

Eurostat --> Matlab

/automated data downloading facility/

Jakub Ryšánek*

The purpose of this document is to introduce the reader to a Matlab based toolkit which facilitates data extraction from the Eurostat database. The toolkit is intended to serve mainly those who need to download the same data repeatedly (say, monthly). Design of the toolkit is such that it is possible to define the downloading procedure once and reuse the same procedure repeatedly in future, i.e. automatically, fast, and without much of a user interaction.

Along with the functionality to download data from a remote source, the toolkit also comes with a convenient time series management functionality (@tsobj class) and visualization capabilities (overloaded plot(.) function). A graphical user interface (GUI) enables comfortable browsing of the Eurostat database contents.

Entire suite of codes comes free of charge, is available online and is likely to be updated in the future.

1. Installation and prerequisites

Only Matlab base installation is needed in order for the codes to run properly, additional toolboxes are not required. The codes have been successfully tested on Matlab versions M2010b, M2012b, M2015b, and on Mac OS X versions M2011A and M2016A.

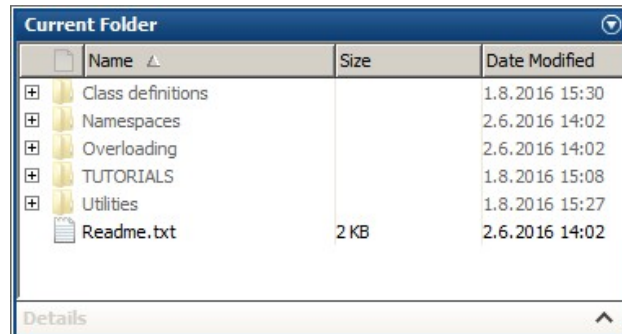
The code library itself is available at MathWorks file exchange platform, though it should suffice just to google up the phrase “eurostat matlab” in order to navigate to the toolkit:

<https://www.mathworks.com/matlabcentral/fileexchange/54564-eurostat-data-downloading-facility>

Codes need to be downloaded from the above link and unzipped into a local folder. The root folder structure is shown in figure 1. To make things accessible in Matlab, all folders need to be added to the Matlab search path (with the exception of the “TUTORIALS” folder).

* jakub.rysanek@cnb.cz, Macroeconomic Forecasting Division/Monetary Department, Czech National Bank

Figure 1. Folder structure



Name	Size	Date Modified
Class definitions		1.8.2016 15:30
Namespaces		2.6.2016 14:02
Overloading		2.6.2016 14:02
TUTORIALS		1.8.2016 15:08
Utilities		1.8.2016 15:27
Readme.txt	2 KB	2.6.2016 14:02

Since the toolkit accesses Eurostat data remotely, the general Matlab preferences require special treatment. Specifically, the user should check the following:

- If user's computer is behind a proxy server, the Matlab preferences must be set up accordingly. Navigate to "Preferences" pane, click "Web" and set properly the Proxy host and proxy port. To check whether your computer is behind a proxy, you can quickly check the Internet Explorer Options (on Windows machines), or the System/Network preferences (on Mac OS X machines).
- Default allocation of Matlab JAVA heap memory in "Preferences/General/Java Heap Memory" must be increased at least to +/-400 MB. Restart of Matlab is needed for the above changes to take effect.

The data get downloaded using web services, therefore there is no need to establish FTP connection[†].

2. Data extraction

Having installed the toolkit we can proceed to data downloading. An easy way to check whether Matlab can see the toolkit files is to initiate the database object by typing `dbEurostat()` directly into the command window. Figure 2 shows the expected result – variable `d` contains an initial structure of the downloading parameters. The structure is mostly empty, because we have not set anything yet.

The data selection process always follows the same workflow – one first needs to pick a table from which the data should be extracted and then impose a set of filtering criteria so that the potentially multidimensional input gets slashed into a reasonably scaled output.

Together with the core toolkit functions there is a "TUTORIALS" folder which contains well documented scripts. The entire data downloading workflow is also shown there.

[†] ...which is forbidden at some workplaces for security reasons.

Figure 2. DB object initialization

```
>> d = dbEUROSTAT()

d =

dbEUROSTAT with properties:

    source: 'EUROSTAT'
      url: [1x1 struct]
    table: ''
    filter: ''
    engine: 'BULK/SDMX'
deleteSourceFiles: 1
    status: 'Table not specified - picktable() might help...'

>>
```

In the next two subsections I describe how to explore Eurostat contents and how to set up automated downloading tasks.

2.1. Data definitions using GUI

As discussed above, the workflow is such that one first needs to pick a specific Eurostat table from which the data will consequently be downloaded. Having initialized the database object, `d`, we can use one of its methods – `picktable(d)` – which opens up a user interface (a GUI), as shown in figure 3. This GUI makes it easy to explore Eurostat database contents in a convenient way, without necessity to leave the Matlab session[‡]. The displayed database contents mirror exactly what the user might be able to see if he/she went online and explored the DB contents directly on the official Eurostat website.

Figure 4 then demonstrates the layout of the data selection GUI. The tree structure of the database level, which is currently being explored, is displayed in the top left part of the GUI (in this example we are in “Database by themes”/“Economy and finance” section). The main table selection panel lists all contents of the currently selected DB level from which the user can keep clicking on the nested contents.

The best way to work with the data selection GUI, however, is to use the search field (see figure 5). Here the user can type a phrase, or just a single word[§], and the results in the main

[‡] Browsing through the DB contents is performed offline so whenever the user has a suspicion that the Eurostat data structure has changed, the following command will update the local version of the table of contents: `>> TOC(dbEUROSTAT(), 'refresh', 1);` This, however, need not be performed very frequently.

[§] The search field can process simple text, regular expressions, and Google-like exact phrases (“...”).

Figure 3. Table selection GUI

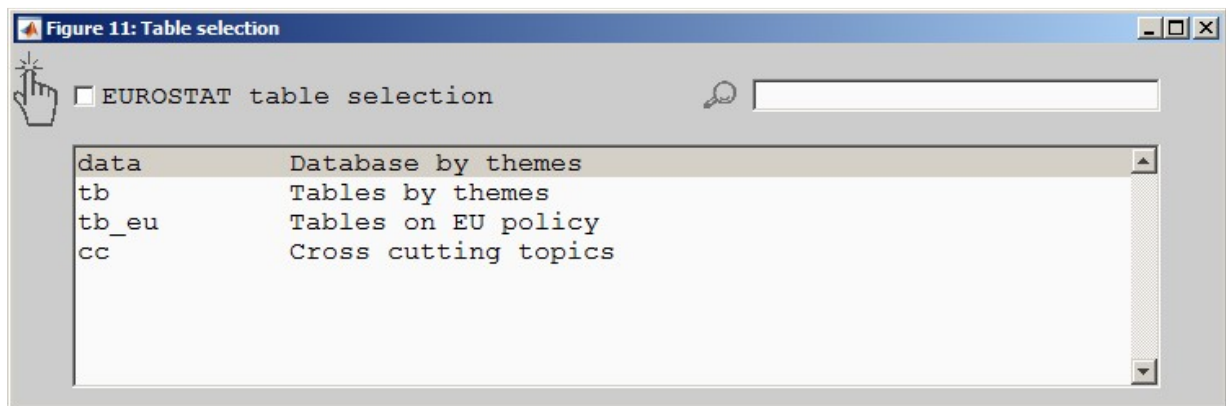


Figure 4. GUI components

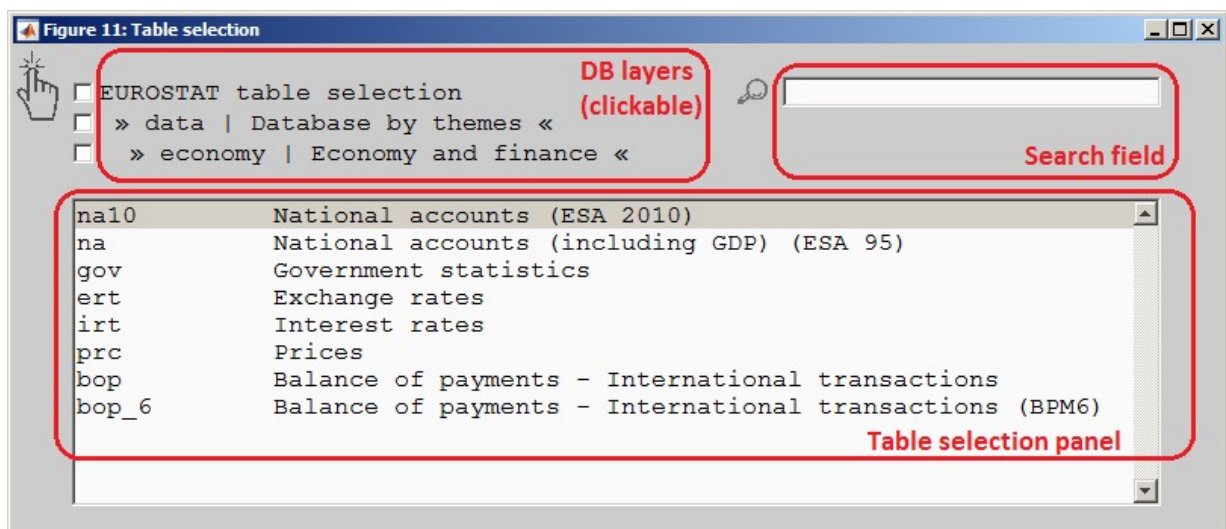
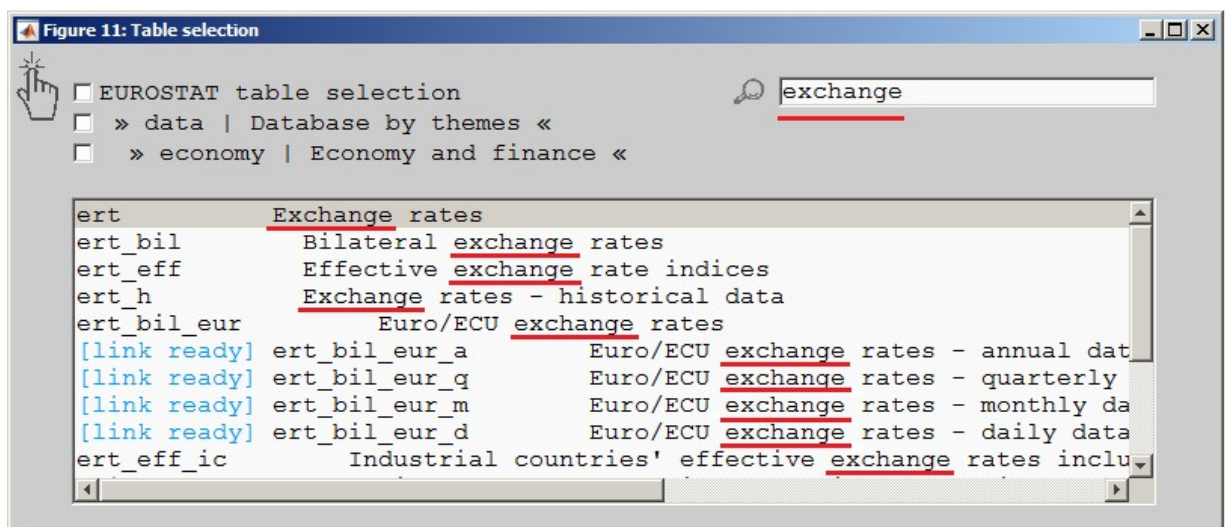


Figure 5. Table selection (use of search field)



panel then list all the DB contents which have something to do with the text entered into the search field, regardless of their position within the tree structure. The search results can either lead to final tables (marked with [\[link ready\]](#) text), or to branch points (higher tree levels containing the sought-after phrase). The search field always operates on the currently selected subtree level and all lower levels (in the example in figure 5 the “Economy and finance” section + all subsections inside), and thus potential search results from the higher (or parallel) tree levels are filtered out.

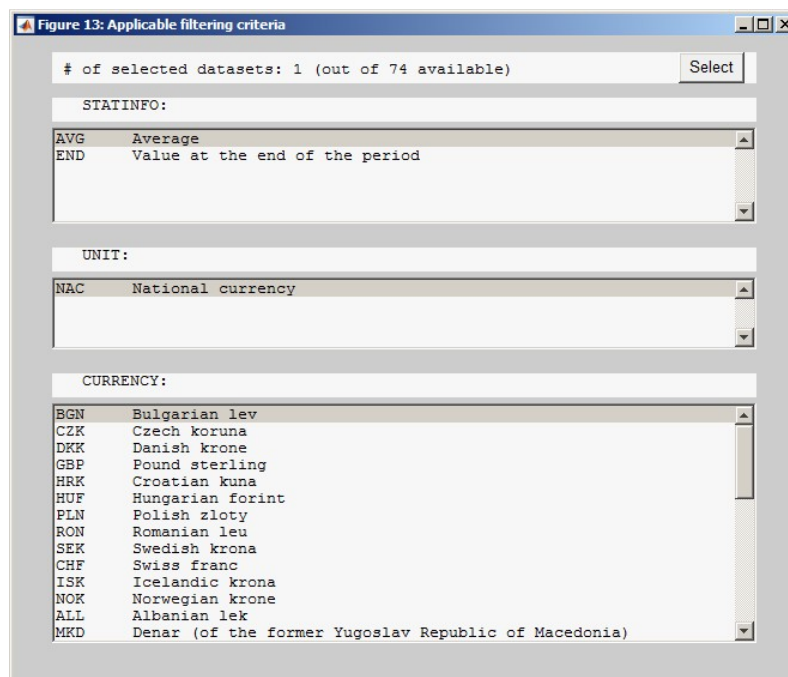
No matter how the desired table was found (by manual browsing or by using the search field), double-clicking the table generates a reference to the local (offline) version of the table of contents and the underlying database object, `d`, should then have a non-empty `table` field.

The next step is to download and process the required data. This is done by calling `tsobj(d)` command. `tsobj()` is a rich class for processing time series data. It is capable of creating time series of desired frequency from scratch directly in Matlab, it can import data from spreadsheets, or delimited text files, or (as in this case), if there is a database object on its input, it will automatically attempt to download data from a remote source.

Multidimensional tables

It is very likely that the requested Eurostat table is multidimensional. Since the toolkit assumes that all data can be fitted into a matrix (1 dimension being reserved for the time line), the user is forced to apply a set of filtering criteria. In such cases another GUI pops up, while calling the `tsobj()` command, in which the user can specify the data comfortably (figure 6 demonstrates the filtering criteria for cross exchange rates against the euro).

Figure 6. Selection of the filtering criteria



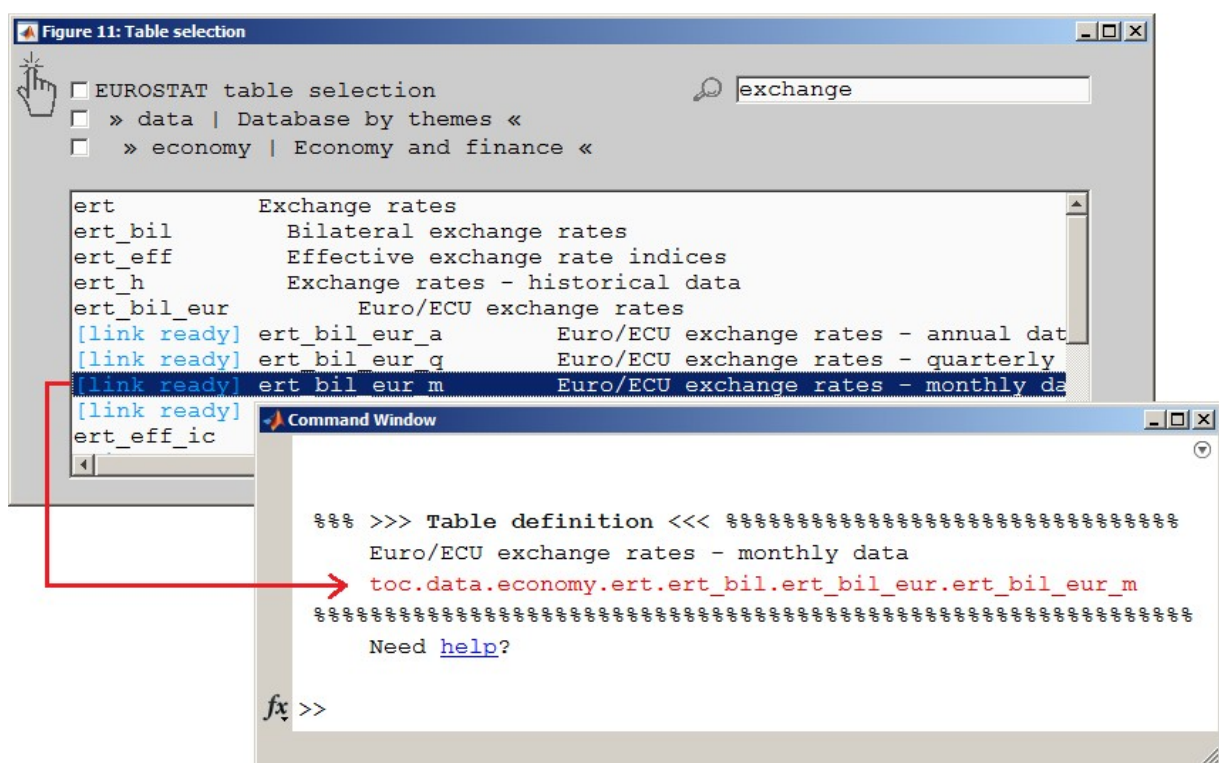
Multiple selection within a single filtering criterion can be achieved by using well-known keyboard shortcuts – CTRL+A (pick all data in a given dimension), CTRL+click (pick just a few entries), SHIFT+arrow (expand the current selection)**. Once the ‘Select’ button gets hit, a new time series object will be created.

2.2. Automated data extraction

Using the previously discussed pair of GUIs (one for the table selection and the other for data filtering) is only a convenient way of exploring the Eurostat database. Repeated downloading, however, requires saving the table definition together with the applied filtering criteria into a script so that they can be re-used in the future.

Selection of a table from GUI #1 (e.g. as in figure 5) triggers two processes. First, the underlying database object (previously marked as d) gets updated, and second, a reference to the selected table is displayed directly into the command window (see figure 7).

Figure 7. Table reference



** CTRL key can be replaced by the command symbol (⌘) on Mac OS X systems.

It then suffices to copy-paste the entire table reference (starting with `toc.`) into a script and save it for the next time. Specifically, the table reference has to be assigned in place of the empty 'table' property of the database object `d`:

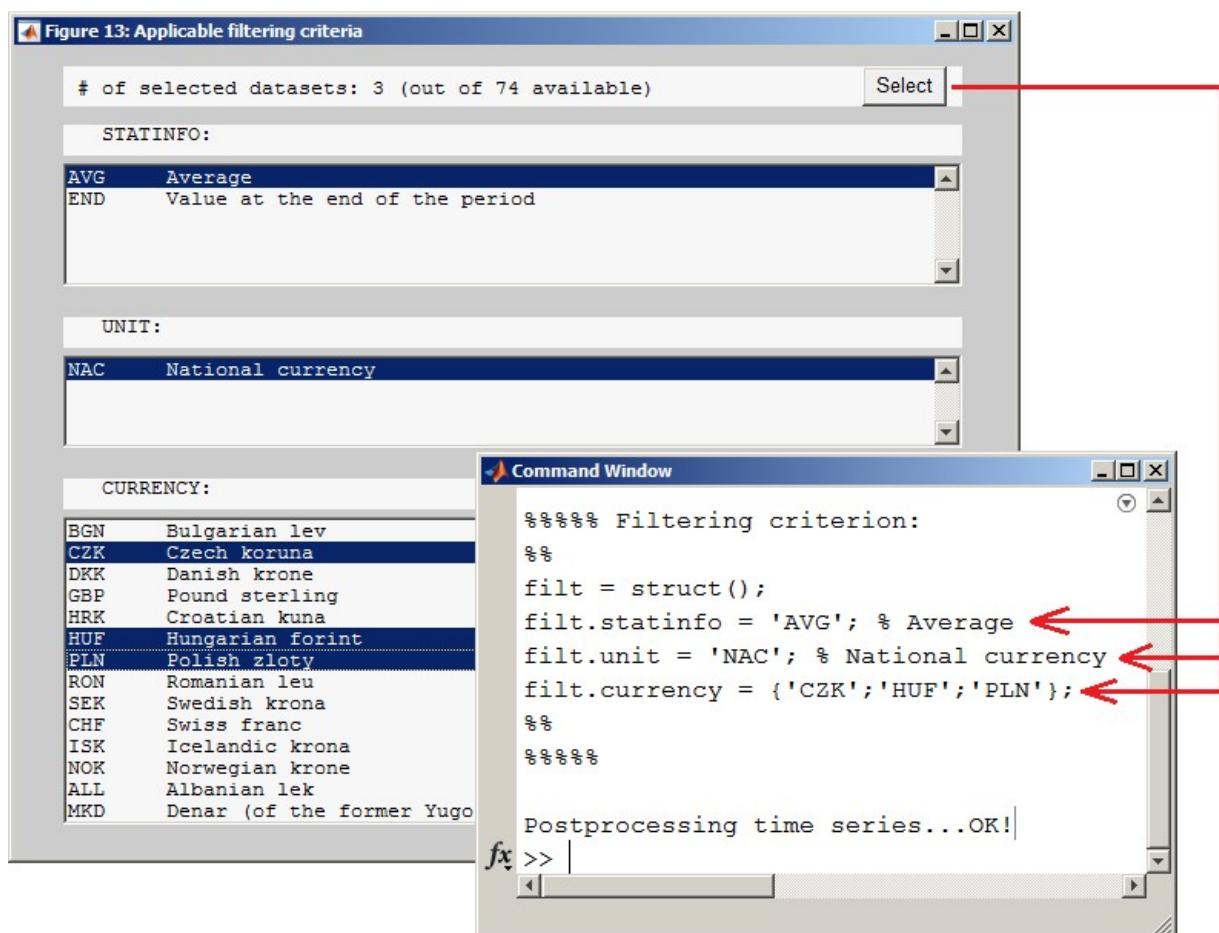
```
d.table = toc.data.economy.ert.ert_bil.ert_bil_eur.ert_bil_eur_m;
```

where the `toc.` object contains the table of contents and is accessible by applying the `TOC()` method, as in the following command:

```
>> toc = TOC(dbEurostat());
```

Application of the filtering criteria in GUI #2 (e.g. as in figure 6) works similarly. Clicking the 'Select' button updates the underlying database object and the set of selected filtering criteria gets thrown into the command window (figure 8). The user can then copy-paste the set of criteria into a script and re-use it in the future.

Figure 8. Selected filtering criteria



The example script in figure 9 summarizes a complete setup for automated downloading of monthly exchange rates of Czech koruna, Hungarian forint and Polish zloty against the euro.

Figure 9. Definition of a downloading task

The screenshot shows a MATLAB script in the Editor window and its execution in the Command Window. The script defines a database object, fetches Eurostat table contents, selects a specific table, applies filtering criteria for average values, national currency, and specific currencies (CZK, HUF, PLN), and finally downloads the selected data.

```

1 % Create a database object
2 d = dbEUROSTAT();
3
4 % Fetch EUROSTAT table of contents
5 toc = TOC(d);
6
7 % [1] Select Eurostat table
8 % -> table definition generated using pickdata(d) command
9 d.table = toc.data.economy.ert.ert_bil.ert_bil_eur.ert_bil_eur_m;
10
11 % [2] Apply filtering criteria
12 % -> the result of tsobj(d) command with empty d.filter=''
13 filt = struct();
14 filt.statinfo = 'AVG'; % Average
15 filt.unit = 'NAC'; % National currency
16 filt.currency = {'CZK'; 'HUF'; 'PLN'};
17
18 d.filter = filt;
19
20 % Download selected data
21 t = tsobj(d);
22

```

The Command Window displays the result of the download, showing a table of monthly exchange rates for CZK, HUF, and PLN from 2015M7 to 2016M7. Below the table, it indicates the object contents: Czech koruna, Hungarian forint, and Polish zloty.

2015M7	: 27.0940	311.5300	4.1524
2015M8	: 27.0410	311.6100	4.1953
2015M9	: 27.0890	313.1400	4.2176
2015M10	: 27.1050	311.2700	4.2508
2015M11	: 27.0390	312.2700	4.2494
2015M12	: 27.0270	314.4000	4.2900
2016M1	: 27.0270	314.6800	4.4074
2016M2	: 27.0400	310.3700	4.3970
2016M3	: 27.0510	311.1500	4.2932
2016M4	: 27.0310	311.4600	4.3106
2016M5	: 27.0260	314.5800	4.4039
2016M6	: 27.0610	313.9800	4.3996
2016M7	: 27.0420	314.3500	4.3964

CZK | HUF | PLN

Object contents: Czech koruna | Hungarian forint | Polish zloty

3. Downloading engines

In principle, there are three ways of obtaining the data from Eurostat:

- (1) bulk downloads,
- (2) downloads based on SDMX queries,
- (3) downloads based on JSON queries.

Bulk downloads (1) refer to direct downloading of data in tab delimited text file format (with extension .tsv). In this case one can treat the database just like a remote hard drive from which the data get copied. Except in this case the data reside on a server so instead of a drive letter one needs a full hyperlink to the requested data (<http://.../filename.tsv>).

Both SDMX (2) and JSON (3) formats go under the label of web services. In practice, the user is expected to enter all data specifications directly to the web browser's address bar, which can be automated. Eurostat provides a thorough explanation of how to construct such web queries using these formats on their official website. The output of such queries comes in different formats – SDMX queries return XML output, JSON queries are in JSON format.

I encourage the interested reader of this manual to go online and explore various web services through which Eurostat enables us, the users, to download the data:

<http://ec.europa.eu/eurostat/data/web-services>.

The downloading toolkit discussed herein has implemented two of the above downloading methods – bulk downloads (with metadata processing using SDMX standard) and downloads in JSON format. Therefore the database object initialization can take up two possible values:

```
d.engine = 'BULK/SDMX';  
  
d.engine = 'JSON';
```

The main reason for implementation of more than one downloading method is that occasionally it can happen that one or the other service is temporarily unavailable. Should such situation occur, the user can give it a try and download the requested data with different downloading engine and still get the requested data.

4. Concluding remarks

- The toolkit codes from MathWorks File Exchange are distributed without the Eurostat table of contents, therefore the first call to TOC(.) method (upon the toolkit installation) downloads the table of contents and saves it locally. As indicated earlier, the local version of the table of contents can always be updated using the 'refresh' option:

```
>> TOC(dbEUROSTAT(), 'refresh', 1);
```

- Since most of Eurostat data are in fact time series, the results of web queries are always post-processed so that the user ends up with data in the time series format. I.e. besides the data itself, the user is left with name of the time series, data frequency and the span of the data (see the time series output in figure 9).
- The toolkit is well designed even for those who do not use Matlab on daily basis. Once the data are successfully downloaded, one can make use of `export()` function which converts the result into a spreadsheet, or into a delimited text file. The user's workflow can then continue outside Matlab. The syntax of `export()` function is shown in the "TUTORIALS" folder which is distributed together with the toolkit codes.