

API and mobile access to Statbank Denmark

ICBS 2014

Lars Knudsen
Statistics Denmark
lak@dst.dk

About the API

Access to 4000+ hypercubes from www.statbank.dk

- <http://api.statbank.dk> (documentation)
- <http://api.statbank.dk/console> (help console)

Features

- Supports streaming (BULK data)
- Supports saved queries made in www.statbank.dk
- Supports both POST and GET requests
- Supports JSON and XML for requesting

1) Choose API call:

Retrieve data

Retrieves the actual dataset.

2) Choose settings for call:

Request method

Language

☐ Use saved query

Table id

Format

Value p

Variable

Variable , value ids for selected values (comma delimited list)

☐ Download

3) Call the API with the chosen settings:

<http://api.statbank.dk/v1/data//CSV>

Execute

4) See the result from the API:

Prettify JSON ☒

Click [Execute] to see the result.

Diagrams

Line

Column

Stacked Column

Stack Column 100%

Bar

Stacked Bar

Stacked Bar 100%

Area

Stacked Area

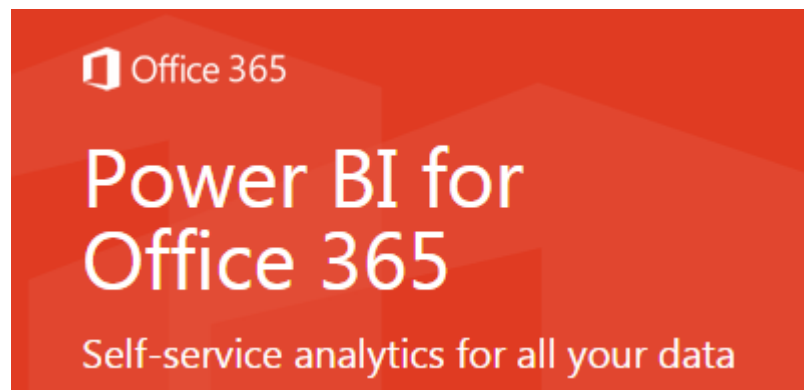
Stacked Area 100%

Pie

Column & Lines

Population Pyramide

Statbank API in ...

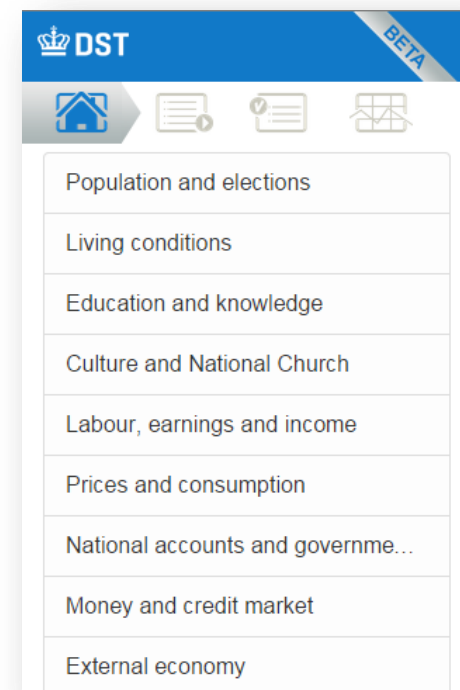


Mobile output database

- <http://m.statbank.dk>

User statistics show that users:

- Stay longer on the mobile version (1:46 / 3:04 min.)
- Bounce less (48 / 39 %)
- More page views (3.1 / 5.0)
- Higher rate of returning visitors (18 / 35 %)



From Excel

The screenshot shows an Excel spreadsheet with the following data:

A1		fx Labour force status (per cent) by seasonal adjustment, employment status				
A	B	C	D	E	F	G
1	Labour force status	(per cent) by seasonal adjustment, employment status, age, sex and time				
2	Unit: Per cent					
3				2013M08	2013M09	2013M10
4	Seasonally adjusted	Employment rate	Age, total	Men	66,4	66,7
5				Women	61	60,6
6			15-24 years	Men	51,9	52,3
7				Women	55,6	56,5
8			25-74 years	Men	69,3	69,7
9				Women	62	61,4
10		ILO-unemployment rate	Age, total	Men	7,4	6,7
11				Women	6,9	7,1
12			15-24 years	Men	15,9	14,4
13				Women	11,6	11,2
14			25-74 years	Men	5,9	5,4
15				Women	6,1	6,3
16		Economic activity rate	Age, total	Men	71,6	71,5
17				Women	65,5	65,2
18			15-24 years	Men	61,7	61,1
19				Women	63	63,6
20			25-74 years	Men	73,7	73,6
21				Women	66	65,5

The 'Demo, Statbank API' dialog box is open, showing a list of subjects:

- Population and elections
- Living conditions
- Education and knowledge
- Culture and National Church
- Labour, earnings and income
- Prices and consumption
- National accounts and government finances
- Money and credit market
- External economy
- Business sector in general
- Business sectors
- Geography, environment and energy
- Other

The dialog box has 'Cancel', '<<', and 'OK' buttons.

Reading data from api.statbank.dk into SAS

The code below lets you read data from api.statbank.dk into SAS. api.statbank.dk is an api provided by Statistics Denmark (Danmarks Statistik) for reading their public data.

The code below gets it's data from the table *FOLK1*. You can get a list of available tables from this url – <http://api.statbank.dk/v1/tables?format=CSV>. The code also sets some filters on the tables, some of the information used in this post comes from [here](#).

```
1 filename dst
2 url
3 'http://api.statbank.dk/v1/data/Folk1/CSV?valuePresentation=Value&Tid=*%K%C3%
4         proxy='http://<your proxy:and port>/'
5         debug
6         lrecl=8192
7         puser="<your userid to proxy>"
8         ppass="<your password to proxy>";
9
10 data test;
11     infile dst dlm=';' encoding="utf-8";
12     format Quarter Sex Municipality Age $50.;
13     input Quarter $ Sex $ Municipality $ Age $ Count;
14 run;
```

[From http://www.ryslander.com/reading-data-from-api-statbank-dk-into-sas/](http://www.ryslander.com/reading-data-from-api-statbank-dk-into-sas/)



🔗 branch: master ▾

dkstat / R / dst_get_data.R



leeper on 22 Jul roxygenize documentation; fix wrong args

2 contributors



45 lines (34 sloc) | 1.552 kb

Raw

Blame

History



```

1  #' This function returns the requested data from the StatBank table.
2  #' (http://www.statistikbanken.dk/statbank5a/ or http://www.dst.dk)
3  #'
4  #' Get data from a table in the StatBank.
5  #'
6  #' @param table Table from StatBank.
7  #' @param ... Build your own query.
8  #' @param query A request for data structured like the basic_basic_query.
9  #' @param lang Language. "en" for english and "da" for danish.
10 #' @param format for now just csv but later JSON and more.
11 #' @param value_presentation for know just ValueAndCode
12 #' @export
13 dst_get_data <- function(table, ..., query = NULL, lang = "en", format = "CSV", value_presentation = "ValueAndCode"){
14   dst_url <- paste0("http://api.statbank.dk/v1/data/", table, "/", format, "?")
15
16   dst_url <- parse_url(url = dst_url)
17
18   ## If query is NULL, then use ... as query
19   if(is.null(query)){

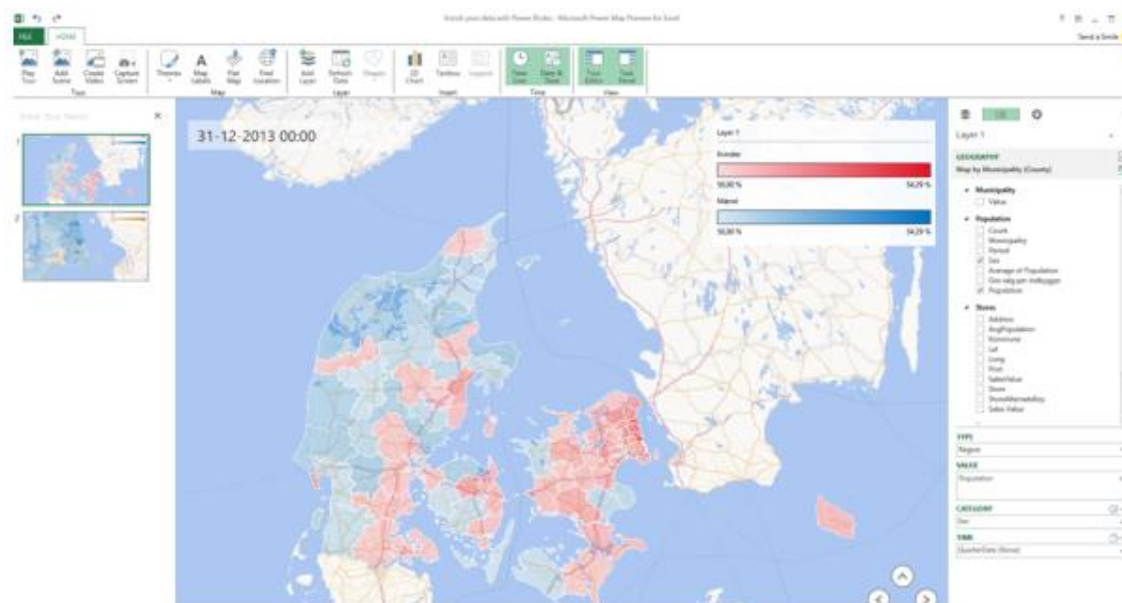
```

[From https://github.com/rOpenGov/dkstat](https://github.com/rOpenGov/dkstat)

Using Power BI to enrich your data using public datasets – part 1

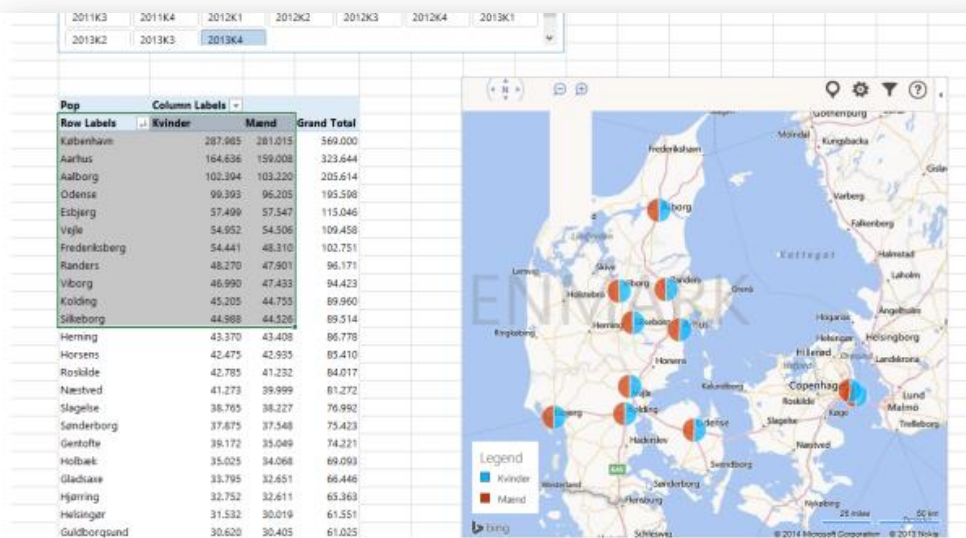
Yesterday I presented at a LinkedIn group – [Microsoft BI Professionals Denmark](#) a session about how you can use Power Query to enrich your datamodels using public datasets.

My demo contained examples on how to use the API from Statistics Denmark [link](#) – to extract all sorts of information about Denmark directly into your datamodel and how you can use the public dataservice <http://geo.oiorest.dk/> to do a reverse look up of Danish regions, municipalities from a specific lat/long coordinate.



You can download my demo files from [here](#).

<http://eriksvensen.wordpress.com/2014/01/14/using-power-bi-to-enrich-your-data-using-public-datasets-part-1/>



Part 2, Using:

- Power Query
- Power Pivot
- Power View
- Power Map

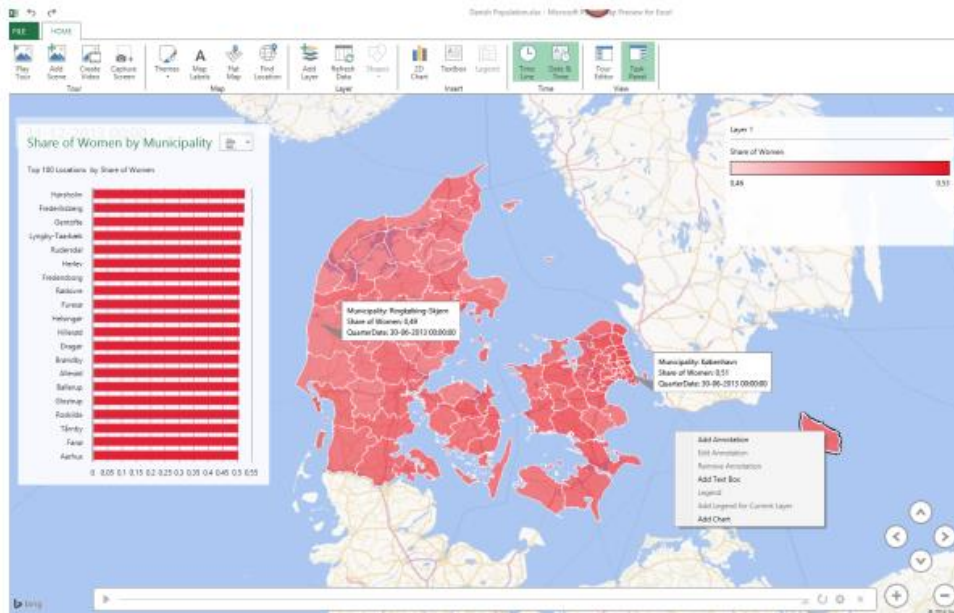
... for Excel

Another way could be in a Power View Report in Excel





And change it to County – and now we have the Danish municipalities



And final result showing the 2 municipalities and that the share of women differs with 2 percentage points.

This is just one of the tables in Statistics Denmark – So if you want to analyze Denmark – Use Power Query and the API – very powerful.

Hope you like it.

Part 2, Using:

- Power Query
- Power Pivot
- Power View
- Power Map

... for Excel