Macroeconomic History Data Project

Alan Freeman 9 December 2020

> Contact address for the project: **alan.freeman@umanitoba.ca**

What is the project?

This purpose of this project is to create a user-friendly database of world and country economic, political and social indicators, drawn from a variety of sources, and create a community of users who will add to the data, and work to improve it.

This implements the objectives of the GERG (Geopolitical Economy Research Group) Data group.

This repository can be used in two ways by two audiences

1. folks who only want the results. These are in the \EXPORTS folder. There are also experimental results in the \DATA\TESTS folder, which are 'Beta' so no promises.
2. folks who want to build their own copy of the database and possibly modify it by adding new data (which is always welcome on the site so feel free to push any modified version.

Where can you see the results?

- There are four ways to access the results, in order of complexity
- The simplest is to download one or more of the Excel files in the `DATA\EXPORTS` folder. Until the project is more stable, these will vary from time to time as we create new datasets (we haven't yet developed an adequate versioning system). But this will give you the chance to see what the project can achieve and how it is used.
- The second is to download the Power-BI books that we will also place in the `DATA\EXPORTS` folder. Again, until the project is stable these will vary, but they will show you the kind of visualizations we are working on
- The third is to access the shared workspace (on the web) of the Power-BI side of the project. Only one snag here: it's not ready yet. But things are moving along: watch this space and drop us a line (contact address at the top of this document) if you'd like to be kept informed.
- The fourth, if you are a coder, is to download this project from Github and run it yourself to produce the complete database locally, which you can then use as you please. In particular (new at 23 October) you can build an Analysis Server which has all the filters and hierarchies ready-made.

Note (9 December)

An 'addin' called `Dispersions Add-In.xlam` has been created to help calculate dispersion measures (Theil, Gini, weighted quartiles). It's in the `SETUP` folder. It drastically simplifies the spreadsheets, but it means if you want to play with the data (as opposed to just viewing it) you will need to install it into your copy of any exported workbook locally.

I'm working on more user-friendly options, and probably will rewrite it as a proper extension that you can get from AppSource. Another workaround would be to put the code in each of the exported result workbooks, but here the disadvantage is that if the code is modified, the workbooks will still contain the old code. Some brief notes on how to install are at the end of this document under 'change notes'

All I can say is 'watch this space'

What are the system requirements?

If you only want to look at the results, you will need the following software:

For Excel, you'll need to make sure your version has the 'Power Pivot' add-on. Nowadays this is bundled with the download but you may have an older version, in which case you'll need to install it

For Power BI you can use a free account (we hope) though if you want to make, and publish, changes or new visualizations, you may have to upgrade to the 'Pro' version

For the 'do it yourself' option, read on...

The code assumes you have installed, probably on a Windows 10 system, the following:

- Microsoft SQL server 2017 or later
- Visual Studio 2019 or later with the SSIS (Integration Services) add-on

It also assumes you know how to use the above software.

How can you make (and modify) your own copy of the database?

From Visual studio, open the 'Macrohistory' solution

Modify the parameters so they point at your SQL server and at the directory containing this clone.

In you use SQL server, create two databased called `macrohistory_oltp`and `macrohistory_rolap`. The data is initially loaded into the OLTP database by two ETL packages called `Setup OLTP Dimension Tables` and `Import OLTP data from External sources` (which should be run in this order). Once that's done, your OLTP database contains a standardised version of the data. This is then copied to the ROLAP database using the package Populate `ROLAP tables from OLTP` in such a way that all the Foreign Keys work properly, which streamlines the data by compacting it and organising it in a star model. As a result, the entire database takes up less than 100MB of space when imported into Excel PowerPivot or PowerBI. By selecting subsets of this data, you can create smaller workbooks in either format, for special purposes.

As of 23 October, the companion Analysis Server (SSAS) project is integrated into this project on github . This uses the data in the SQL server to create a model with the correct measures and hierarchies for the data. You don't have to use it to look at the data, but it helps, because it shows how the model is constructed from the raw data. You can run this in Visual Studio and analyse it with Excel without going to the SSAS server, but I've found it does help to have an SSAS server instance and deploy to it.

Also, if you install the Power BI gateway, you can create Power BI workbooks using direct query to your own analysis server, publish these to Power BI Service, and access the published workbook while it draws its data from your own copy of analysis server. My eventual aim is a public Analysis server that will be common to all workbooks, but we're not there yet.

You'll find that you need to run the first task in each of these packages once, to create the initial tables. While you are doing that, the other tasks will be flagged as having errors, because they expect these tables to exist. If you then close the package and re-open it, the error flags should disappear. There's probably a more sophisticated way of doing this but it doesn't take up much time, so that's for a future tweak. Or maybe you can do it and push your improvements back up to the repository.

That's it, basically. Let me know if you run into problems. Some technical stuff below, somewhat outdated by the widespread use of DAX and other table-transforming software. But it could be a useful place to start.

What is OLAP technology?

OLAP technology, which underlies this project, provides (reduced to its simplest elements) three features that have slowly emerged from preceding database technologies usually referred - somewhat confusingly - as ROLAP, short for Relational Online Analysis Processing.

See This Article for more information (there are many others)

OLAP servers generally distinguish between the quantitative information in the data, which it terms **measures**, and the qualitative description of these measures, termed **dimensions**. A further sophistication is offered by **hierarchies** which organise the qualitative information in the dimensions in human-friendly structures, so that users can interrogate the data in conceptually natural ways.

Quite recently, Microsoft has made the basic functionality of OLAP servers available to most users of its Office suite, by incorporating a facility it calls 'Power Pivot' in its spreadsheet product, Excel. However, data stored in Excel spreadsheets is not shared (unlike Wikipedia) so it's still not possible for a community of users to interact with the data, discuss it, try out various changes to it, and compare the results. This is our eventual aim. Power BI interests us because it is promising to deliver a model for doing exactly the above. But it's not there yet, at least from our experience so far.

Excel is not very good at 'uploading' data from a range of sources, a process known by the jargon word 'ETL', short for 'Electronic Transfer and Load'. This facility is however available with more sophisticated products, in particular Microsoft SQL Server (MSSQL Server), which is currently used to construct the data.

In what follows below we provide a bit more detail on how MSSQL Server is used; you don't need to know this unless you intend to work on the project as a developer.

MSSQL services

The project uses two services provided by Microsoft SQL server: Database services and ETL, which Microsoft refers to as 'SSIS' or sometimes 'SSDS'. To implement the project, at least one instance of MSSQL server providing database services is required, and Microsoft's IDE, called 'Visual Studio' is needed to build the ETL packages.

Visual Studio and ETL

The project is implemented on Visual Studio (2017) with SSIS (Short for SQL Server Integration Services). This allowed me to implement a set of ETL procedures which take raw data, from a variety of sources, and transform it into a standard form.

Because you will install the project on your own database with the files in your own folder, you need to create the databases and modify the project parameters

1. Create two databases (easiest is to put them on one server but they can be on two) called `macrohistory_oltp`and `macrohistory_rolap`.
2. Open 'Macrohistory.sln' in Visual Studio
3. Open 'Project Parameters' and modify the server and root directory properties so that the project can find your server and the source files, at the location where you have downloaded them into your setup

The ETL files contain all the setup objects needed to load the server, using Visual Studio 2017 (VS) or later.
This includes a set of SSIS packages which, when run from within VS, construct the service.

The ETL procedures transform the raw DATA/SOURCE into the above form. The raw DATA/SOURCE ORIGINALS are all in the public domain; the project's cleaned-up copy of these datasets are stored in the DATA/SOURCE folder.

Change Notes

13 September: Created 'EXPORTS' folder to hold datasets and other material to be used in the GERG Data Laboratory Project

25 September New Date Handling System: the Date dimension no longer exists. All Years are converted to MSSQL DateTime format, stored in the Fact Table so there is no need for a separate dimension table.

29 November The hierarchies are being continuously updated. This takes effect in the AS project which is inside the folder structure

9 December The `Change Tracking.xlsx` workbook in `CHANGE TRACKING` now attempts to keep a systematic record of changes including a brief description, the status of the proposed change, which commit it appears in, and how or whether it has been tested, along with any problems noted.

9 December A separate add-in was created to handle Theil, GIni and weighted quartile calculations. This slightly complicates the business of reproducing the data, in that the user now has to install the Add-in. A copy is stored in the `SETUP` folder (called `Dispersions Add-In.xlam`) and this has to be brought into any excel book where this functionality is needed. The add-in has to be copied into a local user folder here: `C:\Users\[your user name]\AppData\Roaming\Microsoft\AddIns`. It should then be brought into the workbook via the File>Options>Addins dialogue and will appear as an available Excel Add-in. Eventually it's proposed to convert this to a distributable form to be downloaded from AppSource or 'Sideloaded' from the project. Since this is under development, we'll document the procedure when it is more stable