Gauthier Vermandel ○ *gauthier.vermandel@dauphine.fr*

A Note on

DB·nomics with MATLAB

Content of the Handout

1. What is DB·nomics?

DB·nomics is an open platform that aggregates global data. Data series delivered in various formats (CSV, JSON, SDMX, HTML). Direct access from your statistical software (R, MATLAB, Python, Excel). Data series are available upon release by the provider. Keep your economic dashboard and economic indicators up-to-date. Each revision is archived to build a real-time database.

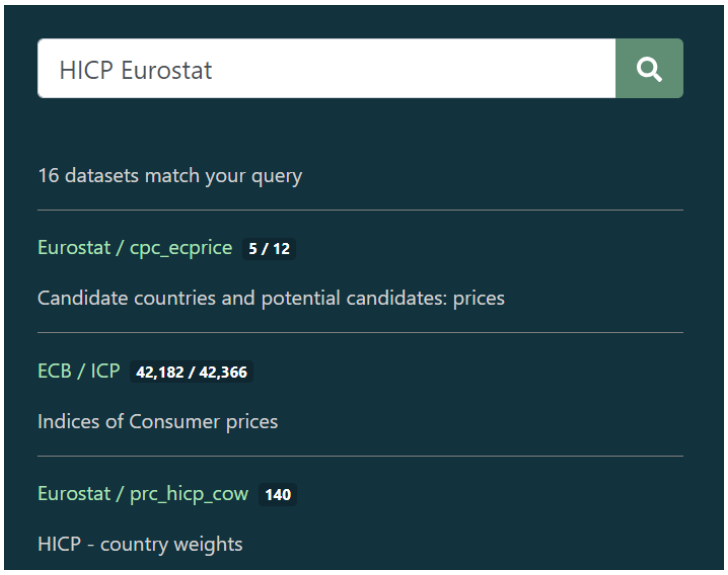
In a nutshell, DB·nomics scraps up to 37 providers (when this note was last compiled). These data are then stored on DB·nomics website, and available on a single platform. If you want check when one of the available database was last scrapped, browse db.nomics.world/views/datasets/last-update.html.

Let's start downloading macro data, first go to :

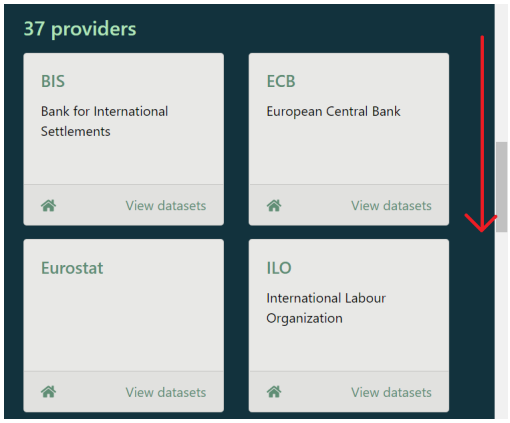
→ next.nomics.world

2. Searching my macro data

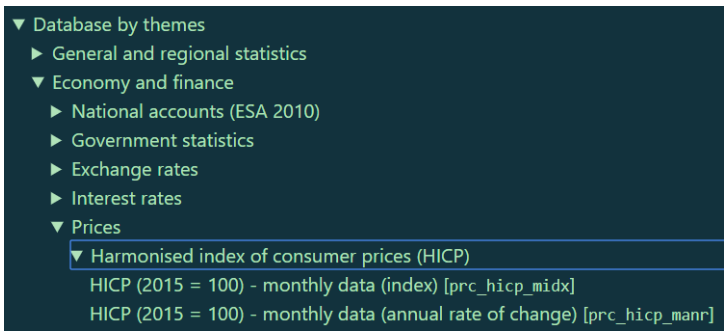
Suppose that I am interested in searching the inflation rate of Euro Area members. On DB·NOMICS, I am making the following request:



As you can see, there are 16 datasets containing HICP. Here is one of the drawback of DB·NOMICS as the search engine is not yet well performing in finding the dataset you are looking for. So I would not recommend you not to employ the search engine, instead, we can directly browse into the provider's repository tree. To do so, go back to next.nomics.world, and scroll down below the search bar until you read the list of data providers. In the latter, I am going to click on Eurostat:



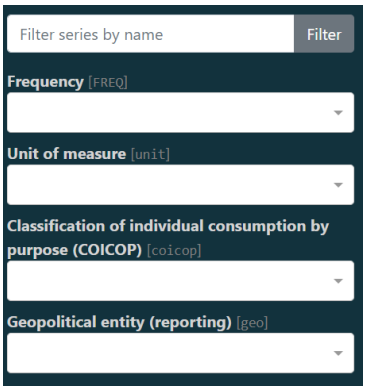
Once you are on Eurostat's page, the same repository tree as the one available on the official Eurostat web's page is also on DB·NOMICS. Eurostat's repository tree is rather intuitive, inflation series can be found right in:



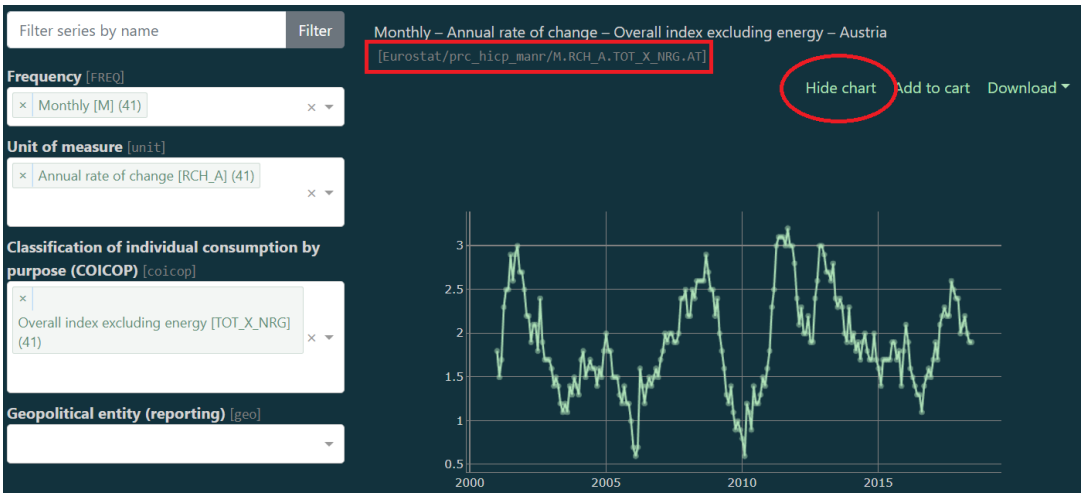
Among all inflation measures available, let us select a monthly index:

HICP (2015 = 100) - monthly data (annual rate of change)

Once selected, on the left, you have a large selection menu (that is different for each type of database) which allows you to find your series rather easily. At this stage of the data selection process, there are still 15,831 series available. This is why the selection menu on the left of the screen has a key role at this stage:



Selecting for instance "*Overall index excluding energy*" in the *Classification of individual consumption by purpose* reduces the number of series down to 41. For instance, the first serie is the inflation rate for Austria, clicking on "show chart" displays a nice plot of the serie. This is really useful when you are looking for a very specific serie.



Along the plot option, DB·NOMICS also offers the possibility to export the data into different formats, such as Excel or JSON.

The Plug-in presented here is going to reads the JSON's code from DB·NOMICS. JSON is just a data format easy to be read by any programming software, such as MATLAB. So basically, MATLAB is going to make a request to DB·NOMICS for the latest version for the serie. Each time serie has an unique token which can be seen on the previous image. For instance, for inflation for Austria, it reads as:

Eurostat/prc_hicp_manr/M.RCH_A.TOT_X_NRG.AT

We are going to use that key to get the real time data of Austrian inflation.

3. Getting my real time data on MATLAB

To get real time data from through MATLAB, only one function is required. This function is called, call_dbnomics(·) and must be in your MATLAB current folder (or in a 'set-path' folder in your MATLAB configuration). This function reads as:

```
[output_mat,output_table,dates_nb] =  call_dbnomics(varargin)
```

Here, varargin is a variable that must be one (or multiple) string(s) that contains the DB·NOMICS key. The setting of varargin is rather flexible and is detailed in the next sub-sections. This function returns three different outputs:

1. output_mat is the serie(s) 2into a matrix format. The first column is the time vector (use datestr(·) function to convert it into many different formats, see MATLAB documentation for further details).
2. output_table is the serie(s) returned into a table format. Column names are referred to as the series requested in varargin while row names are the dates into a string format.
3. dates_nb is simply a vector of dates that can be ready to be interpretable. For instance for annual series between 2000 and 2002, this vector reads as, [2000;2001;2002]. On a quarterly basis, it would reads as, [2000;2000.25;2000.5;...;2001.75;2002].

3.1 One serie

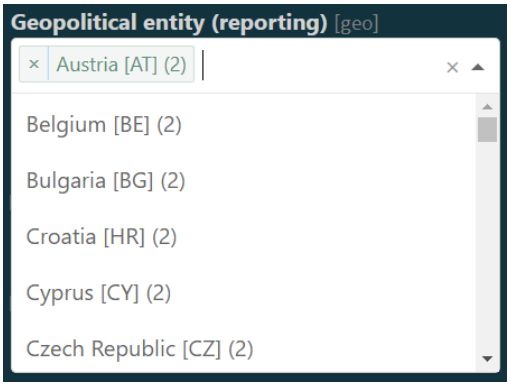
The simplest approach of using call_dbnomics(·) is to use as input only one key. For the Austrian inflation rate, it reads as:

```
[mat]=call_dbnomics('Eurostat/prc_hicp_manr/M.RCH_A.TOT_X_NRG.AT');
```

After executing this code, mat contains in the first column the vectors of dates and in the second column the series of Austrian inflation, these can be called in MATLAB through mat(:,1) and mat(:,2) respectively.

3.2 Multiple countries

It is also possible to collect different countries for one type of series. Here for inflation for instance, the list of countries which are available on dbnomics can be found in the left menu:



We can add these countries by adding '+' in the key and the two letters corresponding to the countries. For instance, the list can be enriched as follows: "AT+BE+BG+CH+CY+CZ+DE" to get data for Austria, Belgium, Bulgaria, Switzerland, Cyprus and Germany with only one command. In MATLAB, this one line code reads as:

```
[mat,tab]=call_dbnomics('Eurostat/prc_hicp_manr/M.RCH_A.TOT_X_NRG.AT+BE+BG+CH+CY+CZ+DE');
```

As for the case of one country, mat(:,1) reports the date vectors, while mat(:,2) is the inflation rate for Austria, mat(:,3) for Belgium, and so on. If starting dates are not the same between requested countries, the function will fill missing values with 'NaN'.

If you prefer tables, these are easier for managing important cross-sectionality in the data (with a lot of different countries). For instance, the tab variable header looks like:

	AT	BE	BG	CH	CY	CZ	DE
1997-01	NaN	NaN	NaN	NaN	1.5	NaN	1.4
1997-02	NaN	NaN	NaN	NaN	2.8	NaN	1.3
1997-03	NaN	NaN	NaN	NaN	2.9	NaN	1.1

3.3 Different series in the same request

As long as the series has the same frequency, it is possible to query different database. For instance if you want unemployment rate from the OECD, inflation from the ECB and output from Eurostat, it is possible to proceed such a request with one single line of code as well. For instance here, if we want to have two types of inflation rates of Eurostat, only two input arguments are required:

```
[data_matrix]=call_dbnomics('Eurostat/prc_hicp_manr/M.RCH_A.TOT_X_NRG.AT',...
'Eurostat/prc_hicp_manr/M.RCH_A.TOT_X_TBC.AT');
```

In addition, we can mix different countries with different types of series. We can add different countries to the previous request:

```
[data_matrix]=call_dbnomics('Eurostat/prc_hicp_manr/M.RCH_A.TOT_X_NRG.AT+DE',...
'Eurostat/prc_hicp_manr/M.RCH_A.TOT_X_TBC.AT+FR+DE');
```

4. Plotting

If you want to proceed to draw your own figures with MATLAB. Using the table output from the plug-in, Here, the row names are converted in number using the datenum(·) that can be used on the x-axis. To be interpretable, the datetick(·) function must be employed after the plot(·) function.

```
[~,output_table]=call_dbnomics('Eurostat/prc_hicp_manr/M.RCH_A.TOT_X_NRG.AT+BE+DE+FR');
% get columns and time vectors
TheTime  = datenum(output_table.Properties.RowNames);
Countries= output_table.Properties.VariableNames;
figure
plot(TheTime,output_table.AT,TheTime,output_table.BE,...
```

```
        TheTime,output_table.DE,TheTime,output_table.FR)
datetick('x','yyyy','keeplimits')
legend(Countries)
```