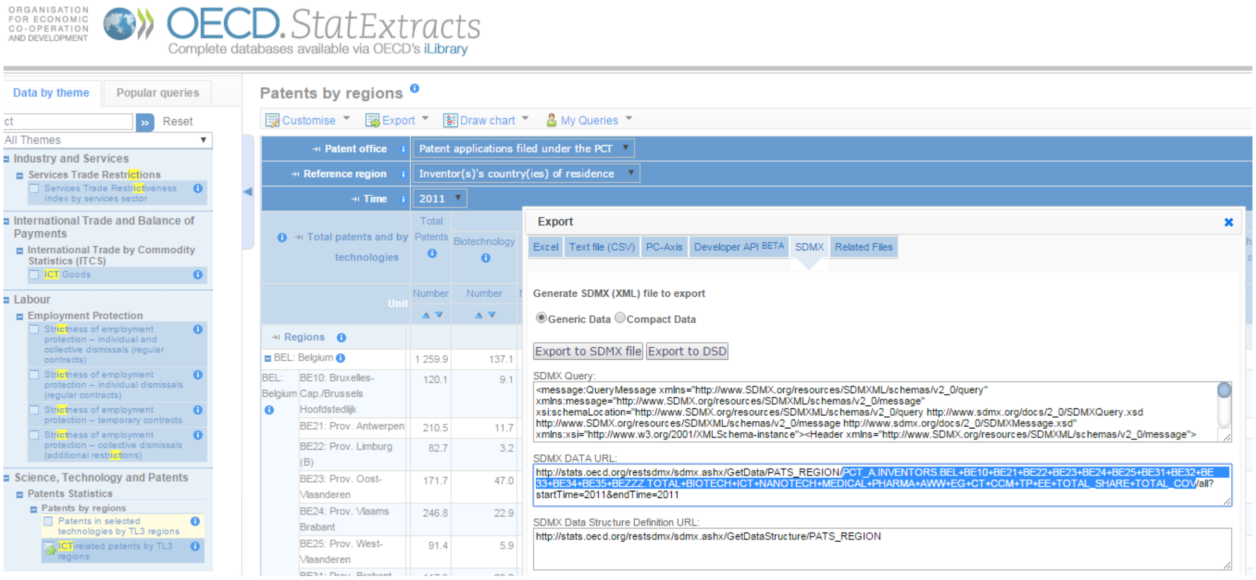
```
browse_metadata(dataset)
```

Alternative data-acquisition strategy

If one does not know exactly what data one is looking for, or if a data set contains e.g. a large number of breakdowns, it is often easier to first explore the data on the OECD stats website and then use the `oecd` package to make the data acquisition programmatic and reproducible. The workflow would then be as follows:

1. Find the data set and apply relevant filters on the OECD website.
2. Select "Export -> SDMX (XML)"
3. Copy the generated filter expression (which follows directly after the data set name, see screenshot below).
4. Insert this expression as the value to the `filter` argument of the `get_dataset` function and set the `pre_formatted` argument to `TRUE`.

```
df <- get_dataset("PATS_REGION", filter = "PCT_A.INVENTORS.BEL+BE10.TOTAL+BIOTECH",
                  pre_formatted = TRUE)
head(df)
```



Generate SDMX (XML) file to export

● Generic Data ○ Compact Data

Export to SDMX file | Export to DSD

SDMX Query:

```
<message:QueryMessage xmlns="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/query"
xmlns:message="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/message"
xsi:schemaLocation="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/query http://www.sdmx.org/docs/2_0/SDMXQuery.xsd
http://www.SDMX.org/resources/SDMXML/schemas/v2_0/message http://www.sdmx.org/docs/2_0/SDMXMessage.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Header xmlns="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/message">
```

SDMX DATA URL:

```
http://stats.oecd.org/restsdmx/sdmx.ashx/GetData/PATS_REGION/PCT_A.INVENTORS.BEL+BE10+BE21+BE22+BE23+BE24+BE25+BE31+BE32+BE
33+BE34+BE35+BEZZZ.TOTAL+BIOTECH+ICT+NANOTECH+MEDICAL+PHARMA+AWW+EG+CT+CCM+TP+EE+TOTAL_SHARE+TOTAL_COV/all?
startTime=2011&endTime=2011
```

SDMX Data Structure Definition URL:

```
http://stats.oecd.org/restsdmx/sdmx.ashx/GetDataStructure/PATS_REGION
```

Other information

The OECD API is currently a beta version and "and in preparation for the full release, the structure and content of datasets are being reviewed and are likely to evolve". As a result, the `OECD` package may break from time to time, as I update it to incorporate changes to the API. If you notice a bug (or if you have suggestions for improvements), please don't hesitate to contact me or send a pull request.

This package is in no way officially related to or endorsed by the OECD.