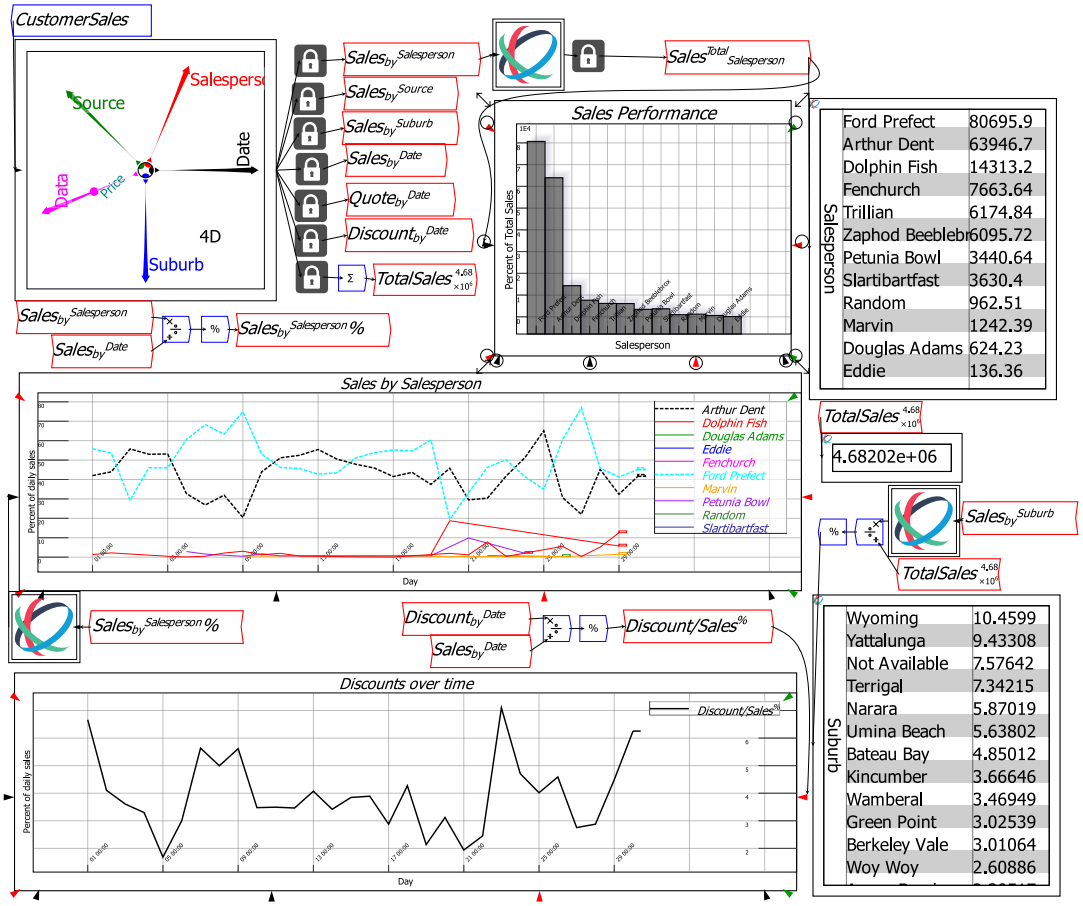
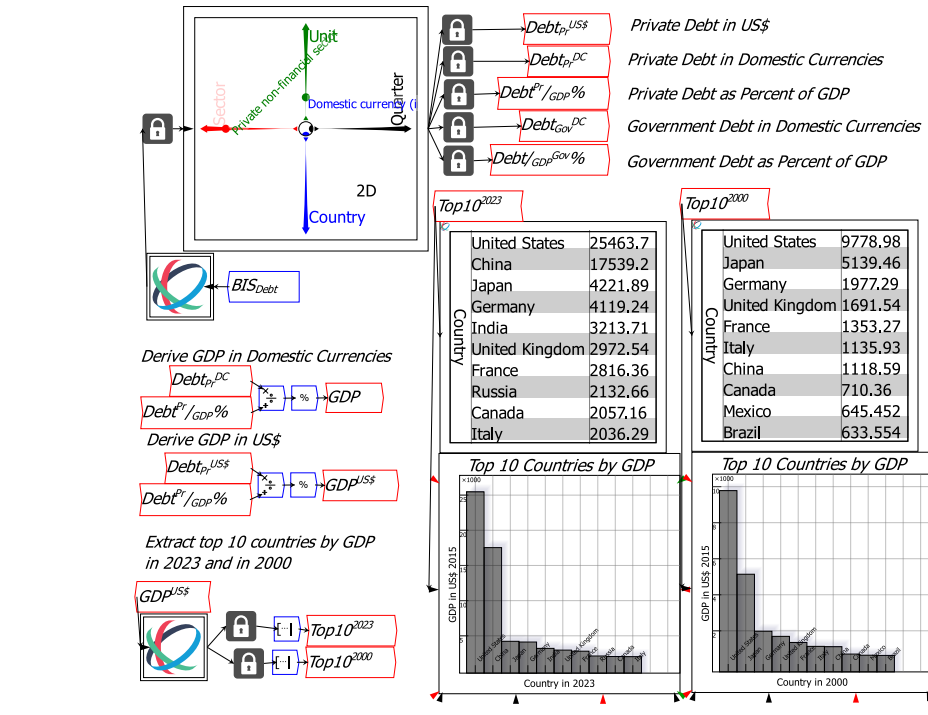
The Little Book of *Ravel*ations

An Introduction to Data Analysis with Ravel





For sales enquiries, please email sales@ravelation.net

Contents

Introduction

Ravel is a new, entirely visual, "no-code" way of analysing any type of data. Broadly speaking, today there are 3 types of computer programs used for data analysis:

1. Spreadsheets (*Excel*, *Google Sheets*, etc.);
2. Business Intelligence programs (*Power BI*, *Tableau*, etc.); and
3. Data-oriented programming languages (*Python*, *R*, etc.)

These give you three basic tools for data analysis:

1. Formulas using cell references;
2. Pivot Tables; and
3. Computer code

Ravel is a whole new type of data analysis computer program, based on its unique and patented GUI ("graphical user interface") object, the Ravel.

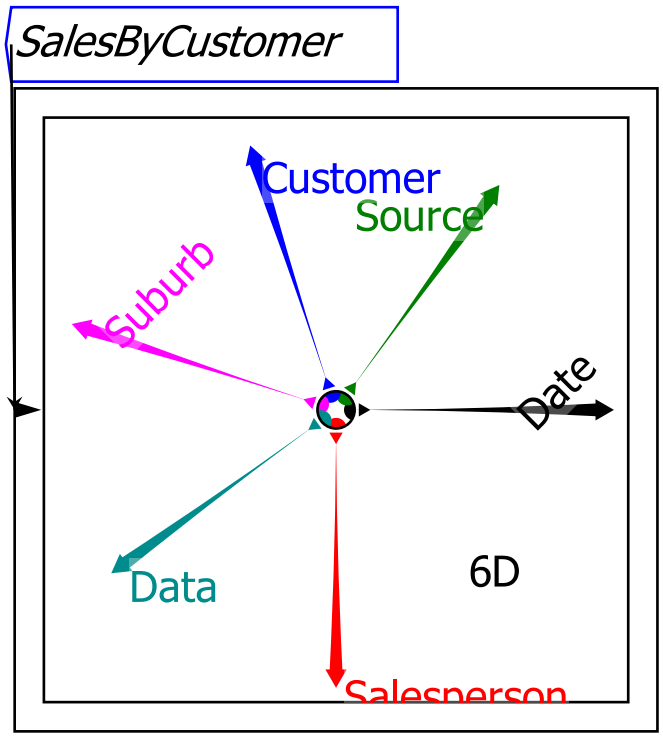


Figure 1: A sample Ravel with six dimensions

And it provides a new tool for data analysis, flowchart equations.

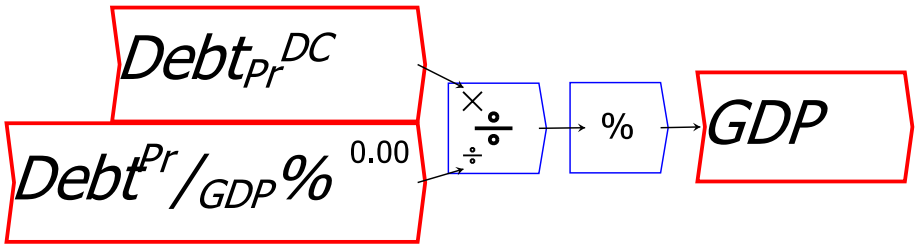


Figure 2: Divide debt in domestic currency by the debt to GDP ratio to derive GDP in domestic currency

The combination of the Ravel with flowchart equations makes data analysis both far easier, and far more powerful, than it is with spreadsheets or standard BI programs. The equivalent cell reference formulas would be both hard to read, and hidden behind the thousands of numbers they would generate.

These are *Ravel*'s two super-powers, in comparison to spreadsheets, BI programs, and data-oriented programming languages:

- A Ravel has as many dimensions as your data, whereas a spreadsheet has only two dimensions: rows and columns; and
- In place of invisible and unauditable cell formulas in a spreadsheet, or complex lines of code in a programming language, there are flowchart formulas which are easy to create and read.

What data is on the menu?

We will use economic data in this tutorial, since it's both easily accessible and publicly available. But Ravel can handle data on any topic, from marketing to palaeontology. Here are 2 examples.

Business

Businesses need to know what the characteristics of its best customers are, which marketing methods are working best, how good profit margins are, which salespeople are performing best, and so on. Many small businesses collect transactional data using *MYOB* and similar programs. But they often don't analyse it, because it's just too complicated to do that with spreadsheets, and they can't afford to hire a dedicated data analyst who can use BI programs for them. Figure 3 shows a sample data file from a real business with anonymized records.

Date	Salesperson	Source	Suburb	Quote	Discount	Price
06/04/2024	Ford Prefect	Bartercard	Wyoming	8967.21	876.3	8090.91
07/04/2024	Slartibartfast	Business Referral	Not Available	227.27	0	227.27
09/04/2024	Arthur Dent	Business Referral	Bateau Bay	3766.6	174.84	3591.76
10/04/2024	Ford Prefect	Business Referral	Gosford	1285.57	0	1285.57
11/04/2024	Arthur Dent	Business Referral	Wyoming	568.9	0	568.9
12/04/2024	Ford Prefect	Business Referral	Bateau Bay	300.98	0	300.98
12/04/2024	Arthur Dent	Business Referral	Yattalunga	470.35	0	470.35
14/04/2024	Arthur Dent	Business Referral	Yattalunga	3996.35	178.17	3818.18
30/04/2024	Ford Prefect	Business Referral	Bensville	1902.5	0	1902.5
02/04/2024	Arthur Dent	Car Signage	Narara	6733.2	641.38	6091.82
02/04/2024	Ford Prefect	Car Signage	Wamberal	2508.6	190.42	2318.18
04/04/2024	Trillian	Car Signage	Wyoming	309.09	0	309.09
09/04/2024	Ford Prefect	Car Signage	Wyoming	2752	161.09	2590.91
11/04/2024	Ford Prefect	Car Signage	Wyoming	6881.2	381.05	6500.15
15/04/2024	Arthur Dent	Car Signage	Narara	4881.4	154.13	4727.27
19/04/2024	Ford Prefect	Car Signage	Booker Bay	2722.76	86.4	2636.36
19/04/2024	Arthur Dent	Car Signage	Not Available	1720.55	0	1720.55
24/04/2024	Ford Prefect	Car Signage	Chain Valley Bay	3289.72	585.63	2704.09
01/04/2024	Ford Prefect	Drive/Walking Past	Aberglasslyn	1455.36	0	1455.36

Figure 3: Sales data for a small business

Figure 4 shows this data loaded into *Ravel*. The first four columns of the spreadsheet—Date, Salesperson, Source, and Suburb—become dimensions in *Ravel*, while there are 3 pieces of information—Quote, Discount, and Price—which are aggregated onto a 5th dimension named "Data".

Several slices of data are made for further analysis: each of the locked variables next to the Ravel is equivalent to a Pivot Table in Business Intelligence programs like *Power BI* and *Tableau*.

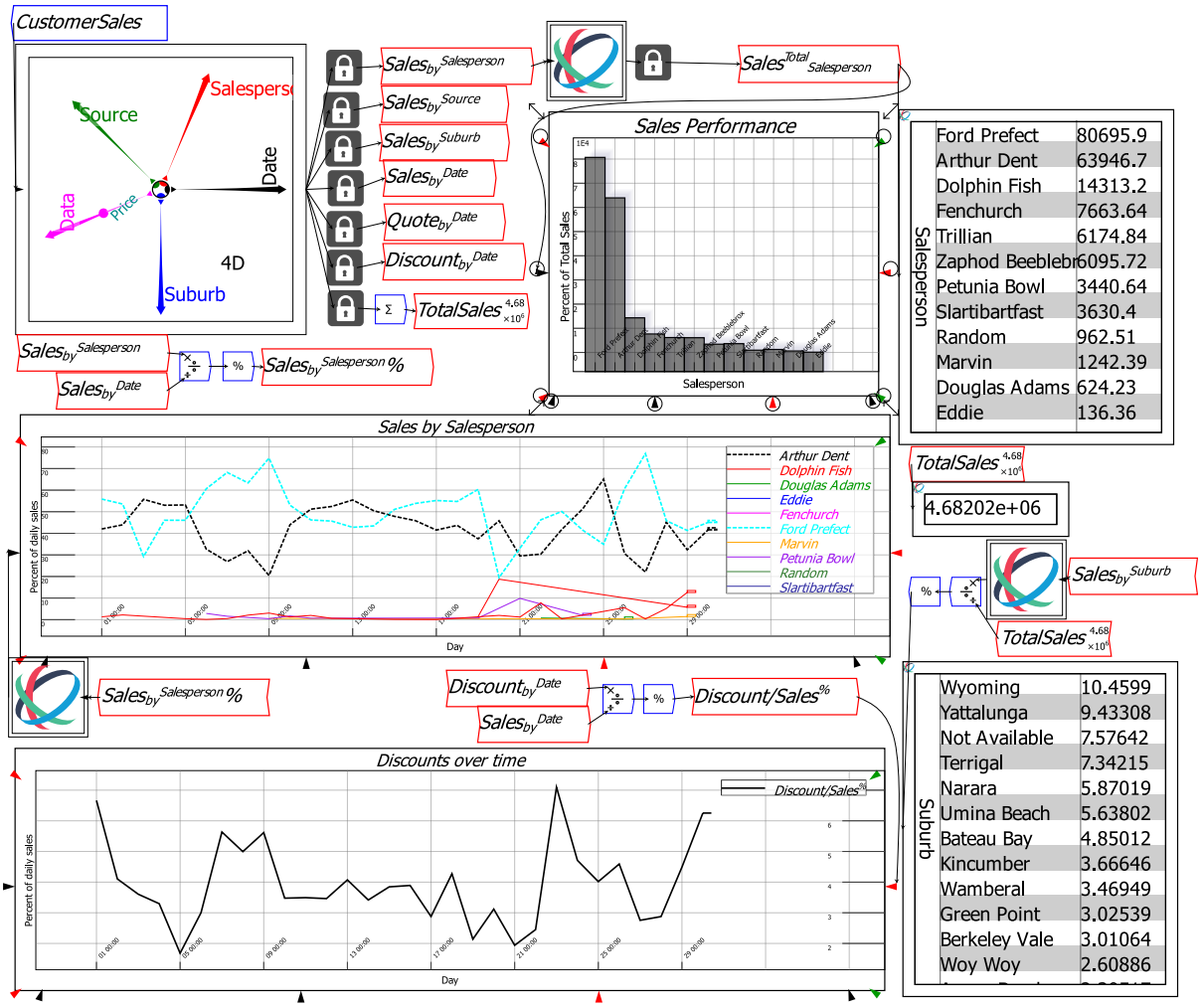


Figure 4: Analyzing sales data with Ravel

Figure 4 uses just the first slice to analyse the performance of the company's salespeople. The other slices can be analysed to work out the best performing suburbs, trends over time, etc. And all the while, the formulas doing the analysis are easily to read, understand, and audit.

Science

There are numerous studies that attempt to measure the temperature of the Earth's atmosphere over time. Figure 5 shows four such estimates, ranging from average annual temperature measurements from 1880 onwards,[1] estimates of global average temperatures for every year from 0AD till 2000,[2] a reconstruction of temperature every thousand years back to 2 million

[1] Lenssen, N., G. Schmidt, J. Hansen, M. Menne, A. Persin, R. Ruedy and D. Zyss (2019). "Improvements in the GISTEMP uncertainty model." *J. Geophys. Res. Atmos* 24: 6307-6326, and Team, G. (2024). GISS Surface Temperature Analysis (GISTEMP), version 4. NASA Goddard Institute for Space Studies. See https://data.giss.nasa.gov/gistemp/.

[2] Neukom, R., L. A. Barboza, M. P. Erb, F. Shi, J. Emile-Geay, M. N. Evans, J. Franke, D. S. Kaufman, L. Lücke, K. Rehfeld, A. Schurer, F. Zhu, S. Brönnimann, G. J. Hakim, B. J. Henley, F. C. Ljungqvist, N. McKay, V. Valler, L. von Gunten and P. k. Consortium (2019). "Consistent multidecadal variability in global temperature reconstructions and simulations over the Common Era." *Nature Geoscience* 12(8): 643-649. See https://doi.org/10.1038/s41561-019-0400-0.

years BCE,[3] and estimates of temperature every million years back to 540 million years ago.[4] The first three series show deviations from a benchmark average, while the final series shows estimated surface air temperatures in degrees Celsius.

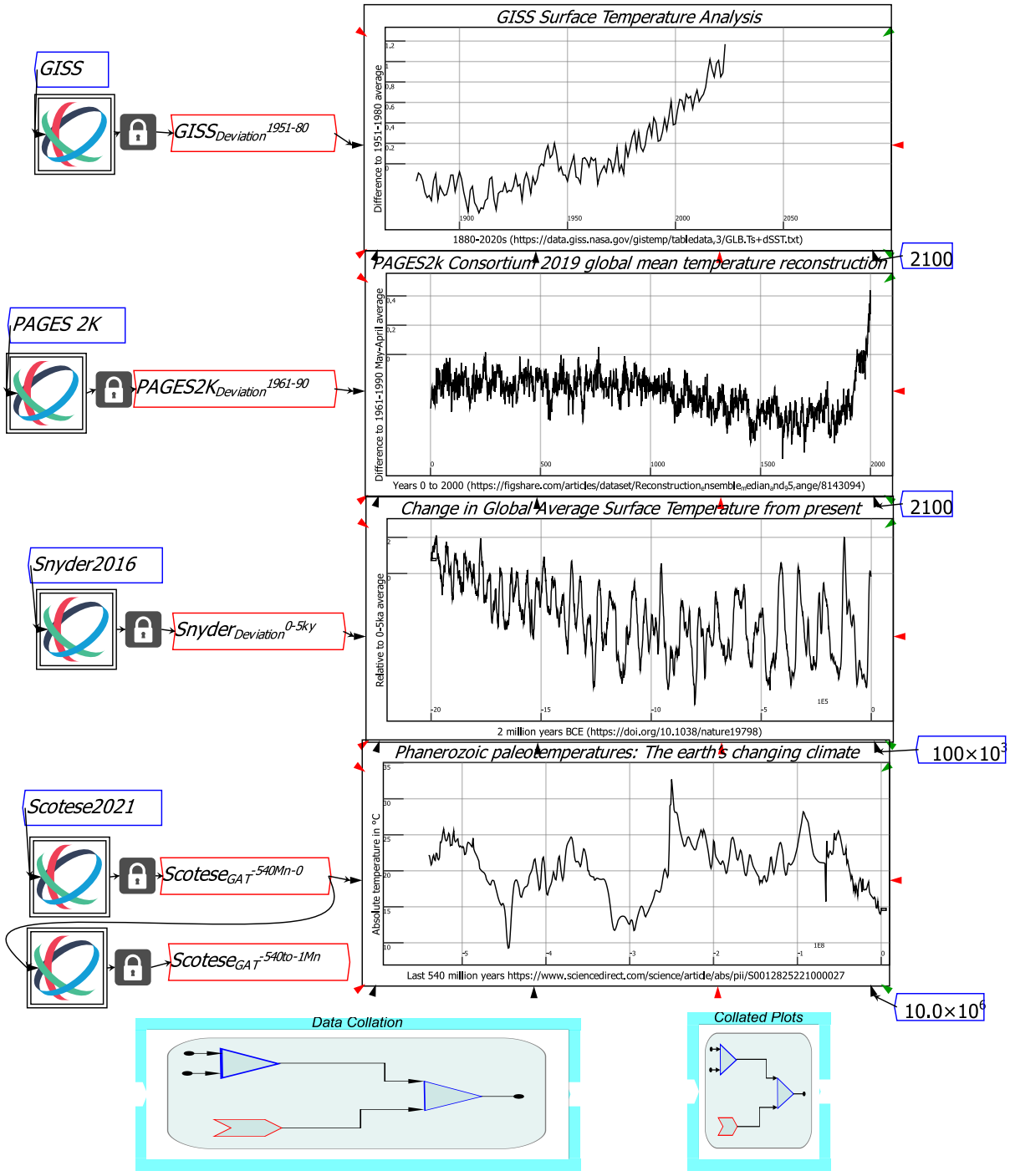


Figure 5: Four data sets of global average temperature giving data by years, thousand years, and millions of years

[3] Snyder, C. W. (2016). "Evolution of global temperature over the past two million years." Nature 538(7624): 226-228. See https://doi.org/10.1038/nature19798.

[4] Scotese, C. R., H. Song, B. J. W. Mills and D. G. Van Der Meer (2021). "Phanerozoic paleotemperatures: The earth's changing climate during the last 540 million years." *Earth-Science Reviews* 215: 103503. See https://dx.doi.org/10.1016/j.earscirev.2021.103503.

The formula shown in Figure 6 collates these into one data series from 540 million years ago to 2023, with time steps ranging from millions of years to single years.

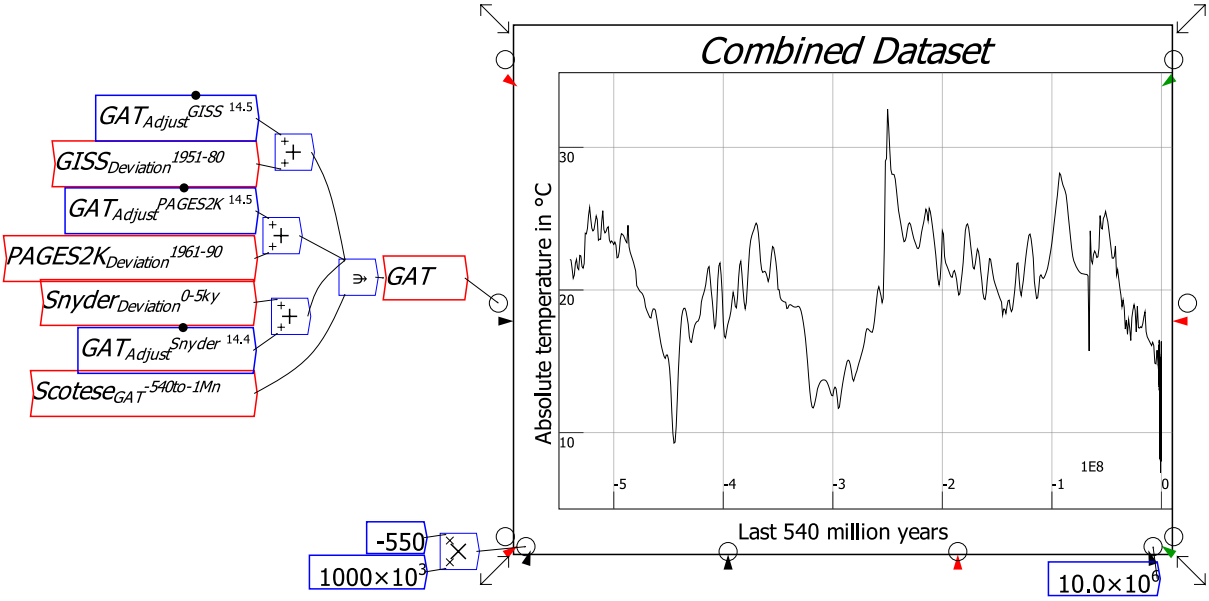


Figure 6: The data series adjusted and collated using Ravel flowchart formulas

This enables plots ranging from all years from 0 to 2023 (the top plot in Figure 7), to every million years since the extinction of the dinosaurs (the last plot), to be easily generated from the collated data.

Figure 7 emphasises just how briefly human civilisation has existed: all of human sedentary civilisation, since the oldest known monolithic buildings were built roughly 12,000 years ago in modern-day Turkey, occurs in one brief warm spell at the top of a 6°-degree “heat hill”, in the middle of a highly unusual period of glaciation cycles over the last 1.5 million years.

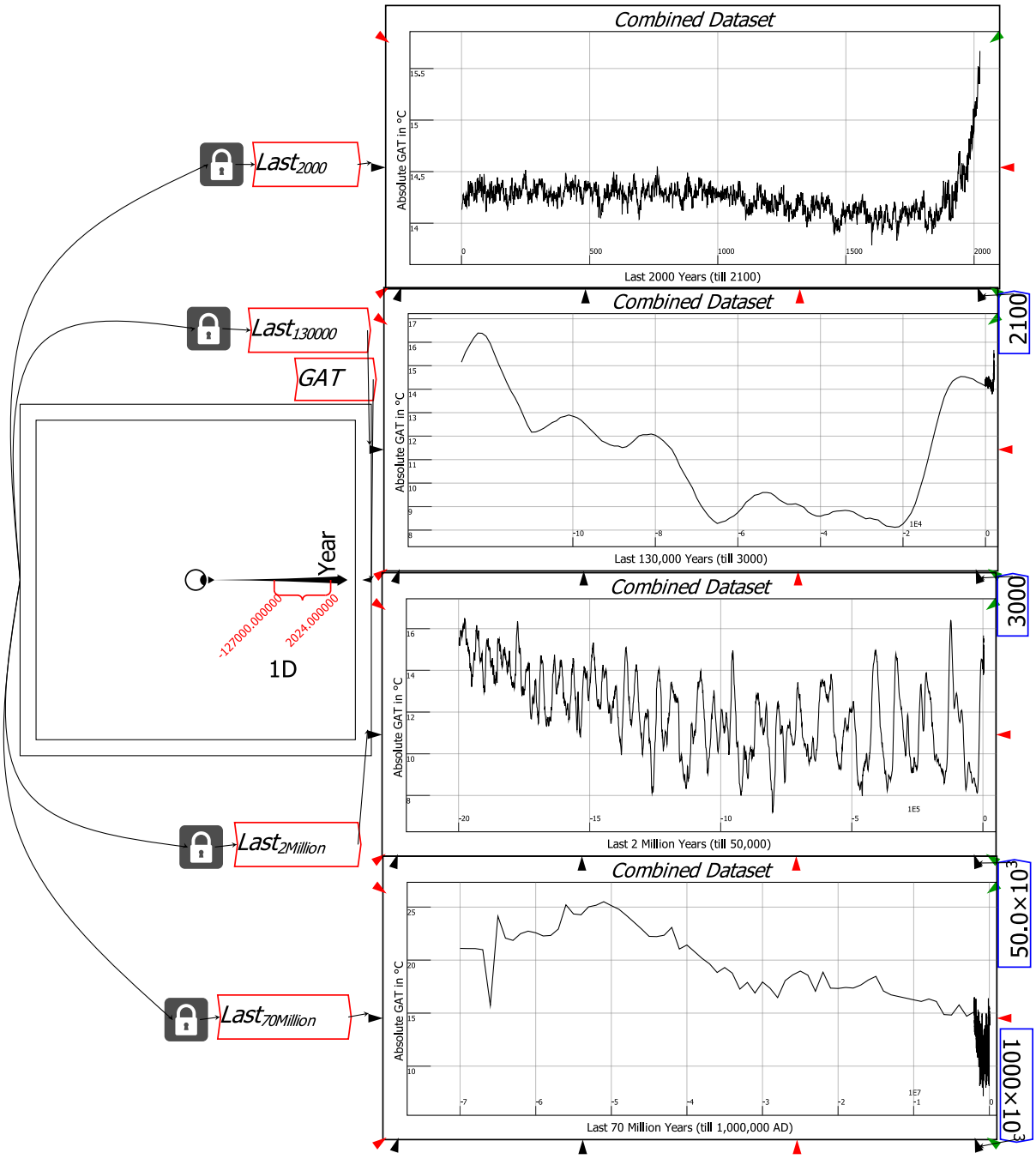


Figure 7: Human civilisation has existed for a brief and highly unusual period in the Earth's climate history

There are many free databases of economics information, and in the remainder of this tutorial guide, we'll mainly use economic data to demonstrate how to use *Ravel*. But as these examples from small business and climate research show, *Ravel* can analyse any type of data, more easily, powerfully, and transparently, than can be done with a spreadsheet or BI program, or via a data-oriented programming language.

Economics and Finance

The *Bank of International Settlements* provides many free databases, on banking, debt, house prices, consumer prices, and interest rates (see https://data.bis.org/). One piece of information the BIS lacks though is GDP. However, since there is data on debt in each country's domestic currency (and also in US dollar terms), and debt as a percentage of GDP, it is possible to derive

GDP data—in both domestic currencies and the US dollar—by a simple pair of formulas. If you divide debt in domestic currencies by the debt to GDP ratio in percentage terms, and then multiply the result by 100 (to compensate for the fact that, by using percentages, you're dividing by 100 times the debt ratio), your formula gives you GDP in domestic currency terms. The same trick also works for debt in US dollar terms—see Figure 8.

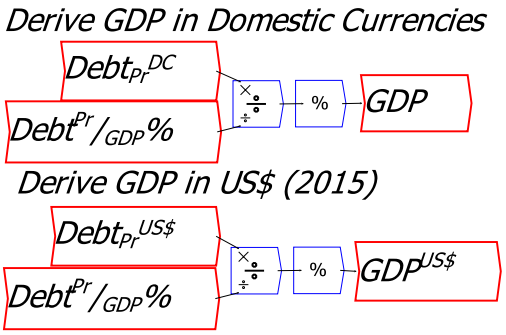


Figure 8: Two simple flowchart formulas that replace tens of thousands of cell reference formulas

The same calculations could be done in a spreadsheet, but the relevant cell reference formulas would need to be replicated tens of thousands of times—once for each combination of the 43 countries and 333 quarters in the BIS database. Figure 9 shows this formula in Ravel, and its use to find the ten biggest economies in the world in 2000 and 2023.

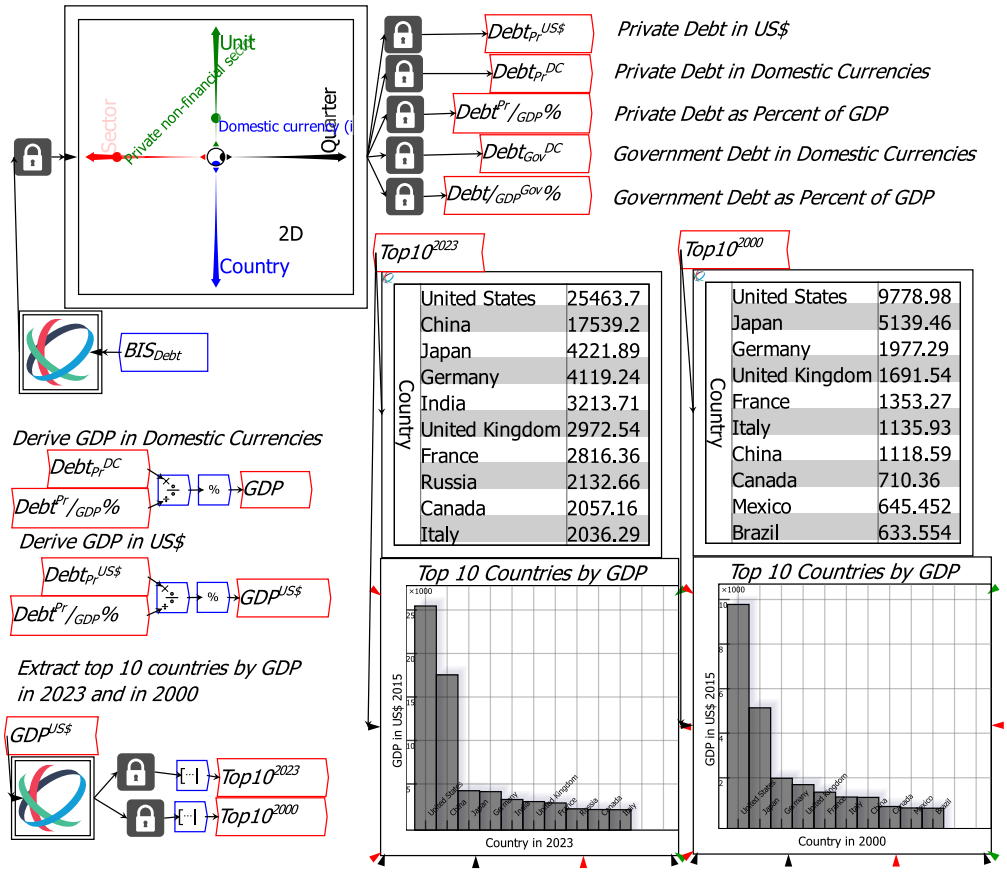


Figure 9: Deriving GDP in US$ for the 43 countries in the BIS Database

Getting Started

The figure below shows the Ravel interface, with five objects placed on it, and five strings of text. You place objects on the canvas by clicking on the relevant icon in the widget bar (and its sub-menus, which pop out above and to the right of widgets when needed), and then clicking where you wish to place them on the canvas. Text can be entered either by clicking on the text icon  , or by typing a hash character # on the canvas and then typing—the text after the # will be treated as a text string.

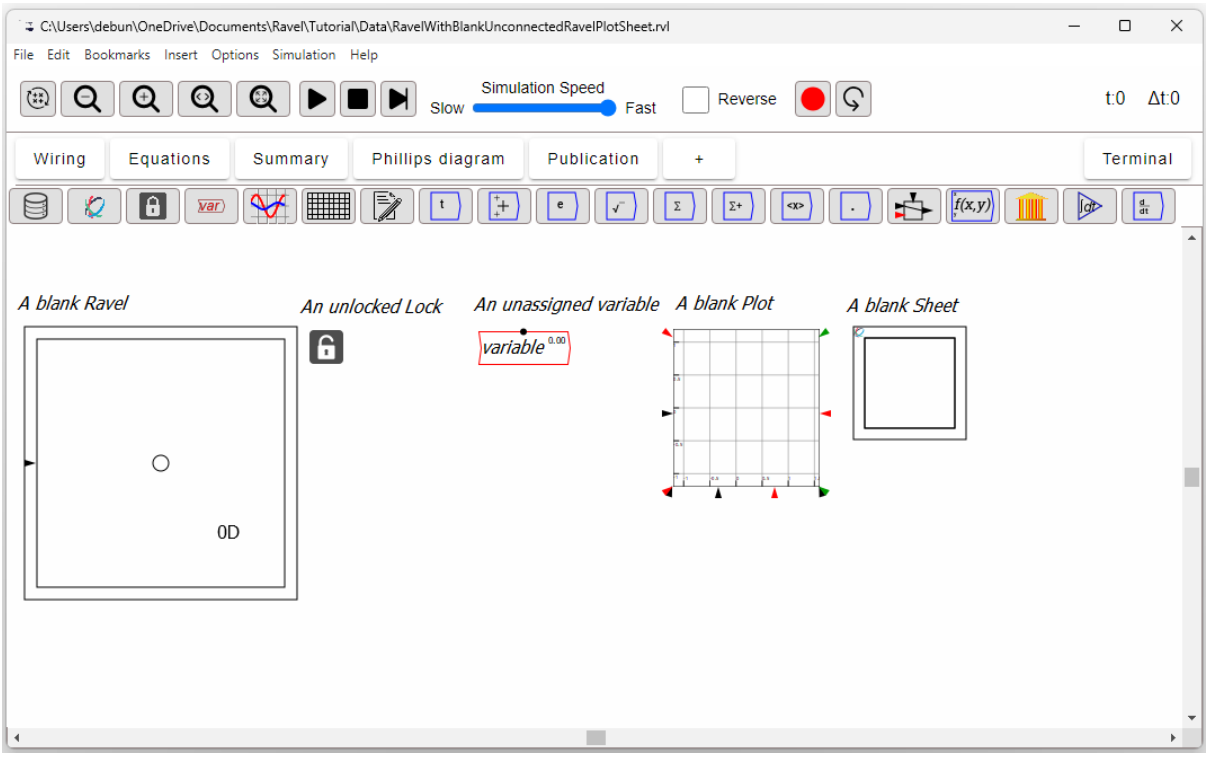
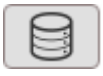


Figure 10: A blank Ravel, Plot and Sheet

Ravels, Locks, Variables, Plots and Sheets are created by clicking on the 2nd to 6th widgets respectively in the widget bar directly above the canvas—see Figure 11:

Figure 11: The key widgets for importing data, creating a Ravel, Plot and Sheet

The first widget  starts the import data process, which we discuss at the end of this tutorial.

Objects in *Ravel* are linked to each other using arrows, which are drawn by clicking in the output port of one widget and connecting to the input port on another. You must left-click in the circular output port to create an arrow, but you needn't click precisely in the input port to complete it: *Ravel* will snap to the nearest output port when you release the mouse button. Wires can be curved, as explained later in this tutorial.

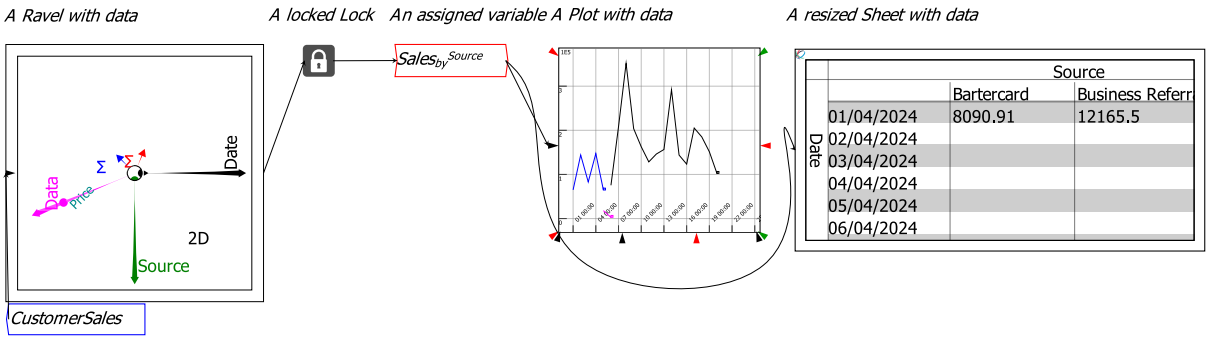


Figure 12: Ravel objects wired together

Transcend Cells, Sheets & Tabs with Ravel

The basic entity in a spreadsheet is the cell, and data analysis in a spreadsheet requires the use of cell reference formulas. It's easy to write simple formulas in a spreadsheet, but the task becomes unwieldy with complex formulas, or once you need more than simply row and column coordinates to characterize your data. Cell reference formulas are also inherently unauditable.

Spreadsheets and "Business Intelligence" (BI) programs like *Power BI* and *Tableau* get around these limitations using Pivot Tables. But most spreadsheet users don't touch the Pivot Table feature in Excel and its rivals because they find them difficult to use, while only professional data analysts use BI programs.

The basic entity in *Ravel* is an inherently multi-dimensional object we call a Ravel—see Figure 13, which shows a 4-dimensional Ravel.

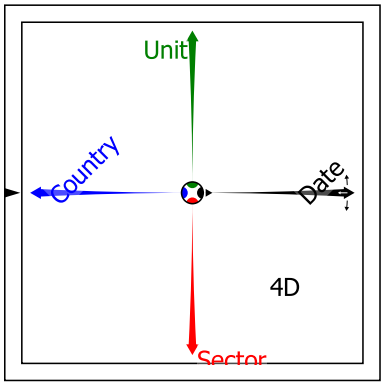


Figure 13: A Ravel with 4 dimensions

Manipulating a Ravel is far easier than using a Pivot Table. You use the axes of the Ravel to control the data it outputs. Figure 14 shows that same Ravel, set up to output the average private non-financial sector debt level across the 43 countries in the database, as a percentage of GDP, for all quarters between 1955 and 2022.

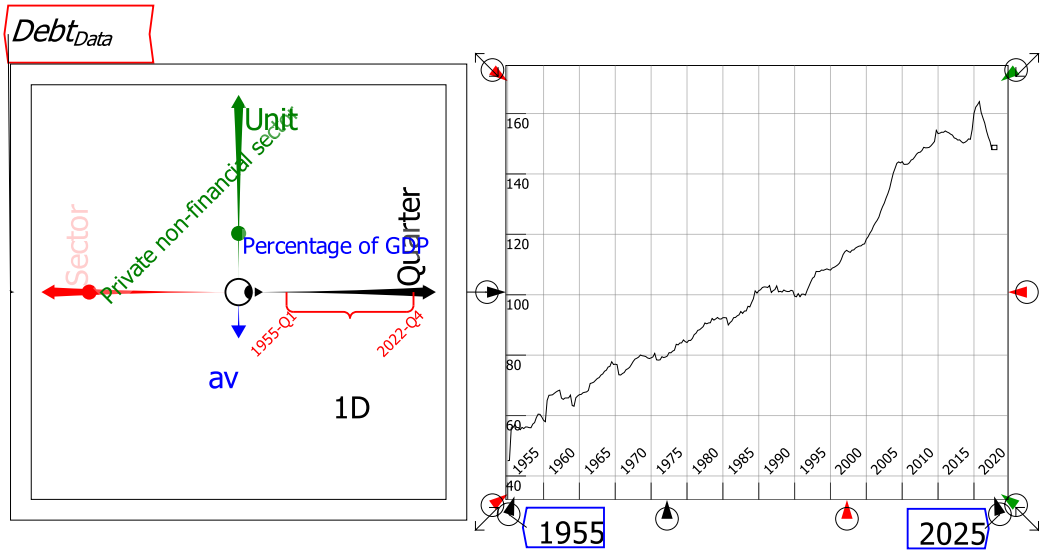


Figure 14: A Ravel returning the average private debt level across 43 countries, between 1955 and 2022, as a percentage of GDP

Slicing, Dicing and Rotating Data

When a Ravel is first attached to data, it outputs the entire data set—which is indicated by the dimension count **4D** in the lower right quadrant of the Ravel in Figure 13.

The BIS database on house prices has twelve dimensions, most of which are superfluous; these were reduced using the data import procedures explained later in this tutorial, to leave just four significant dimensions:

1. *Date*: Quarterly data from 1927 till 2024 (388 entries)
2. *Unit of Measure*: House Price Index (2010 = 100), and annual inflation rate (2 entries)
3. *Value*: Nominal or Real (CPI-deflated) prices (2 entries)
4. *Reference Area*: Country or Region (62 entries)

You see the data in a Ravel by attaching a Sheet or Plot to it output port:

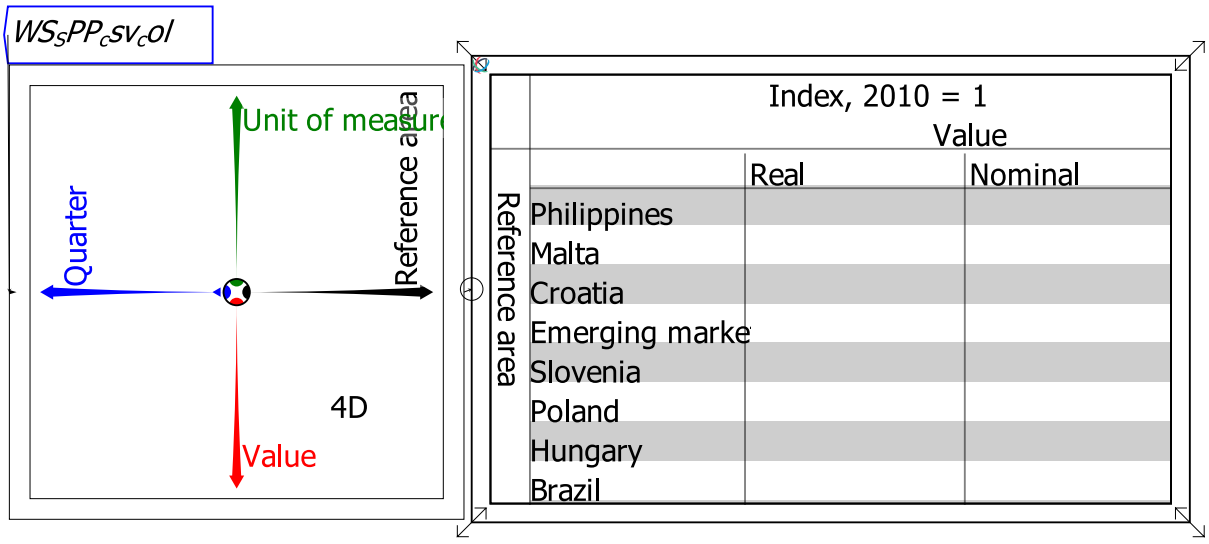


Figure 15: Output from a Ravel before data selection and axis rotation

The right-pointing axis of the Ravel determines what is shown on the rows of the sheet, while the down-pointing axis determines what is shown by the columns. At present, the displayed axes are *Reference Area* and *Value*. The sheet shows a slice of that data for the first entries in the file: Nominal on the *Value* axis, Index from the *Unit of Measure* axis, and the first two entries in *Reference Area*, and the first eight entries from *Quarter*.

It would be more useful to see the data by *Reference Area* by *Quarter*. To get that view, click the left mouse button on the arrowhead of the *Quarter* axis, hold the button down, and rotate the axis into the down direction, which is currently occupied by the *Value* axis. When you release the mouse button, the *Quarter* axis will replace the *Unit of measure* axis in the down direction, and the data in the Sheet will now have countries by rows and Quarters by columns.

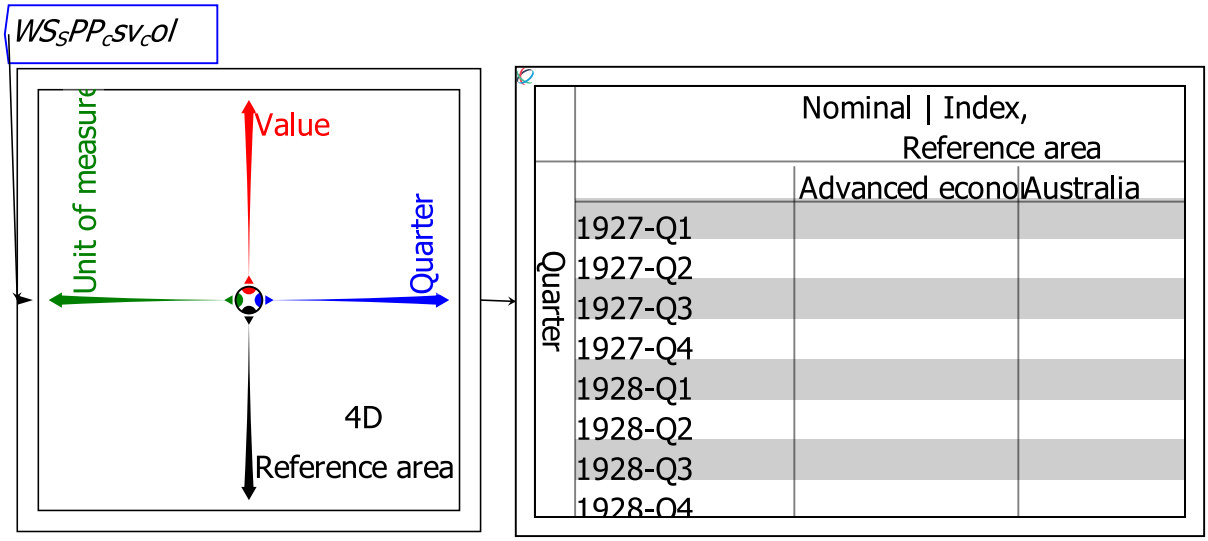


Figure 16: Output from a Ravel after Pivoting the data cube via rotating the axes

The Sheet is still blank, because there is no data for the current selections—the Index for Nominal House Prices in the very first Quarters of the data set (1927 and 1928), for the first few countries in the file (in alphabetical order).

To see data immediately, take advantage of a feature of a Sheet: it can display the first few rows and columns of data (the default setting), which we call the Head, the last few (the Tail), or a few of both (Head and Tail).

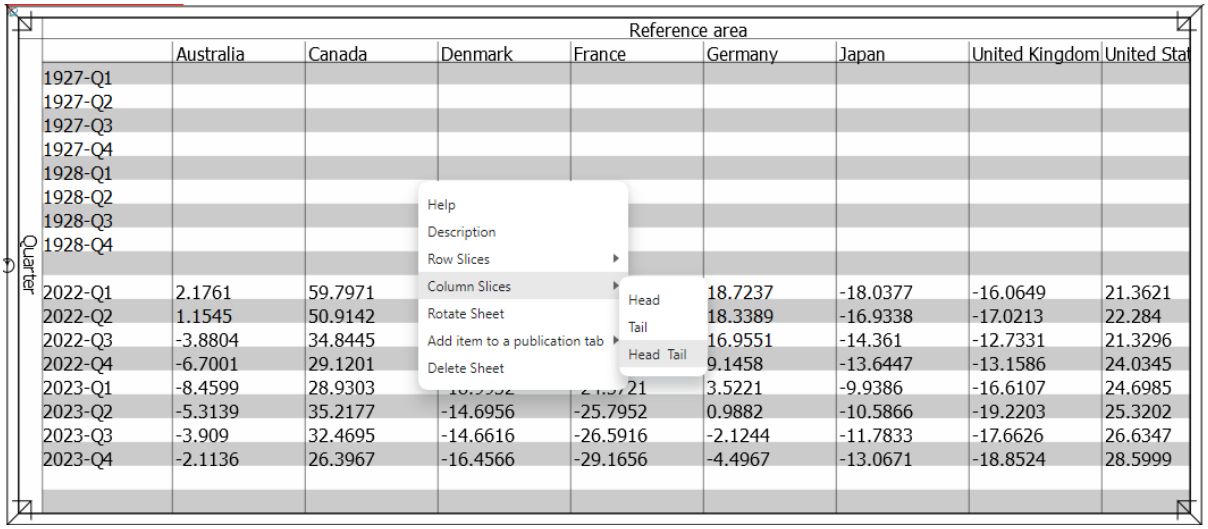


Figure 17: Head and Tail selected for Rows and being applied to columns

To show the last few rows of Date data, rotate the Date Axis to where Reference Area is (which will put the dates in the rows and countries in the columns). Then right-click on the Sheet and choose Row Slices/Tail. The sheet will then show the first countries in the data file, and the last quarters.

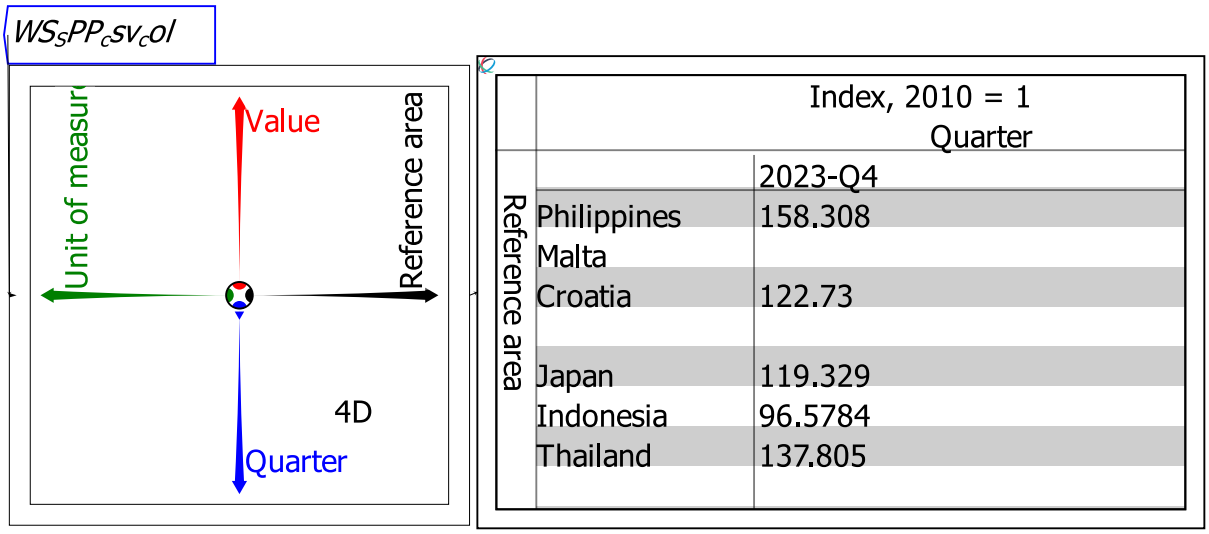


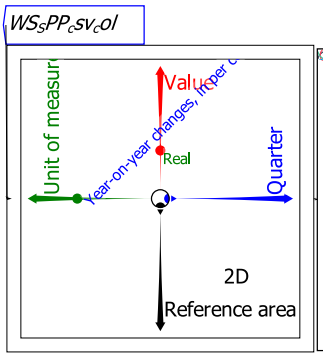
Figure 18: Output from a Ravel after axis rotation and selecting Head-Tail to display sheet contents

The data still shows the Nominal Index data, since these are the first entries in the other two axes. You can control the entry shown using the selector dots on those two axes: these are the coloured dots that are currently within the inner circle of the Ravel. This is called slicing the data.

Selector dots can be moved:

- By the mouse. Click on a dot and drag it to the required selection; or
- By the arrow keys. Use the mouse to move the cursor so that it is hovering over an axis; then use the up (or right) arrow key to move the dot out towards the arrowhead on an axis, or the down (or left) arrow key to move back towards the center.

To see the Real (CPI-adjusted) annual rate of change of house prices, use the selector dot on those two axes. That selection is shown below—where Date has also been rotated to the rows so that Countries are shown by the columns.



Quarter	Reference area				
	Advanced econo	Australia	Austria	Belgium	Brazil
2022-Q1	7.5261	14.7438	6.1627	-1.6331	-5.4766
2022-Q2	4.6602	6.7951	4.8801	-2.7945	-7.8309
2022-Q3	0.5382	-2.8586	0.9063	-4.4214	-6.1292
2022-Q4	-2.757	-10.114	-4.8376	-5.822	-4.4648
2023-Q1	-4.8682	-12.2902	-8.4897	-3.1892	-3.9672
2023-Q2	-4.6906	-9.2779	-10.2016	-2.9003	-2.318
2023-Q3	-2.7614	-2.8638	-9.0924	-2.5596	-2.7597
2023-Q4	-0.8091	2.7659	-7.3811	2.5228	-2.54

Figure 19: Output from a Ravel after slicing the data using selector dots

Calipers

You can select a contiguous range of data on an axis by inserting a caliper, using the "Toggle axis calipers" item on the context menu for an axis. Calipers initially cover the entire range of the data—see Figure 20.

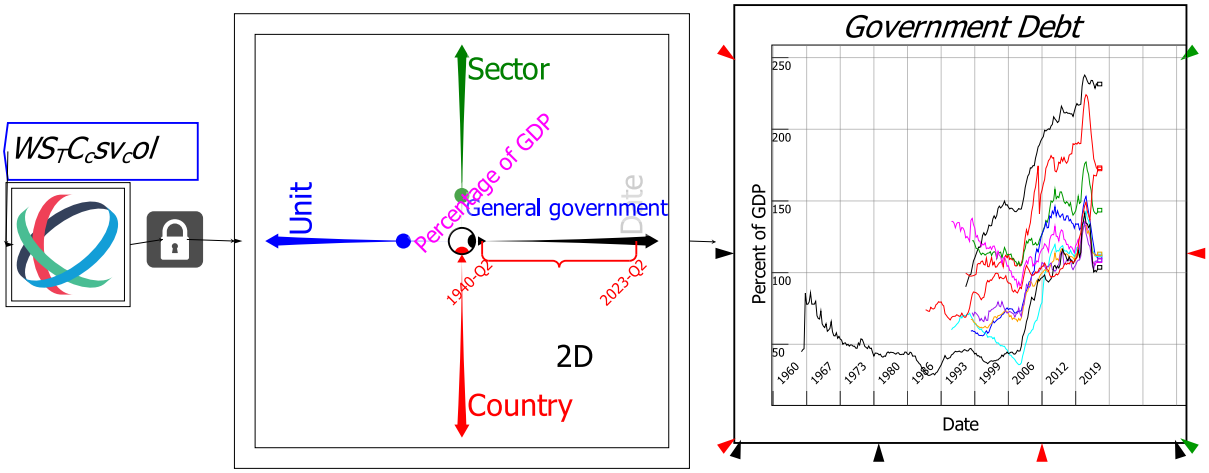


Figure 20: Calipers applied to a Date axis but not yet calibrated

You then move the ends of its parenthesis to specify where you want the excerpt to begin and end—see Figure 21. If you click on the tip of the parenthesis, you move the entire selection, while maintaining the number of elements it selects on the axis.

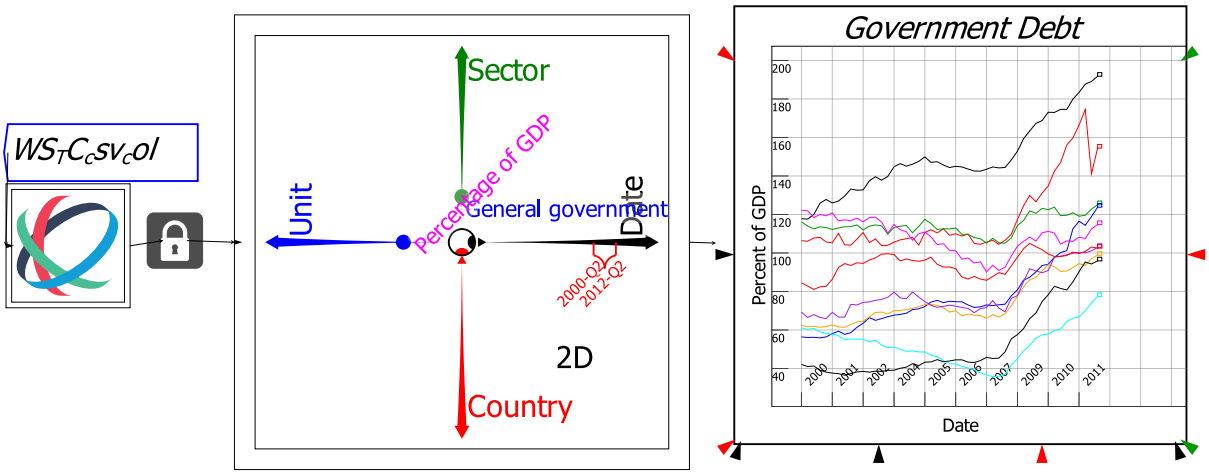


Figure 21: Calipers applied and calibrated to data between 2000 and 2012

Future releases of *Ravel* will support multiple named calipers on an axis, which will enable non-contiguous sets of data to be displayed on the one Sheet or Plot.

Reducing dimensions by Reductions or Rollups

Often you will wish to work with only a summarized form of a dimension. That is done by "collapsing" or "rolling up" an axis towards the centre of the Ravel.

Click on the axis you wish to roll-up, and drag its arrowhead towards the centre of the Ravel. This collapses the axis and aggregates the data according to the current setting of "Set Next Aggregation" on the right-click context menu. Six forms of aggregation are supported: Sum, Product, average, standard deviation, minimum and maximum. Once you have selected one of these, it will be applied to the next axis that is rolled up.

Figure 23 shows a roll-up performed on the Reference Area axis, where the aggregation method chosen is the average, "Pick Slices" has also been used to select just European countries.

A limited range of dates have been chosen using a caliper—using the context-menu choice "Toggle axis calipers". The caliper has been applied to the Date axis to select all months from

1950 till the latest data in 2024 (later versions of Ravel will support multiple calipers on an axis, to enable non-contiguous data ranges to be selected when desired).

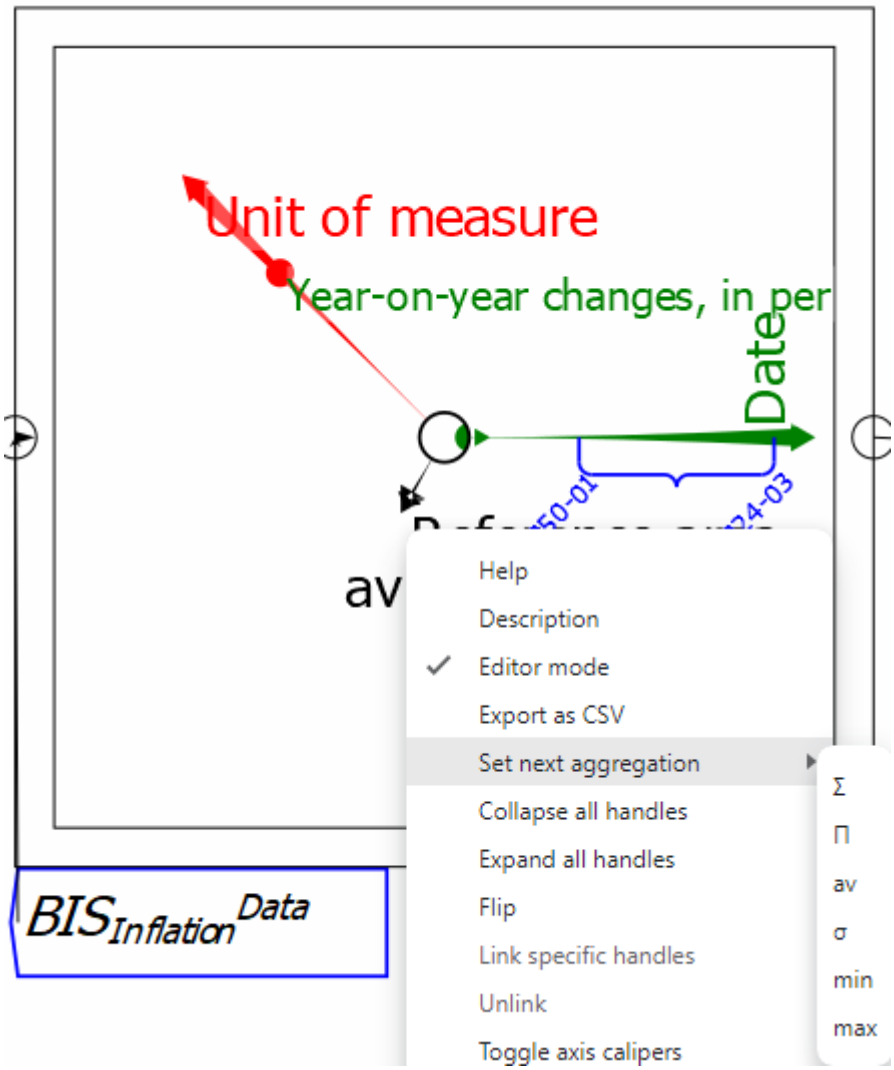


Figure 22: Setting the next aggregation command on a roll-up

With these operations performed, the Ravel then outputs the average inflation rate for Europe for the last 75 years.

Figure 23: The right-click/context menu for setting the next aggregation method.

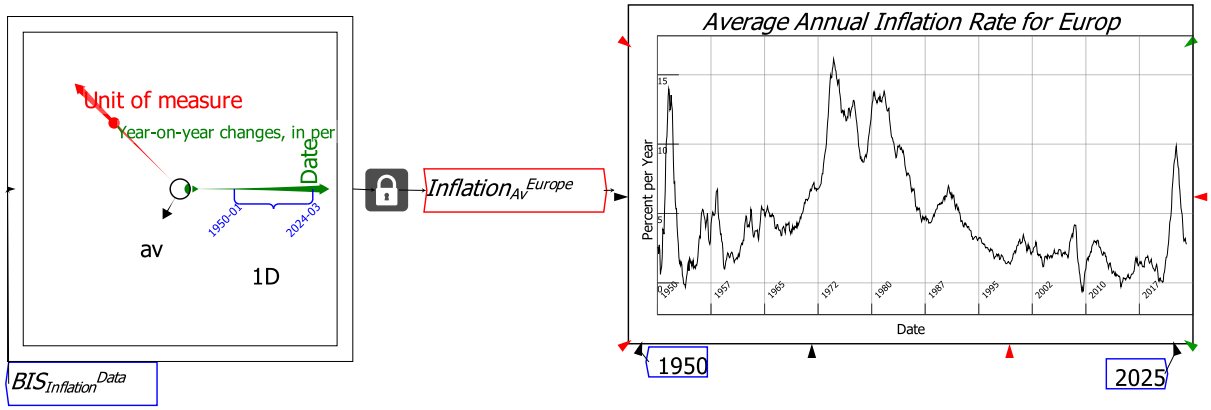


Figure 24: Collapsing ("Rolling Up") a Ravel dimension to summarise data

You can do a lot to analyse data just by exploiting the features of the Ravel, but the real power of *Ravel* the program comes when you combine Ravels for data selection with flowchart formulas for analysis. This necessitates using named variables, and here *Ravel* is much more flexible than Spreadsheets or BI programs.

Naming Variables

Variables should have meaningful names, so that it's easy for yourself and others to understand your analysis. *Ravel* uses the LaTeX text formatting language to enable users to create very informative variable names: English-language names can be augmented by superscripts and subscripts, and you can even use Greek letters.

A variable or parameter name can just be straight text (including spaces), but superscripts and subscripts, used wisely, can make the variables much easier to read.

- To subscript a single character, precede it with an underscore "_" character;
- To superscript it, precede it with a caret "^" character;
- To subscript or superscript a string of text, enclose the text inside curly brackets (parentheses): {}; and
- To write a Greek character, type the backslash key "\" followed by the English word for the Greek character.

Ravel Output		English entry	Ravel Input
BIS_{HPI}^{Data}		BISHPIData	BIS_{HPI}^{Data}
$\Delta Credit_{HH}^{\%GDP}$		DCreditHH%GDP	\Delta{Credit}_{HH}^{%GDP}
Α	α	Alpha	\Alpha, \alpha
Β	β	Beta	\Beta, \beta
Γ	γ	Gamma	\Gamma, \gamma
Δ	δ	Delta	\Delta, \delta
Ε	ε	Epsilon	\Epsilon, \epsilon
Ζ	ζ	Zeta	\Zeta, \zeta
Η	η	Eta	\Eta, \eta
Θ	θ	Theta	\Theta, \theta
Ι	ι	Iota	\Iota, \iota
Κ	κ	Kappa	\Kappa, \kappa
Λ	λ	Lambda	\Lambda, \lambda
Μ	μ	Mu	\Mu, \mu
Ν	ν	Nu	\Nu, \nu
Ξ	ξ	Xi	\Xi, \xi
Ο	ο	Omicron	\Omicron, \omicron
Π	π	Pi	\Pi, \pi
Ρ	ρ	Rho	\Rho, rho
Σ	σ	Sigma	\Sigma, \sigma
Τ	τ	Tau	\Tau, \tau
Υ	υ	Upsilon	\Upsilon, \upsilon
Φ	φ	Phi	\Phi, \phi
Χ	χ	Chi	\Chi, \chi
Ψ	ψ	Psi	\Psi, \psi
Ω	ω	Omega	\Omega, \omega

Figure 25: Table of Greek characters supported by Ravel

When you combine characters, use the {} to differentiate, for example, Greek letters from English. For example, Figure 26 has the variable ΔHPI_{Real}, and that is created by typing {\Delta}HPI_{Real}.

You can enter variable names either by clicking on the Var widget and then choosing to enter a Variable or Parameter; or by simply typing anywhere on the wiring canvas. The latter action will bring up a text entry window, and when you press Enter (or click on OK) the variable and parameter definition window will pop up. Press Enter or click OK on that form—after adding additional details if desired, such as a Short or Long Description: the Short description functions as a "Tool Tip" which you'll see when the mouse hovers over an entity, and it's also used for Plot Legends—and the Variable or Parameter will be attached to the mouse cursor. Click where you want to place it on the canvas.

Attaching Locks and Variables to Ravels

The *BIS* house price data file combines information on House Price Indices (where the base year is 2010, so all indices are 100 during 2010), and the annual rate of change of house prices, with data on both Nominal and Real (CPI-deflated) prices. To analyse the data, it is useful to separate it into *House Price Index* information and *House Price Inflation* information, and to focus on Real rather than Nominal Prices. That is done by attaching the output of the Ravel to Locks, and the Locks to named Variables that you create.

Figure 26 shows two variables HPI_{Real} and ΔHPI_{Real}. They are joined to the Ravel via locks, and once a lock is closed, the output from that lock remains the same, even if the selection on the source Ravel is altered.

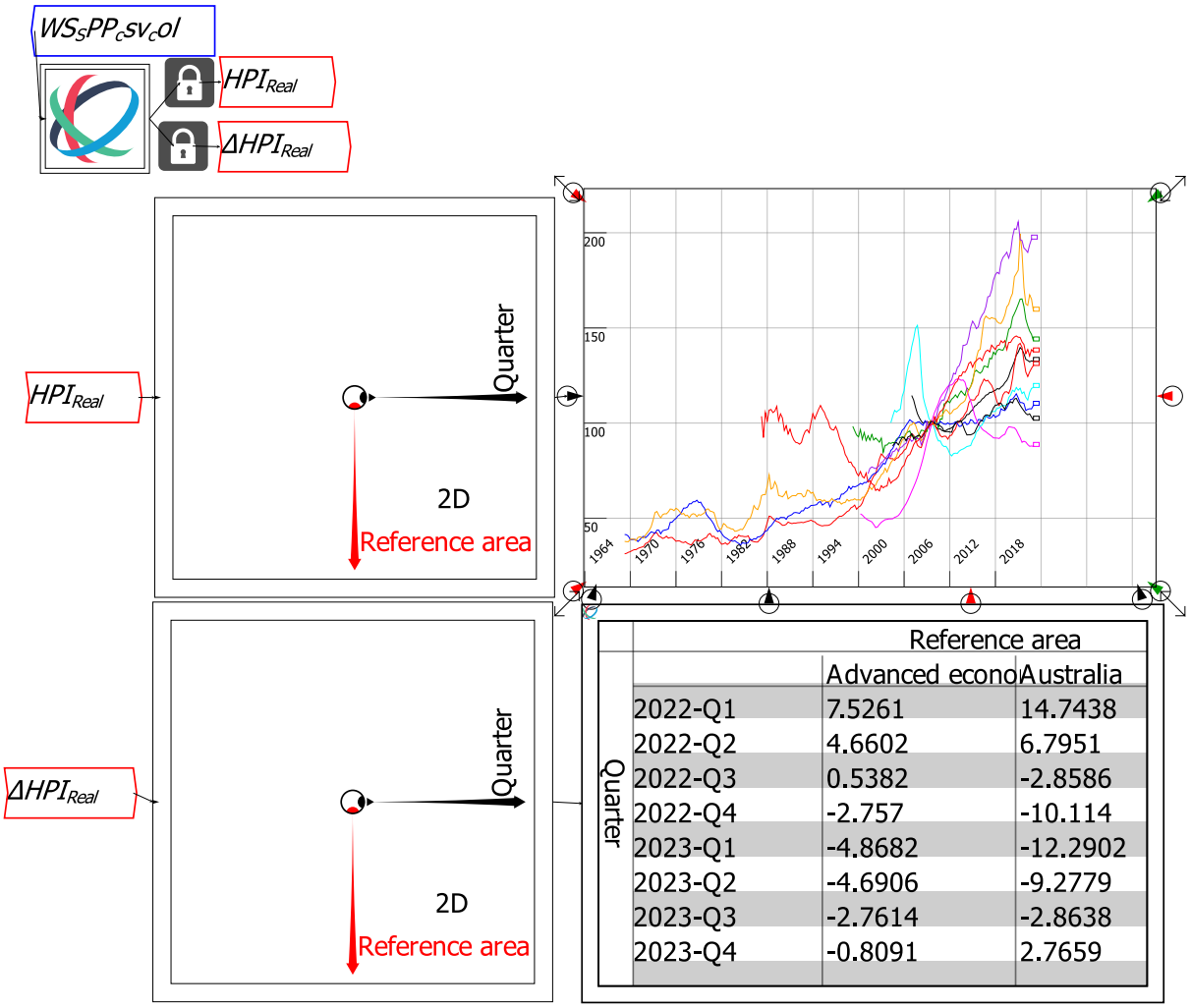


Figure 26: Using Locks to assign output from a Ravel to variables

With the data separated into index and inflation data, we can now focus on those subsets of the data rather than the entire source file. The house price inflation data in Figure 26 suggests a possibly useful piece of analysis: on average, house prices in Advanced economies fell during 2023. To see whether this was a common phenomenon or isolated to a few countries, why not compare the average for all advanced countries (the first entry on the Reference Area axis) to the data for each advanced country?

Basic Analysis using *Ravel*

Figure 29 shows the result of doing this. Firstly, select just the data for "Advanced Economies" using the selector dot, and attach that to a new variable $\Delta HPI_{Advanced}^{Avg}$.

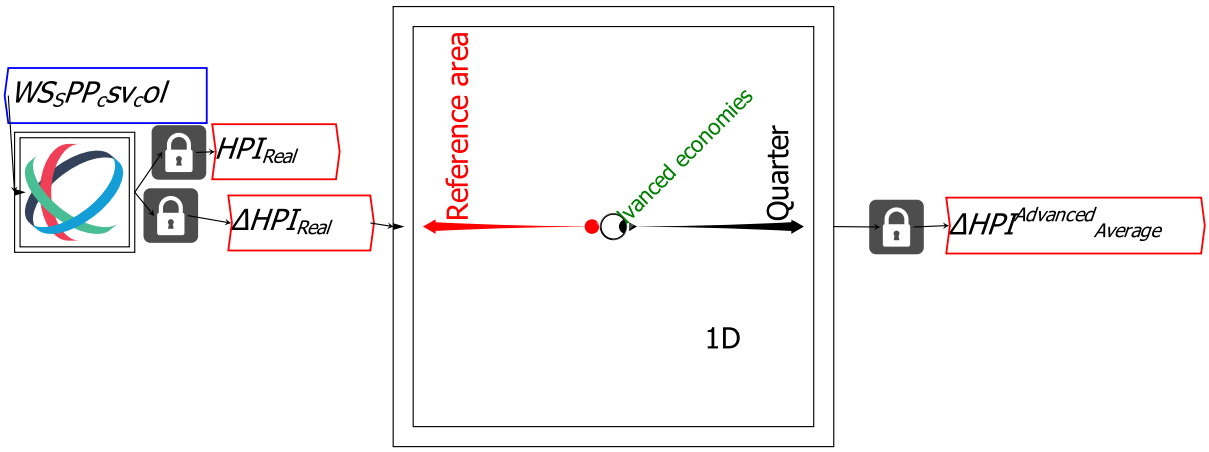


Figure 27: Creating a new variable containing just the data for "Advanced economies"

Then select a number of advanced economies from the axis (Australia, Canada, Denmark, etc.) using the right-click menu option “Pick axis slices”, and assigning that to another variable $\Delta HPI_{Advanced}$.

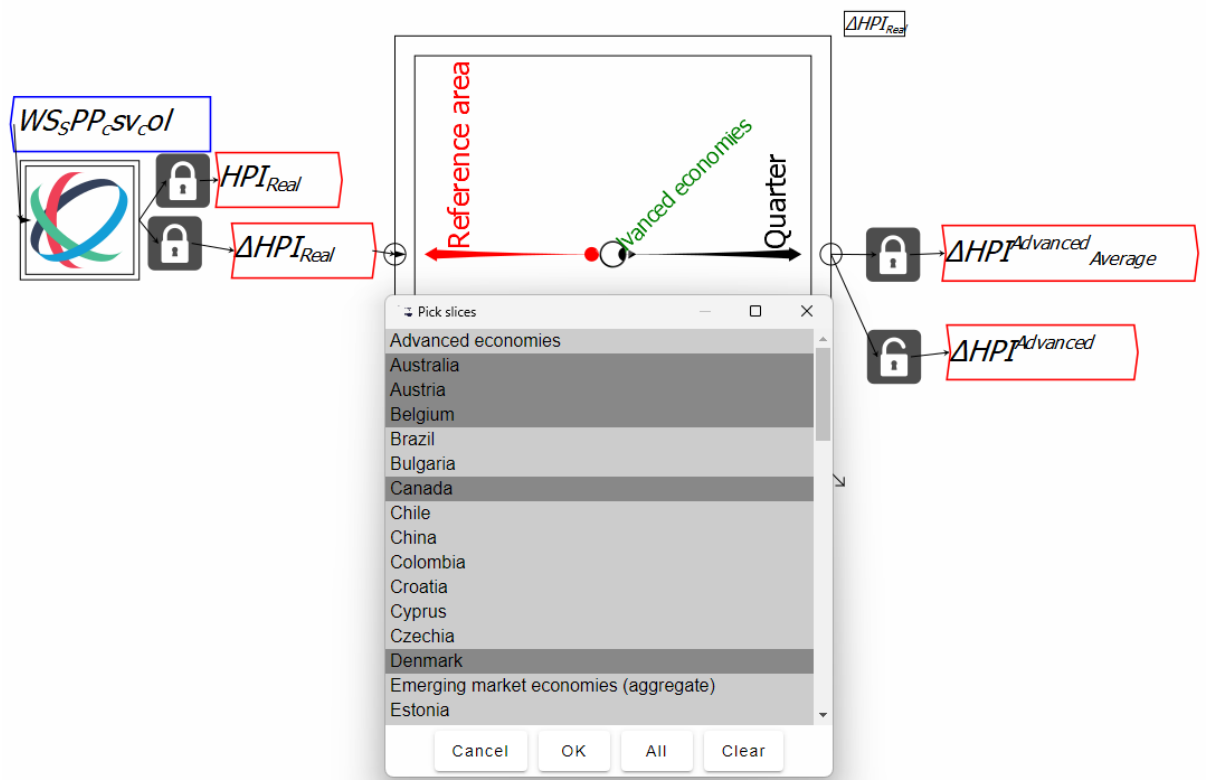


Figure 28: Selecting only advanced economies using the Pick Slices right-click context menu

This can then be subtracted from the $\Delta HPI_{Advanced}$ array to give you the information you were after: how much house price inflation in each advanced economy differed from the average for all advanced economies. Attach another Ravel after the calculation, select a specific date, and choose “Sort by Value” from the context menu for the Reference Area axis.

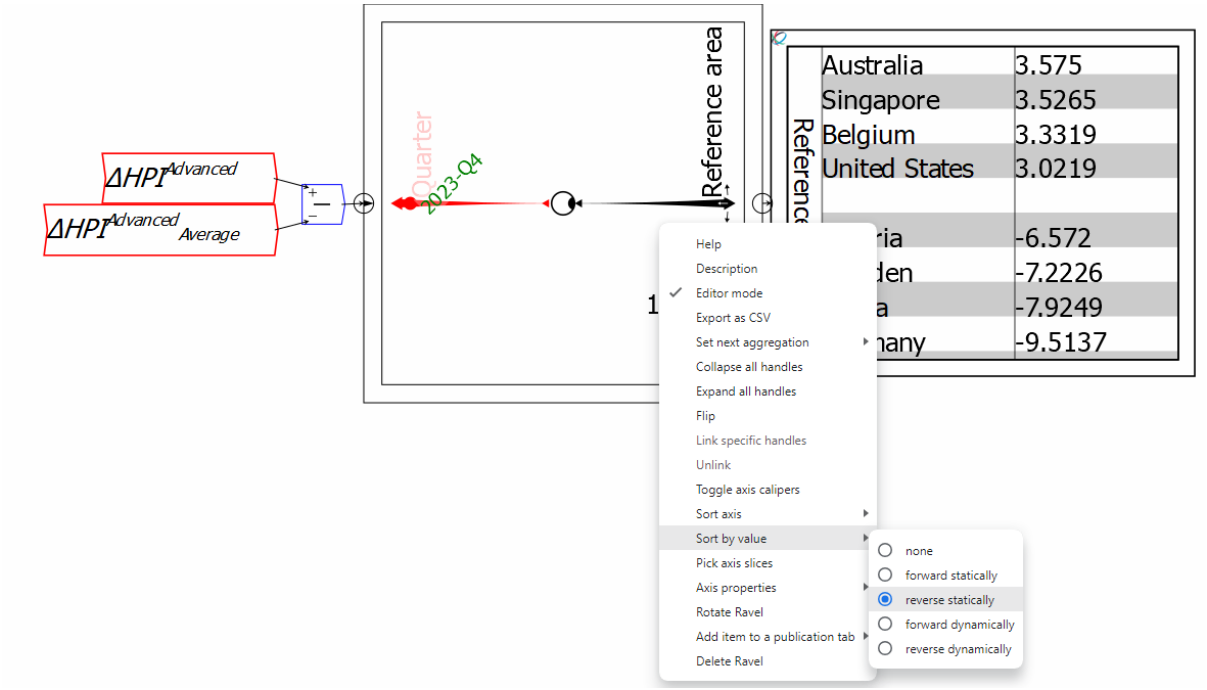


Figure 29: A simple calculation using Ravel's flowchart mathematical operators

This now returns the countries in reverse order of the difference between their national house price inflation rates and the average for all advanced economies.

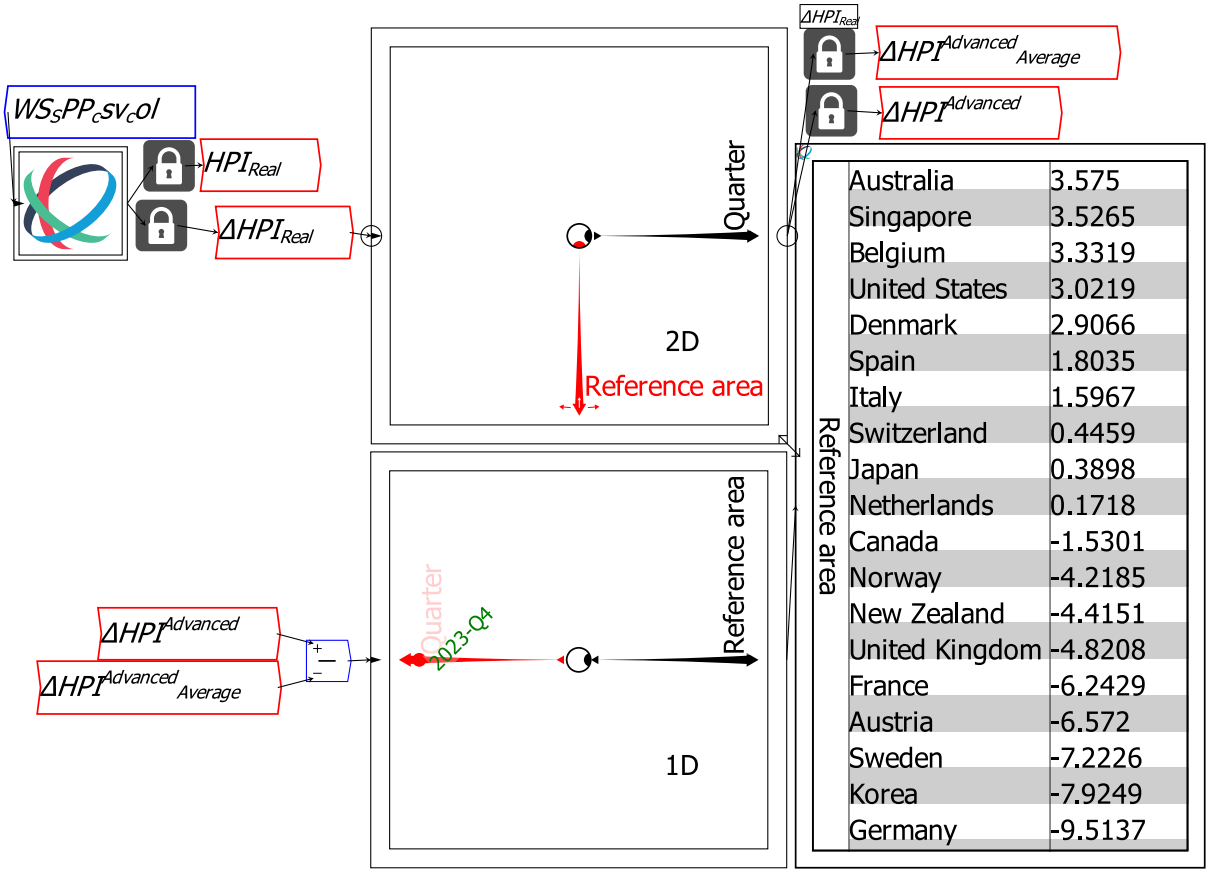


Figure 30: Heavy price falls in France to Germany pulled the average down

You could do the same operation with a spreadsheet, but it would be cumbersome and difficult to document. The initial spreadsheet formula would be something like:

C2-$B2

Column B would contain the advanced economies average data, and columns C to about N would contain the data for each advanced economy. The dollar sign is there to ensure that when you replicate this formula across the array of 388 quarters and about 15 countries, the column subtracted is always that for Advanced economies.

One formula in *Ravel* thus takes the place of about 6,000 replicated cell formulas. Unlike spreadsheets, the formulas in *Ravel* are visible, self-documenting, and easy to read.

Other features of a Ravel

Figure 31 shows a Ravel attached to a Sheet and a Plot, with selections made on two of the Ravel's 4 axes, so that the output is 2-dimensional: Country by Date, with Date being the x-axis for the Plot and the row entries for the Sheet.

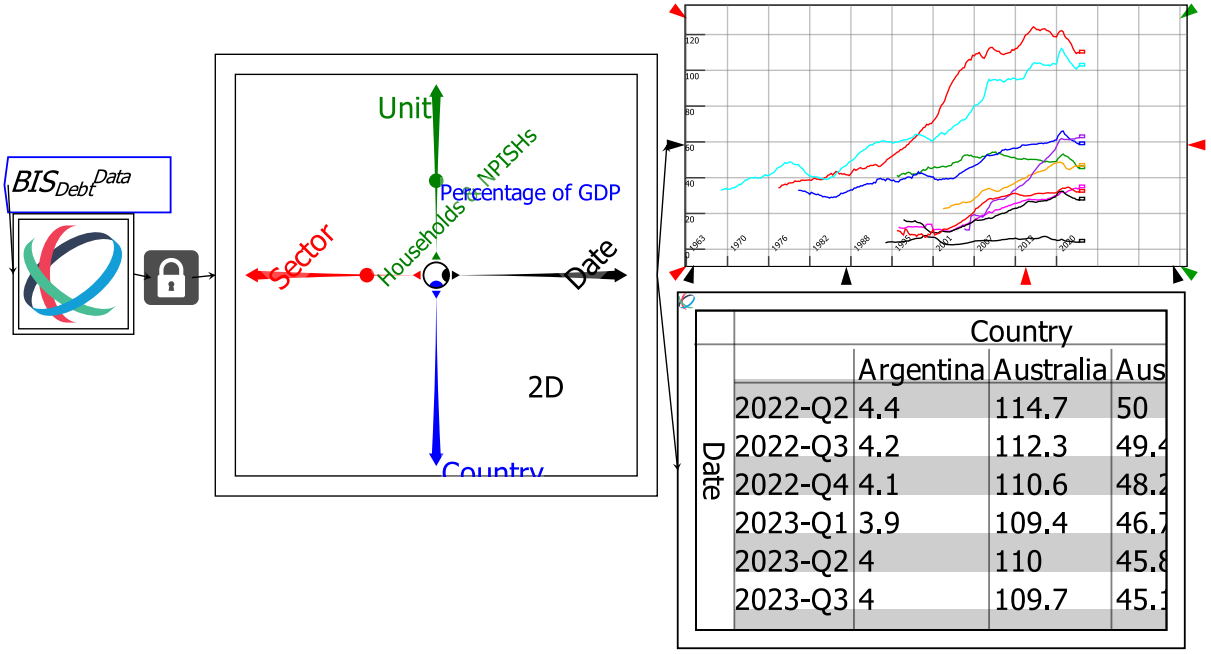


Figure 31: Rotating data by flipping Ravel axes

If you rotate the Country axis to the right, Country becomes the X-axis on the Plot and the column entry on the Sheet.

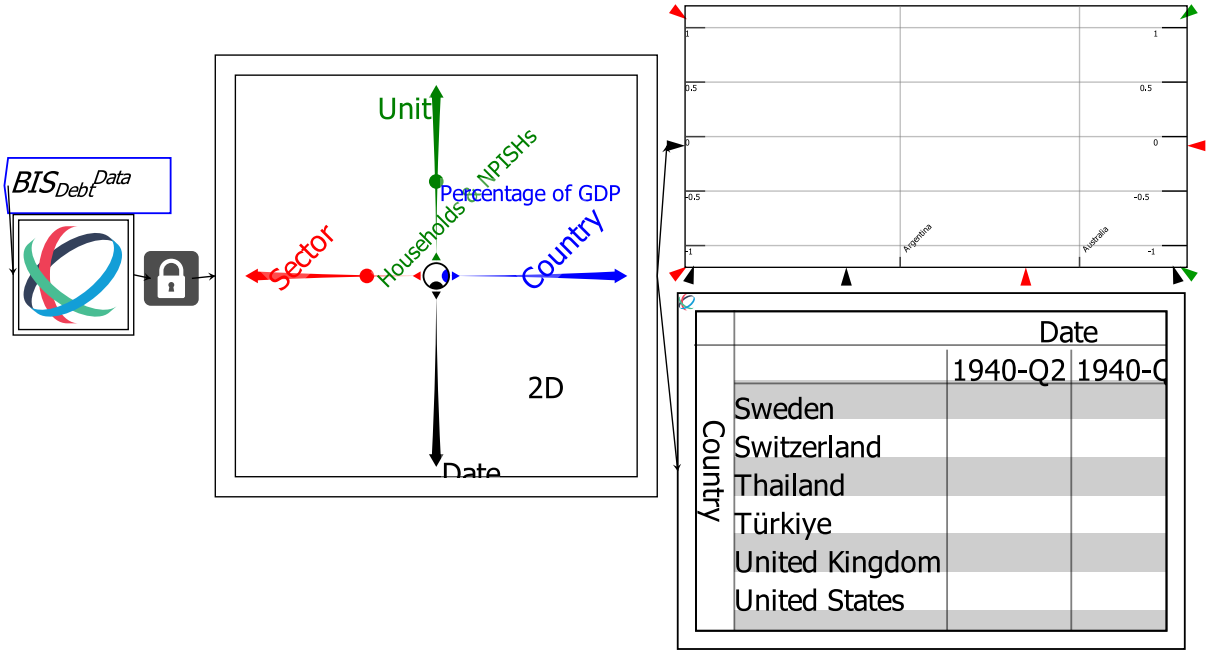


Figure 32: Rotating data by flipping Ravel axes

That's not very informative initially, because most of the entries in the data are blank for early dates. Once you select a quarter with data, there could be up to 43 bars on the horizontal axis (one per country or region) which is very cumbersome.

To reduce the number of countries displayed, right-click on the Country axis and choose "Pick axis slices". This brings up the menu shown in Figure 33, where you can pick an individual country by clicking on its name, or select several by control-clicking on the names. You can turn a selection on and off the same way.

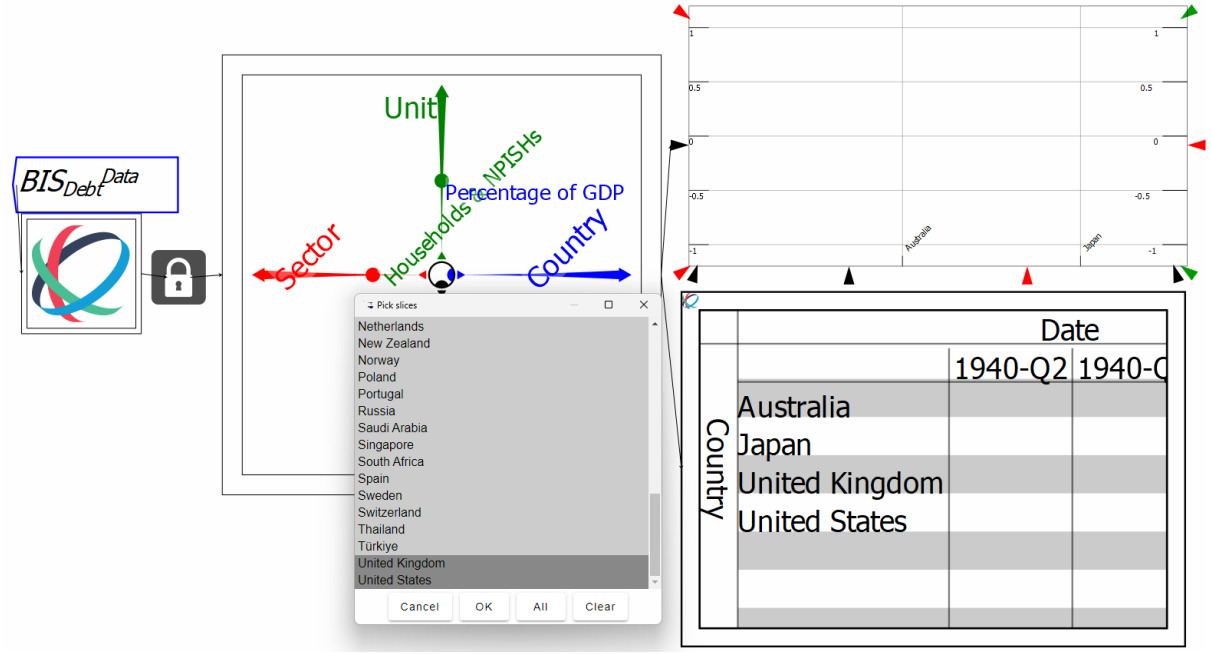


Figure 33: Selecting data points on a Ravel Axis

There's no data at the beginning of the Date axis, but if you now move the selector dot on the Date axis—either by grabbing the selector button with the mouse, or using the up or left arrow

keys to move forward; control-arrow key can also be used, which forces an immediate recalculation of the Plot and Sheet—you can produce a bar chart of the data at that point in time.

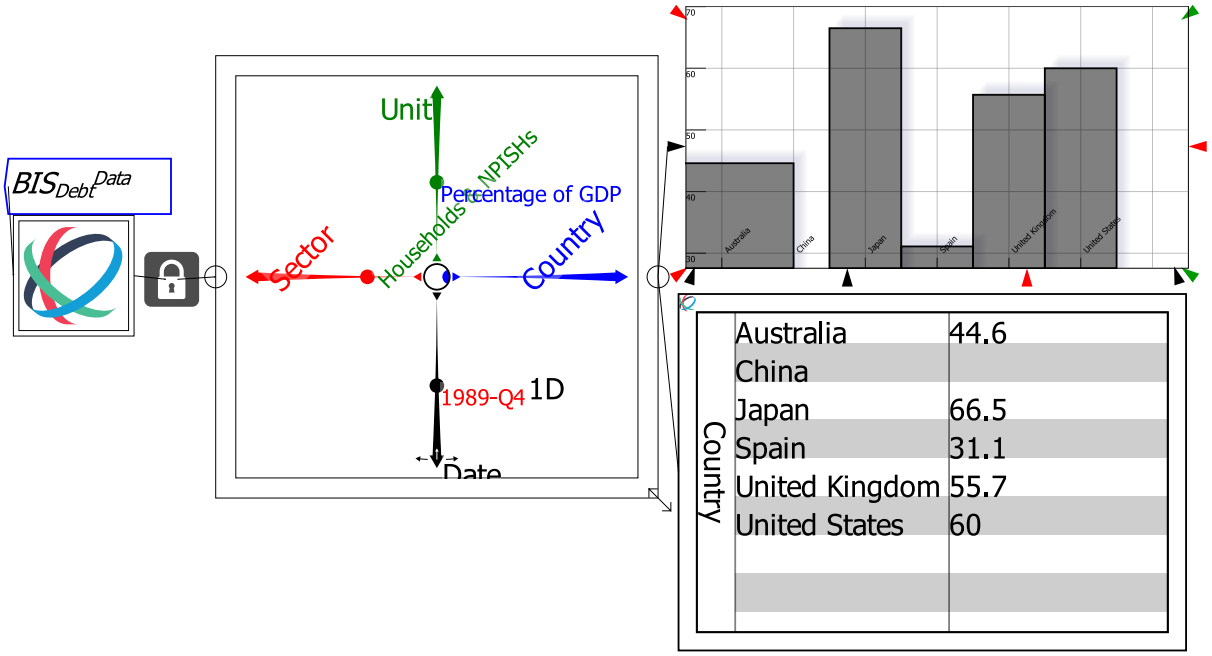


Figure 34: Selecting data points on a Ravel Axis

Ravel axes can perform statistical operations as well, using the rollup function as explained earlier, the “Set next aggregation” command.

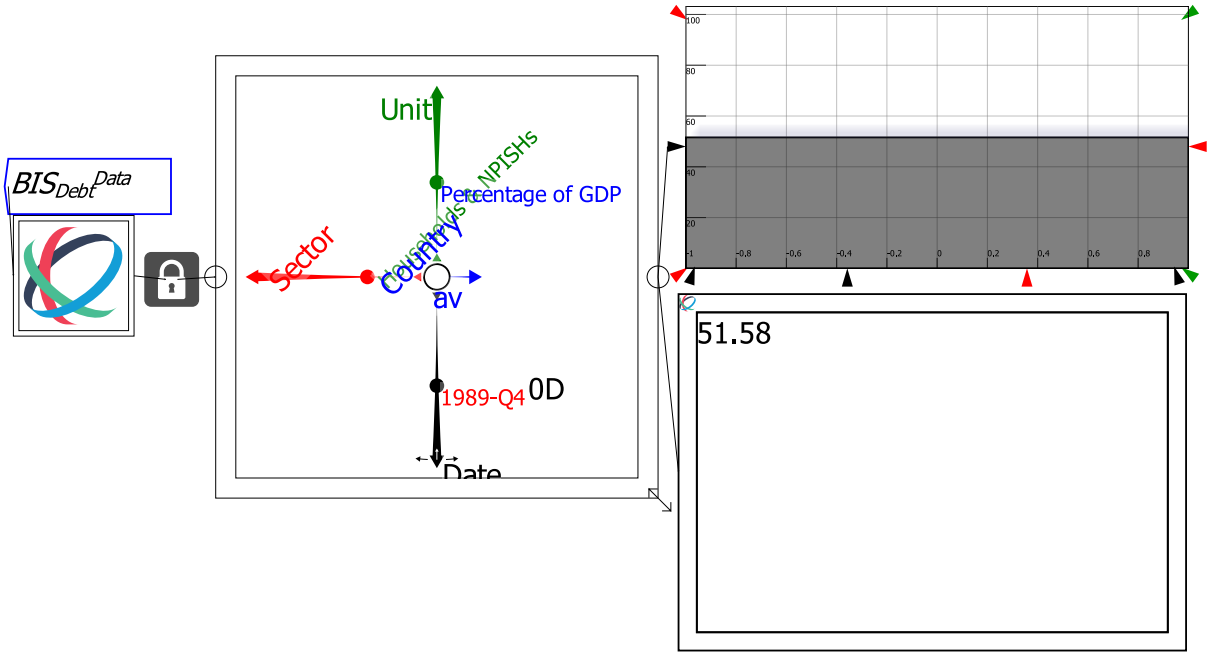


Figure 35: Calculating averages etc. using Ravel’s axes

This feature is best combined with the Lock, so that you can create a time series—in this case, the average level of household debt by country over time—and use it in later analysis:

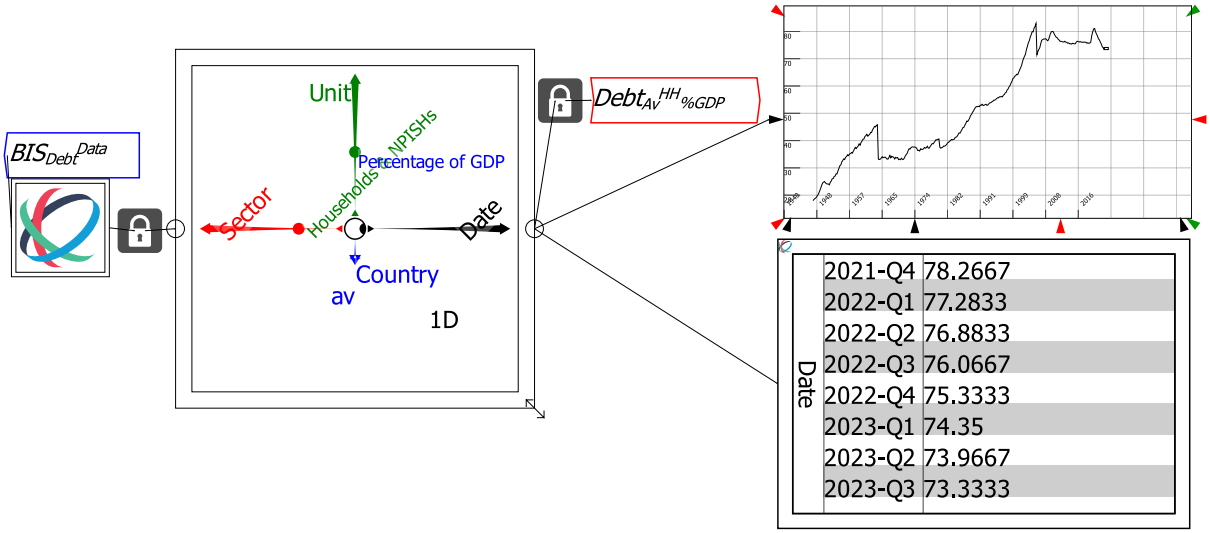


Figure 36: Calculating averages etc. using Ravel's axes

Ravel's mathematical operators

Ravel has a large number of mathematical operators which are accessed via the widget bar. The top-level menu items are shown in Figure 37.

The menus are respectively:

- Binary operators that take 2 (or more) inputs;
- Fundamental constants: e, π, 0, 1, ∝, and the percentage operator %;
- Unary operators with only one input;
- Array operators—sums and products of arrays and array element identifiers;
- Running sum and product and difference operators;
- Statistical functions (mean, median, standard deviation, etc.); and
- Mathematical operators for handling multi-dimensional data: dot, inner and outer products, etc.

Figure 37: The mathematical operators menu bar

Click on one of these menu widgets, and the relevant selection of operators will be displayed above the bar. Click on one of these operators, and it will be attached to the mouse cursor for you to place on the canvas. The full complement of operators, aligned with the menu which invokes them, is shown in Figure 38.

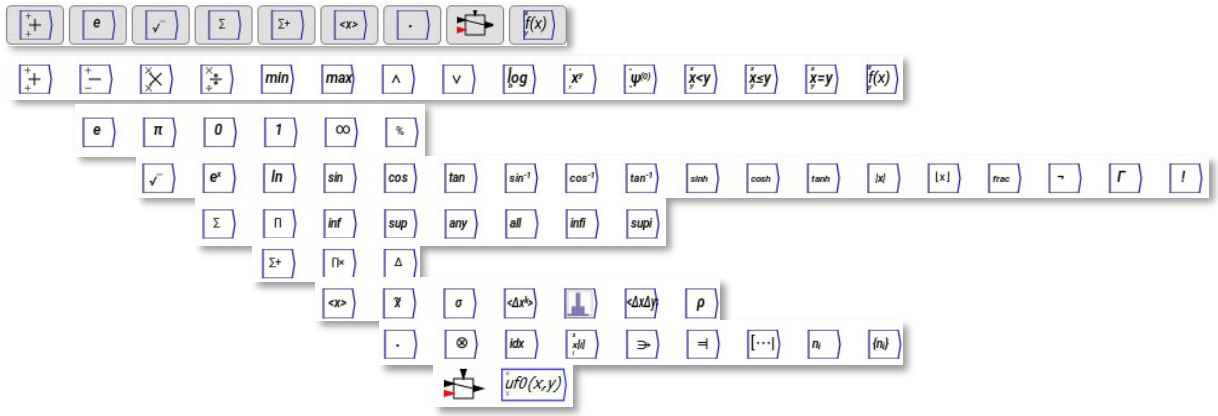


Figure 38: Ravel's full complement of mathematical functions

Full details on each of these operators is given via in-context help: hover over an operator, bring up the right-click/context menu, and choose Help.

Operators can be entered onto the canvas by either clicking on the relevant menu and then operator, or by typing its shortcut on the canvas.

Table 1: The binary menu 

Operator	Operation	Shortcut
	Add two or more variables together	+
	Subtract two or more variables. Inputs to the upper (+) and lower (-) inputs are first added together, then the sum of the input to (-) are subtracted from the sum of the inputs to (+)	- <Enter>
	Multiply two or more variables together.	*
	Divide one or more variables by others. Inputs to the upper (x) and lower (÷) inputs are first multiplied together, then the product of the input to (x) are divided by the product of the inputs to (÷)	/
	Return the minimum of two inputs	min
	Return the maximum of two inputs	max
	Return 1 if two inputs are both true (equal to or greater than 1), else return 0	and
	Return 1 if one input is true (equal to or greater than 1), else return 0	or
	Return the log of the inputs to base a.	log
	Return x to the power of y	pow
	Polygamma function. For mathematicians and a component of the Minsky simulation engine	
	Return 1 if x<y else return 0	le

x x<y y	Return 1 if x≤y else return 0	leq
x x=y y	Return 1 if x=y else return 0	eq
x uf0(x,y) y	Return the user-defined function with inputs x and y	

Table 2: The fundamental constants menu e

Operator	Operation	Shortcut
e	Parameter equal to Euler's e ≅2.718281828459…	euler
π	Parameter equal to π ≅3.1415926535897932…	pi
0	Parameter equal to 0	zero
1	Parameter equal to 1	one
∞	Parameter for Infinity—generates 1.79769 multiplied by 10^{308}	inf
%	Takes an input variable and multiplies each element in it by 100	%

Table 3: The unary menu √

Operator	Operation	Shortcut
√	Return the square root of the inputted variable	sqrt
e^x	Return Euler's e raised to the power of the inputted variable x	exp
ln	Return the natural logarithm of the inputted variable	ln
sin	Return the sine of the inputted variable	sin
cos	Return the cosine of the inputted variable	cos
tan	Return the tan of the inputted variable	tan
$\sin^{-1}$	Return the inverse sine of the inputted variable	
$\cos^{-1}$	Return the inverse cosine of the inputted variable	
$\tan^{-1}$	Return the inverse tan of the inputted variable	
sinh	Return the hyperbolic sine of the inputted variable	
cosh	Return the hyperbolic cosine of the inputted variable	
tanh	Return the hyperbolic tan of the inputted variable	
|x|	Return the magnitude of the inputted variable	

⌊x⌋	Return the integers of the inputted variable	
frac	Return the inputted variable minus the integer parts	
¬	The Not operator. Return 0 if the input is <0.5 otherwise return 1	
Γ	Return the Gamma function of the input (for mathematicians)	
!	Return the factorial of the input	

Table 4: The reduction menu Σ

Operator	Operation	Shortcut
Σ	Sum the inputs	
∏	Multiply the inputs	
inf	Infimum: return the smallest value along an axis	
sup	Supremum: return the biggest value along an axis	
any	Return 1 if any value along an axis is non-zero, otherwise return 0	
all	Return 1 if all values along a given axis are nonzero, otherwise return 0 if any are zero.	
infi	Infindex. Return the index (the position in an array) of the smallest value along an axis	
supi	Supindex. Return the index (the position in an array) of the greatest value along a given axis.	

Table 5: The scan menu Σ+

Operator	Operation	Shortcut
Σ+	Running sum. Return the running sum of the input along a given axis	
∏×	The running product of the input tensor along a given axis	
Δ⁻	Prior difference. Calculate the change in values along an axis looking back ***n*** steps (the number of steps ***n*** prior is entered in the edit form). The first ***n*** entries on the axis are truncated.	
Δ⁺	Post difference. Calculate the change in values along an axis looking forward ***n*** steps (the number of steps ***n*** prior is entered in the edit form). The last ***n*** entries on the axis are truncated.	

Table 6: The statistics menu <x>

Operator	Operation	Shortcut
<x>	Return the mean (average) of the input	

$\tilde{x}$	Returns the median along a named dimension	
σ	Returns the standard deviation along a named dimension	
$<\Delta x^k>$	Returns the k-th moment about the mean along a named dimension.	
	Computes the histogram along a named dimension. The number of bins is set on the Edit menu	
$<\Delta x \Delta y>$	Computes the covariance of two tensors along a named dimension. This returns a single number when two one-dimensional variables are compared, otherwise it returns a matrix of the covariance patterns across the additional dimensions	
ρ	Returns the correlation coefficient of the inputs. This returns a single number when two one-dimensional variables are compared, otherwise it returns a matrix of the correlations patterns across the additional dimensions.	
	Linear regression. This takes two arguments—the dependent (top input) and independent variable (bottom input)—and returns a linear regression prediction for the dependent variable.	

Table 7: The tensor menu ▪

Operator	Operation	Shortcut
▪	Inner Product. Generates a single number by multiplying all the elements of one vector (or tensor) by another.	
⊗	Outer Product. Generates a matrix by the product of one vector with another	
idx	Index. Returns the values in an array which are true for a logical condition.	
$x_{[i]}^{x}$	Gather. Returns values from the top input (x, the variable from which values will be gathered), given the values in the bottom (i, the index)	
∋↠	Meld. Takes two or more variables with a shared dimension and combines their data on that dimension.	
⫤	Merge. Takes two or more variables with shared dimensions and combines them on a new dimension specified in the Edit field.	
[⋯|	Slice. Extracts a specified number of data points from an input variable (the top ten, etc.)	
n_i	Size. Returns the number of elements along a named dimension	
$\{n_i\}$	Shape. Returns a vector of axis sizes	

Analysing Related Data Using Shared Dimensions

The previous examples used a single imported data file. *Ravel* can import data from numerous sources into one file, which enables easy analysis of the relationships between the data, using their shared dimensions.

The next image shows two data files—on House Prices and Debt, both from the *Bank of International Settlements*—imported into one *Ravel* file. The two Ravels share the dimensions of

Date, *Country* and *Unit*. Six variables are defined from them, with self-explanatory names: $Debt_{HH}^{DC}$, for example, is household debt in domestic currency. All these variables inherit selected dimensions from the source data.

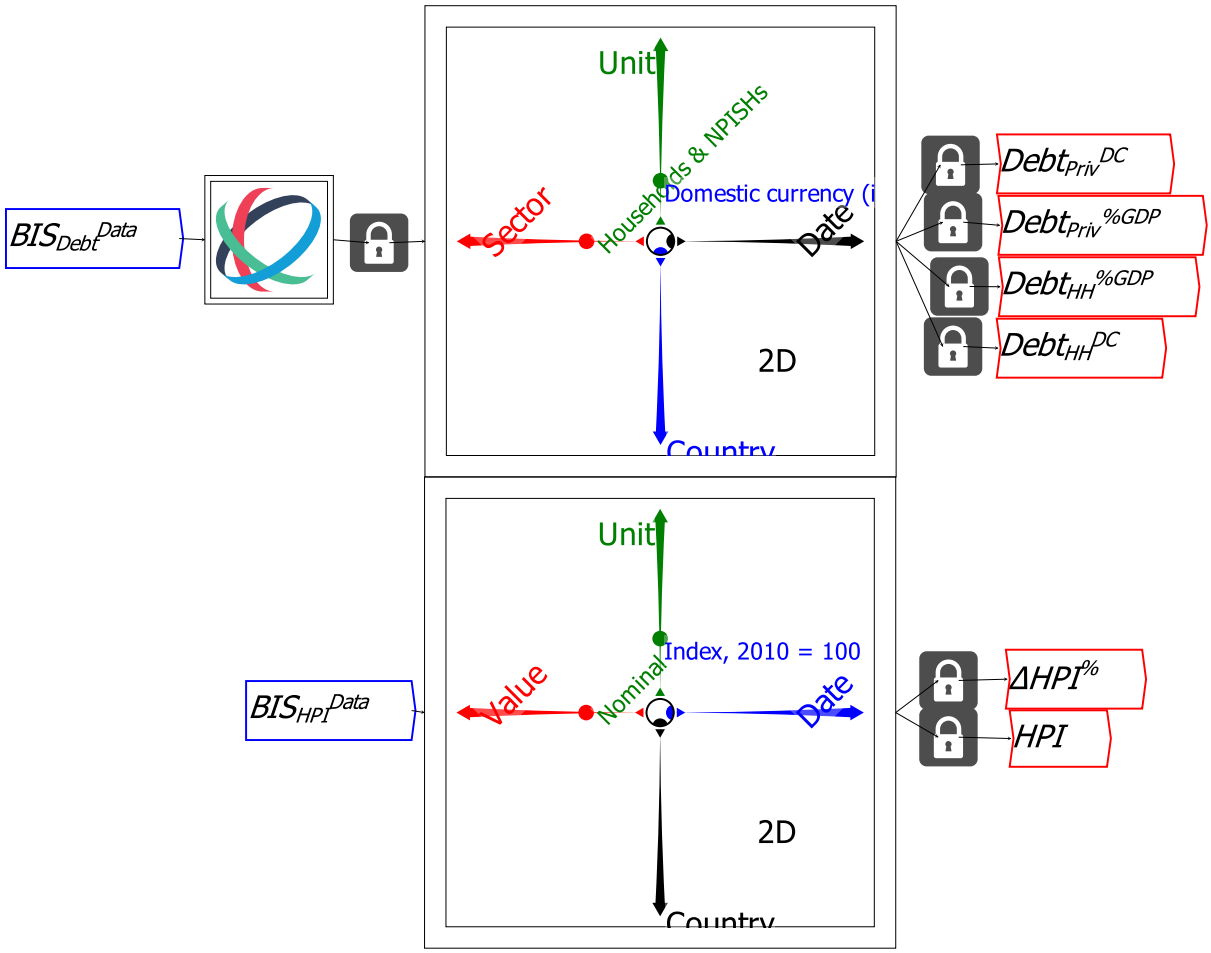


Figure 39: Allocating specific slices of data to variables

The entries on *Units* are different—Domestic currency, US$ and Percent of GDP for the debt data, Index of and change in house prices for the House Price data—but those on *Country* and *Date* are the same. This allows them—and any Ravels derived from them—to be linked, so that a selection of *Country* or *Date* made on one Ravel is also made on the other. This makes it straightforward to analyse relationships between data files.

$Debt_{Priv}^{DC}$ and $Debt_{Priv}^{\%GDP}$ are already linked, because they are derived from the same data file. When variables are derived from different data files—$Debt_{HH}^{\%GDP}$ is derived from the BIS file "WS_TC_csv_col.csv", while *HPI* data comes from "WS_SPP_csv_col.csv", we need to link the two source Ravels.

You do this by attach those variables to Ravels, selecting the two Ravels using click and drag—hold the left mouse button down on the canvas and then drag the mouse until the desired Ravels are selected (they will be greyed out).Then choose "Link Selected Ravels" from the right-click menu. If you do this without the mouse being over one of the Ravels, you will get the menu shown in Figure 40:

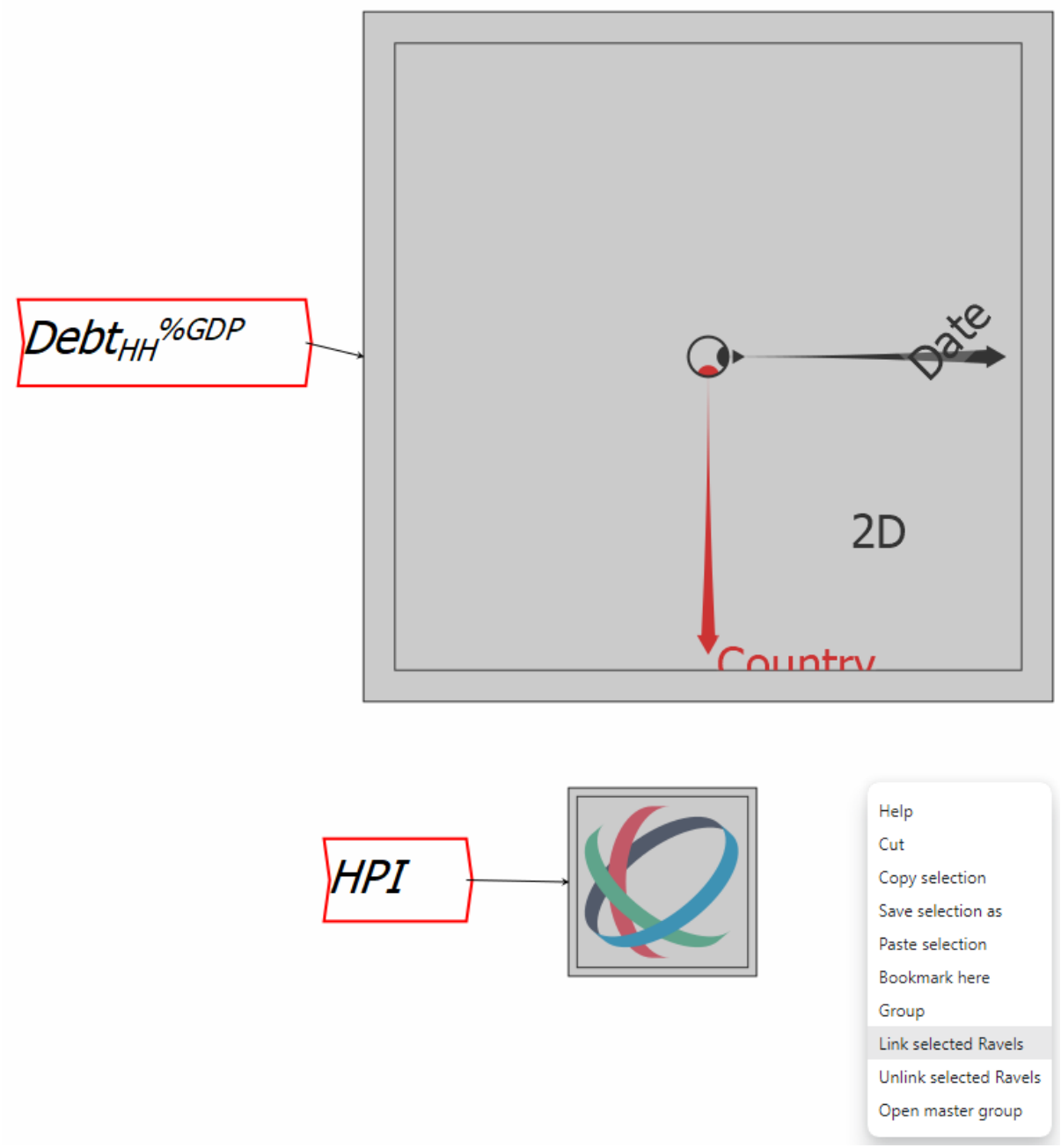


Figure 40: Linking two or more Ravels together

If your mouse is hovering over one of the Ravels, you will get the menu shown in Figure 41, where the menu item "Link specific handles" gives you a more detailed capacity to control which features of the Ravels are linked.

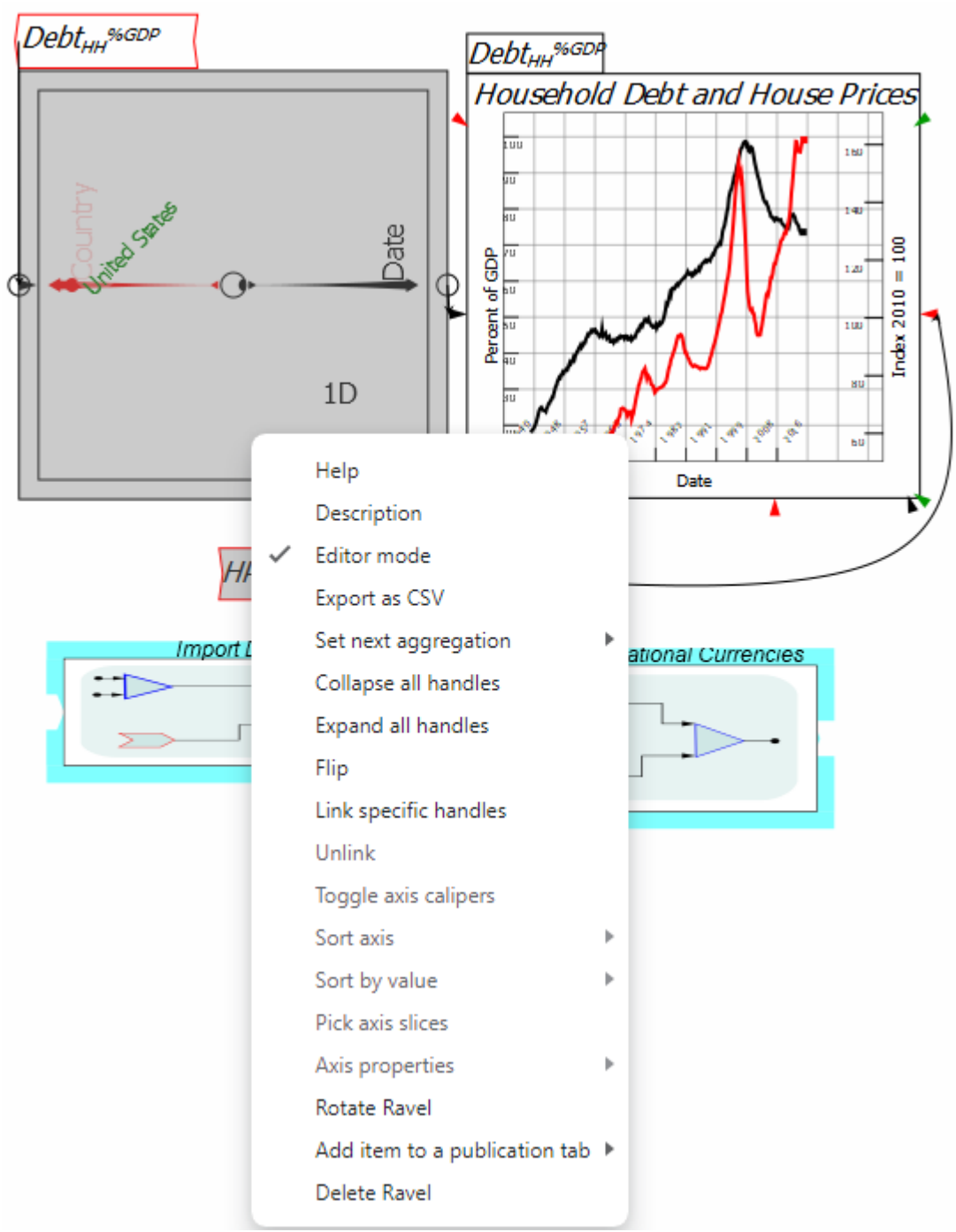


Figure 41: Linking two or more Ravels on specific characteristics

With these two Ravels linked, the household debt to GDP ratio can be plotted against the House Price Index, as shown below. Figure 42 shows household debt and house price data for the USA, but if the selector dot were moved to Spain, it would show the same two data series for Spain.

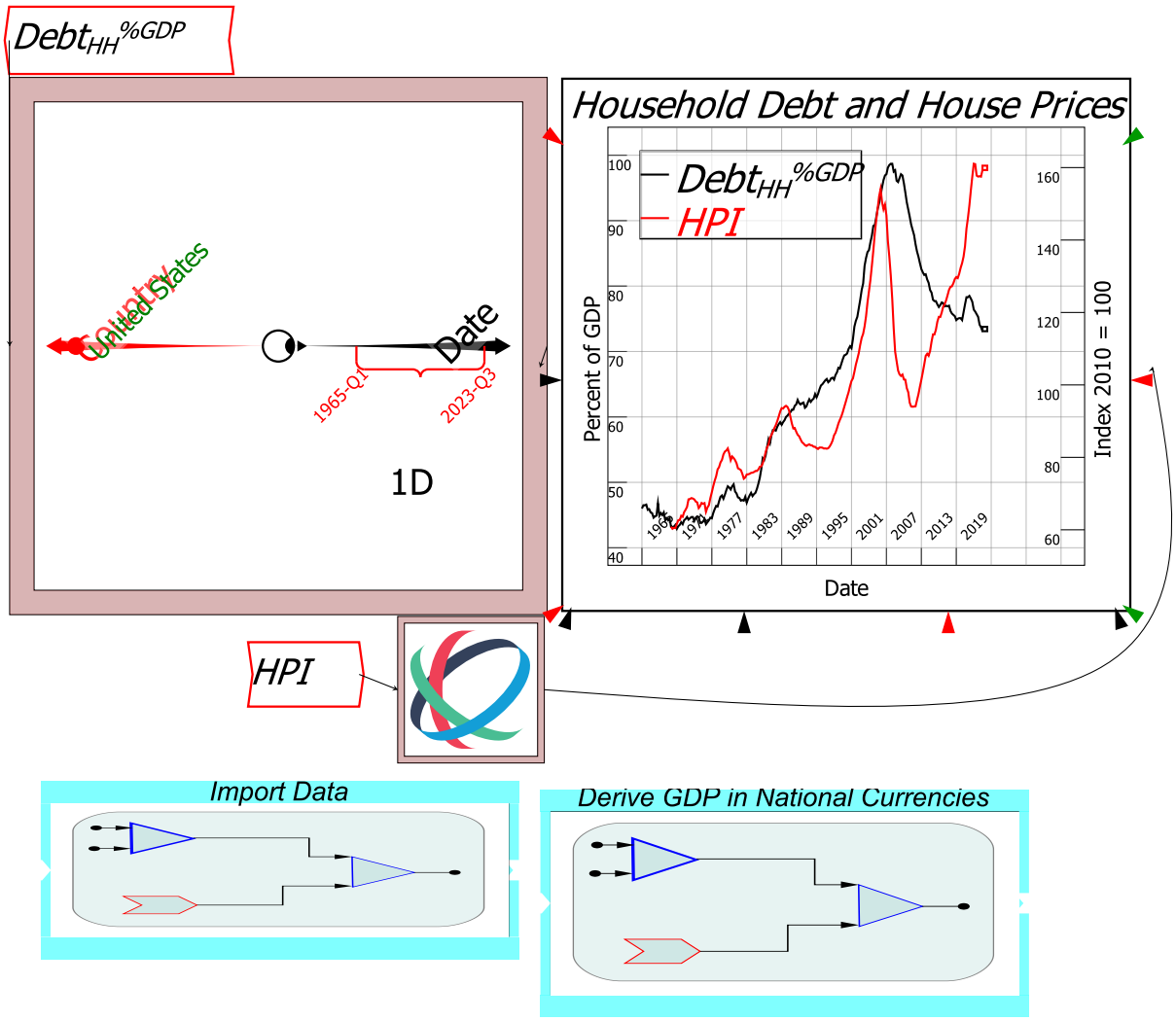


Figure 42: Comparing related variables from different data files.

Figure 42 illustrates a few more features of *Ravel*:

- The data import procedures and the calculation of GDP have been put in *Groups*, to reduce clutter. A group is created by click-and-drag selecting of items, followed by using the right-click menu on the selected widgets, and choosing "Group". Grouping is explained later under *Organizing your Data* on page 52;
- A calliper has been applied to the Date axis. Callipers let you select a contiguous range of information on a Ravel axis. This is a right-click menu option which is accessible when the mouse is hovering over the axis;
- The variable $Debt_{HH}^{\%GDP}$ has been "Flipped" on its axis. This, as usual, is a right-click menu option for almost all objects in *Ravel* (you can also rotate any object via the Rotation field in its Edit form);
- The wire connecting the HPI Ravel to the right-hand axis on the plot has been curved. This is done by clicking on a wire and then dragging. Only one curve has been added here, but multiple curves can be added by repeating the process;
- A plot has been inserted; and
- The plot has been given axis labels. The form to control the appearance of plots is, as usual, a right-click menu item "Options" (double-clicking, which is another way to

access the Edit menu for other entities in *Ravel*, brings up an enlarged version of the Plot in a separate window).

The comparison of debt to house price data isn't promising: US house prices rose with rising household debt from 1965 till 2007, and fell with falling household debt from 2007 till 2013; but then house prices rose while household debt fell from 2013 onwards. Superficially, there's no consistent relationship.

However, some researchers argued during the Global Financial Crisis that it wasn't the *level* of household debt and house prices that are causally related, but *change in household credit* and *change in house prices*.[5]

The argument is simple. Most houses are bought primarily with mortgage debt, so that change in mortgage debt is the main source of monetary demand for housing. Given how inflexible supply of housing is, *change in mortgage debt* is therefore a primary determinant of the *level* of house prices. This implies a relationship between *change in the change* in mortgage debt and *change* in house prices. To simplify things, call the change in mortgage debt "mortgage credit". Biggs, Meyer and Pick's assertion is that the main factor causing change in house prices is *change in mortgage credit*.

The *BIS* doesn't supply data on mortgage debt, but most of household debt is actually mortgage debt, and unsecured household debt (mainly credit cards and car loans) is a relatively stable share of GDP. Household debt can therefore be used as a proxy for mortgage debt here.

To check this theory out, we need to derive credit—the change in debt—from the debt data. This uses the backward difference operator 

, which is on the Scan 

menu. The next figure shows the steps in using this operator.

Firstly, the variable $Debt_{HH}^{DC}$ is attached to the Delta-minus operator's input port. Next, the Edit menu for the Delta operator is activated from the right-click menu, and the operator is applied to the Date dimension; the operator is given the argument 4, which tells the program to compare

[5] Biggs, M., T. Mayer and A. Pick (2010). "Credit and Economic Recovery: Demystifying Phoenix Miracles": http://ssrn.com/abstract=1595980.

the value now to the value four quarters earlier (which is one year). Finally, the value of this calculation—for multiple countries and quarters—is assigned to the new variable $Credit_{HH}^{DC}$.

The same process is repeated to get change in household credit, and finally this variable is divided by GDP to produce $\Delta Credit_{HH}^{\%GDP}$: see Figure 43.

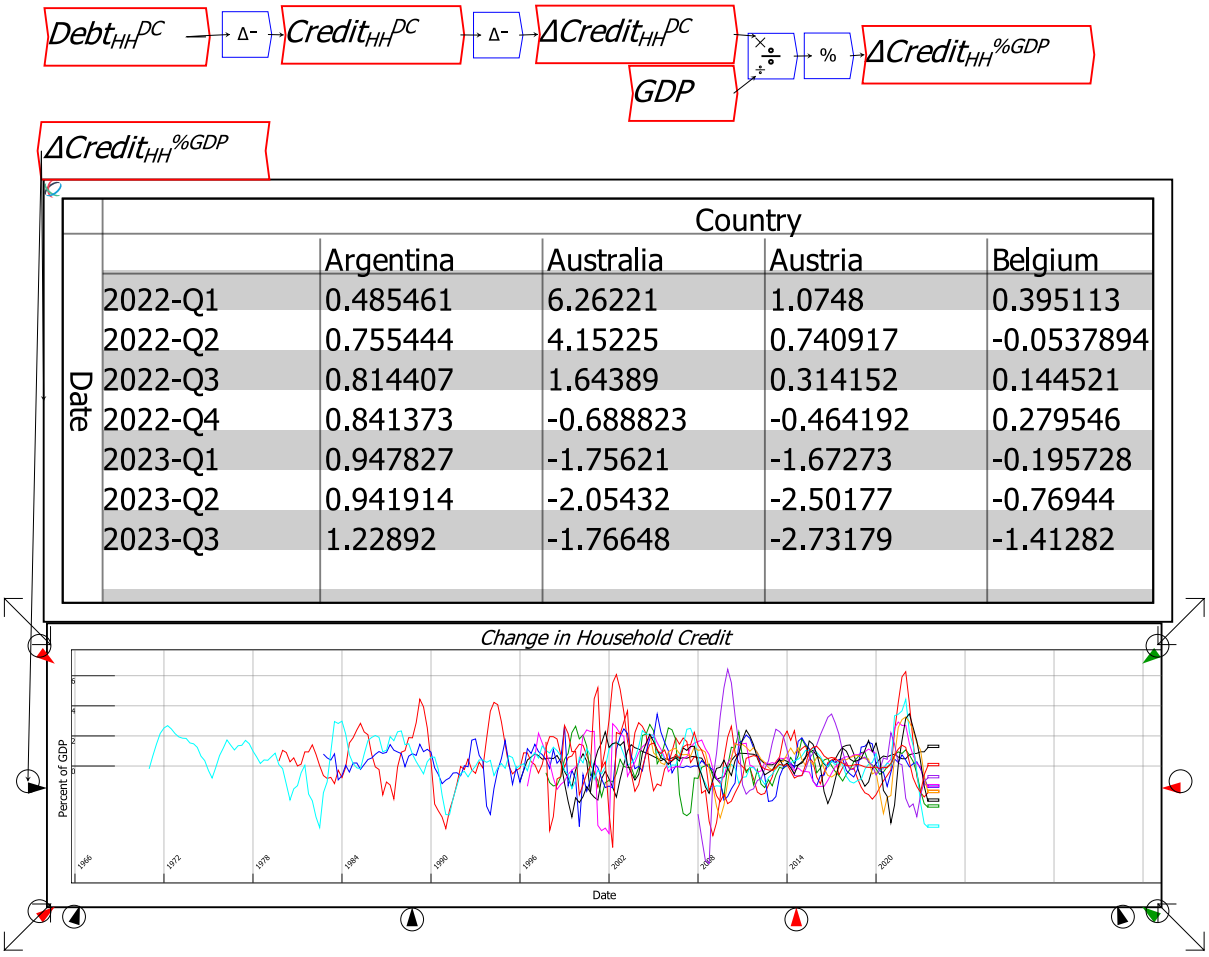


Figure 43: Using the Difference ("Delta") operator

We can now compare this variable to change in house prices. The relationship between changes in household credit and changes in house prices is obvious in Figure 44.

There are currently two difference operators in *Ravel*: the backward difference operator Δ⁻ used here, and the forward difference operator Δ⁺. Future releases of Ravel will include time-based difference operators, where the change is specified in terms of date units (day, month, quarter, year) rather than the number of data intervals.

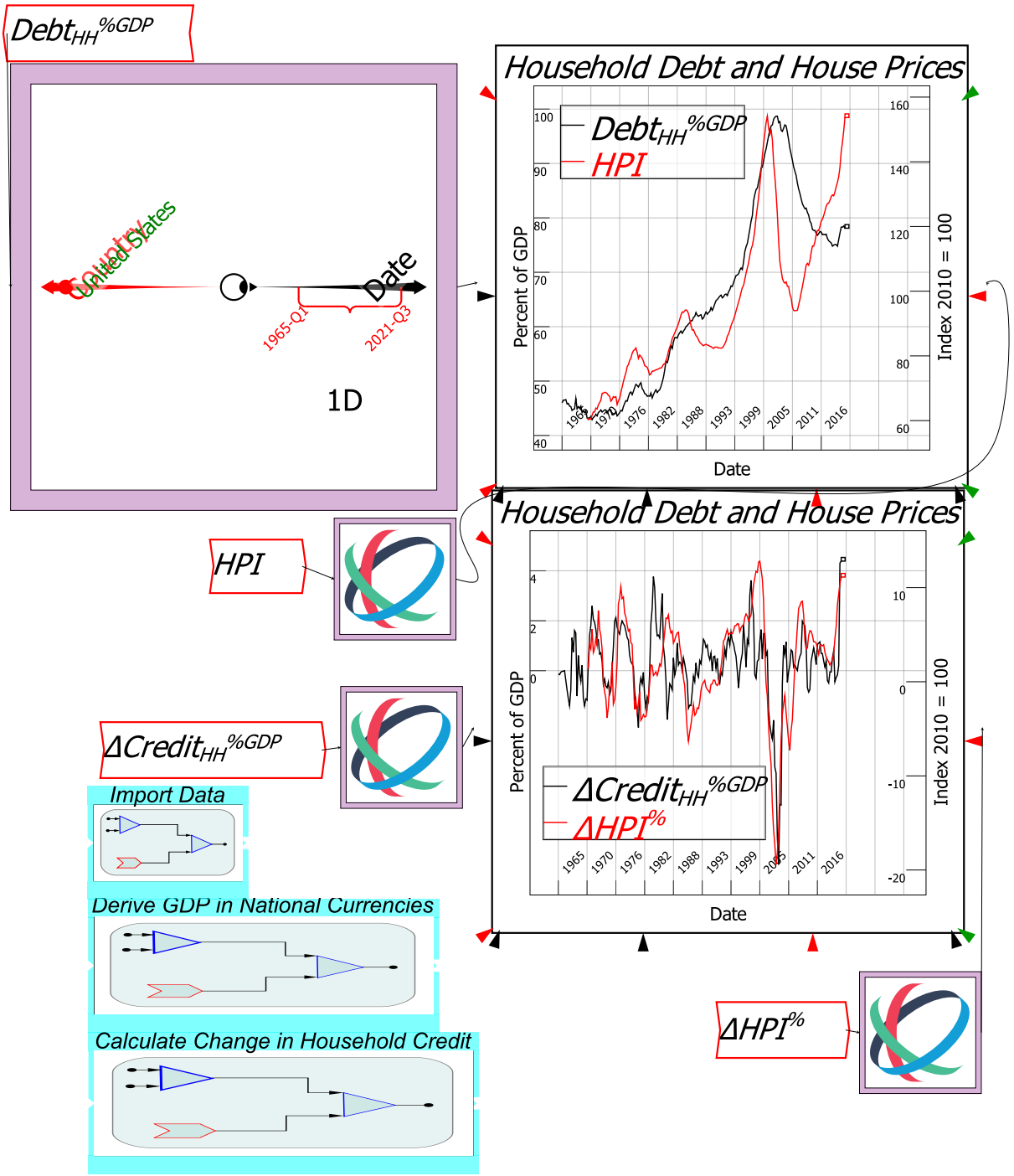


Figure 44: Finding relationships in your data using Ravel

The figure above shows the results for the USA, but Ravel has calculated these results for every country in the BIS database. These simple and easily read equations in this Ravel take the place of tens of thousands of obscure and difficult to read cell reference formulas in a program like Excel.

Exploring Hypothetical Data Using Parameters

Thus far we have used *Ravel* to explore existing data. But there can also be hypothetical data, as in breakeven analysis for sales planning: given estimated costs of production, at what level of

sales will a hypothetical new product become profitable? The capacity to do this with spreadsheets was a major factor in the popularity of the first ever spreadsheet program, *Visicalc*.

Ravel can enable the same sort of analysis using *parameters*. Thus far the only parameter we have encountered is *dataImport*, which loads an external CSV data file. *Ravel* also enables parameters to contain a user-specified number, the value of a variable, or several number-generating functions: *iota*, *one*, *zero*, *eye*, and *rand*. Figure 45 shows the variable/parameter definition form with the drop-down menu of different parameter types.

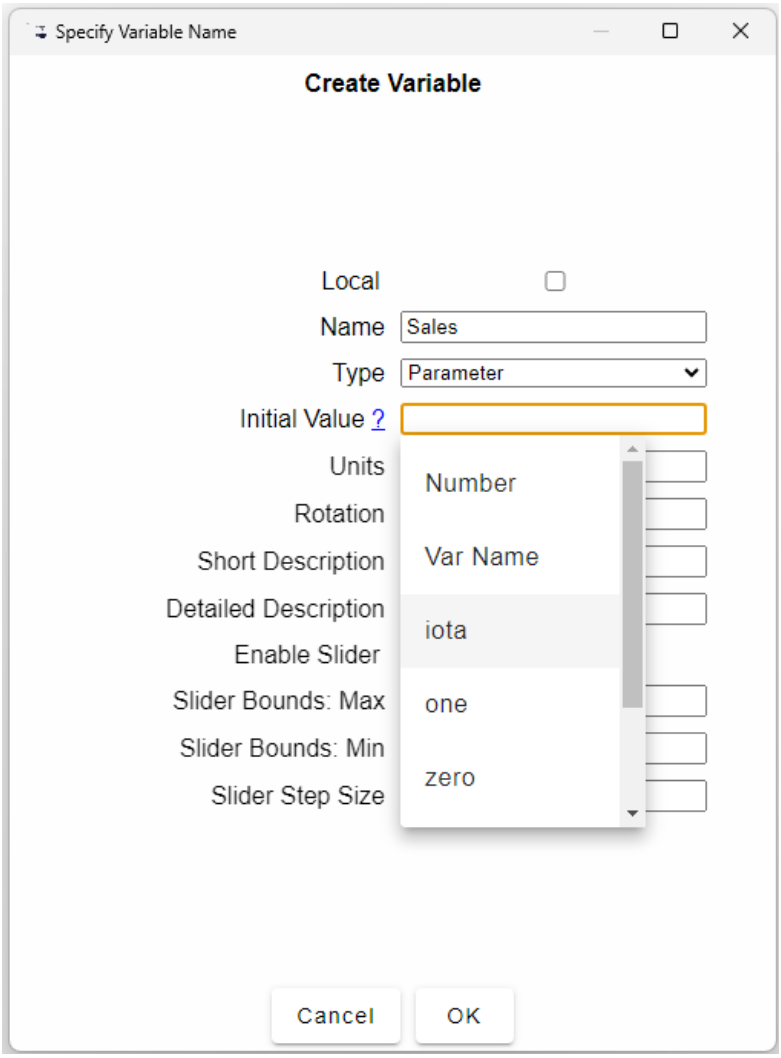


Figure 45: The Edit variable/parameter form showing part of the drop down menu

The effects of these functional inputs to a parameter are given in Table 8.

Table 8: Inputs for parameters

Function	Effect	Example
Number	Enter a single scalar number	4.669201
Var Name	Enter the name of an existing variable	Initializer
iota	An array of numbers starting at zero and incrementing by 1	iota(100): a column of numbers from 0 to 99 Iota(15,15): a 15x15 matrix of numbers from 0 to 224

one	An array where every element is the number 1	one(10,10): a 10x10 array of the number 1
zero	An array where every element is the number 0	zero(10,10): a 10x10 array of the number 0
eye	An array where every diagonal element is the number 1	Eye(3,3): a 3x3 array with 0 on the off-diagonal and 1 on the diagonal
rand	An array of random numbers between 0 and 1	rand(10,2): a 10x2 array of random numbers

Figure 46 shows the output of some of these functions.

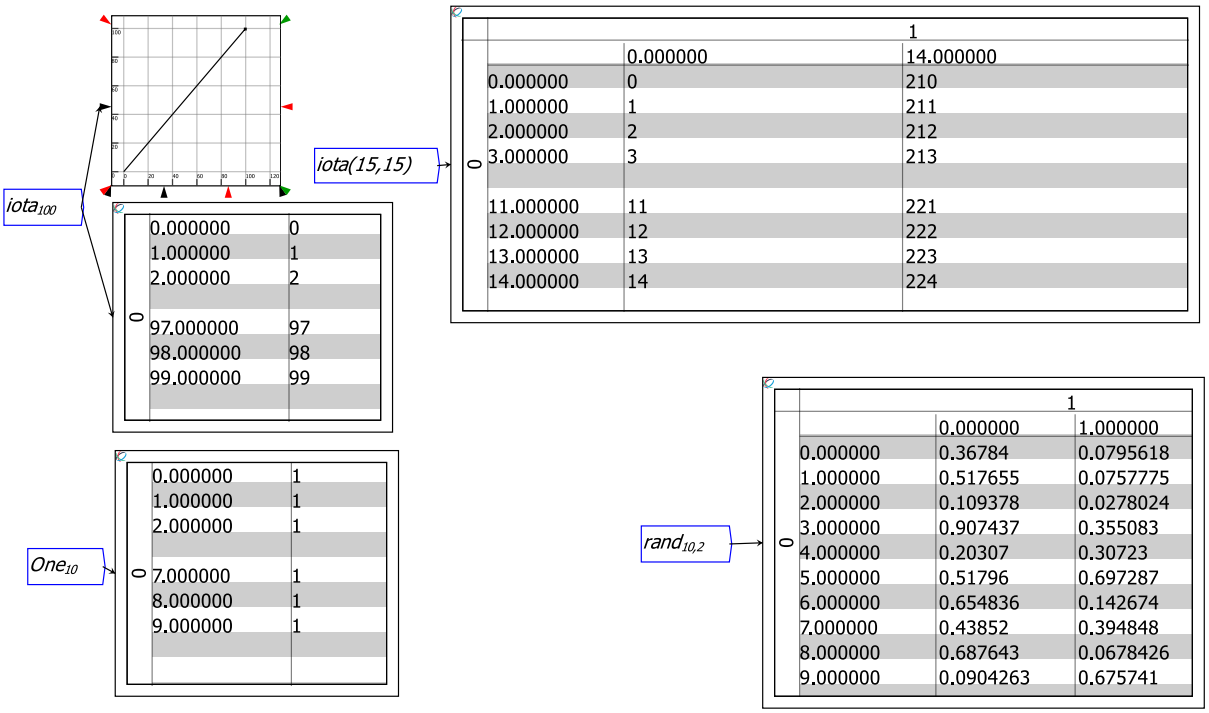


Figure 46: Examples of numbers generated by parameter functions

These arrays of numbers can then be combined using *Ravel*'s mathematical operators to do "What if?" analysis. Figure 48 shows a breakeven analysis for a factory with a fixed cost of $1 billion, a production range of between zero and 500,000 units per year, with variable costs per unit of $25,000 and a variable markup, currently set to 1.25.

The parameter *Sales* has the argument *iota(500000)*, which produces an array of half a million numbers between zero and 499,999. Formulas which depend on Sales—such as the $Cost_{Total}$, $Cost_{Fixed}{}^{PerUnit}$ and $Cost_{Total}{}^{PerUnit}$ formulas shown in Figure 47—would require 1.5 million formula replications in a spreadsheet.

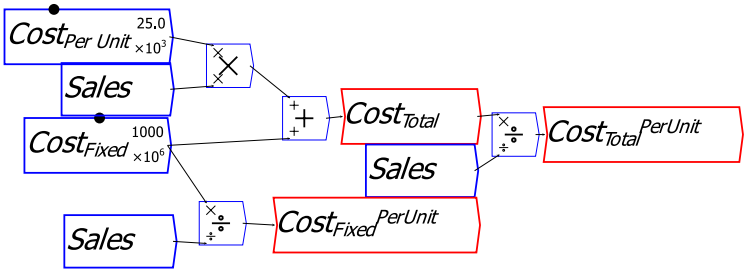


Figure 47: Cost formulas based on the hypothetical data in Sales

The markup can be changed using the arrow keys or mouse to see the impact on the breakeven point and profitability.

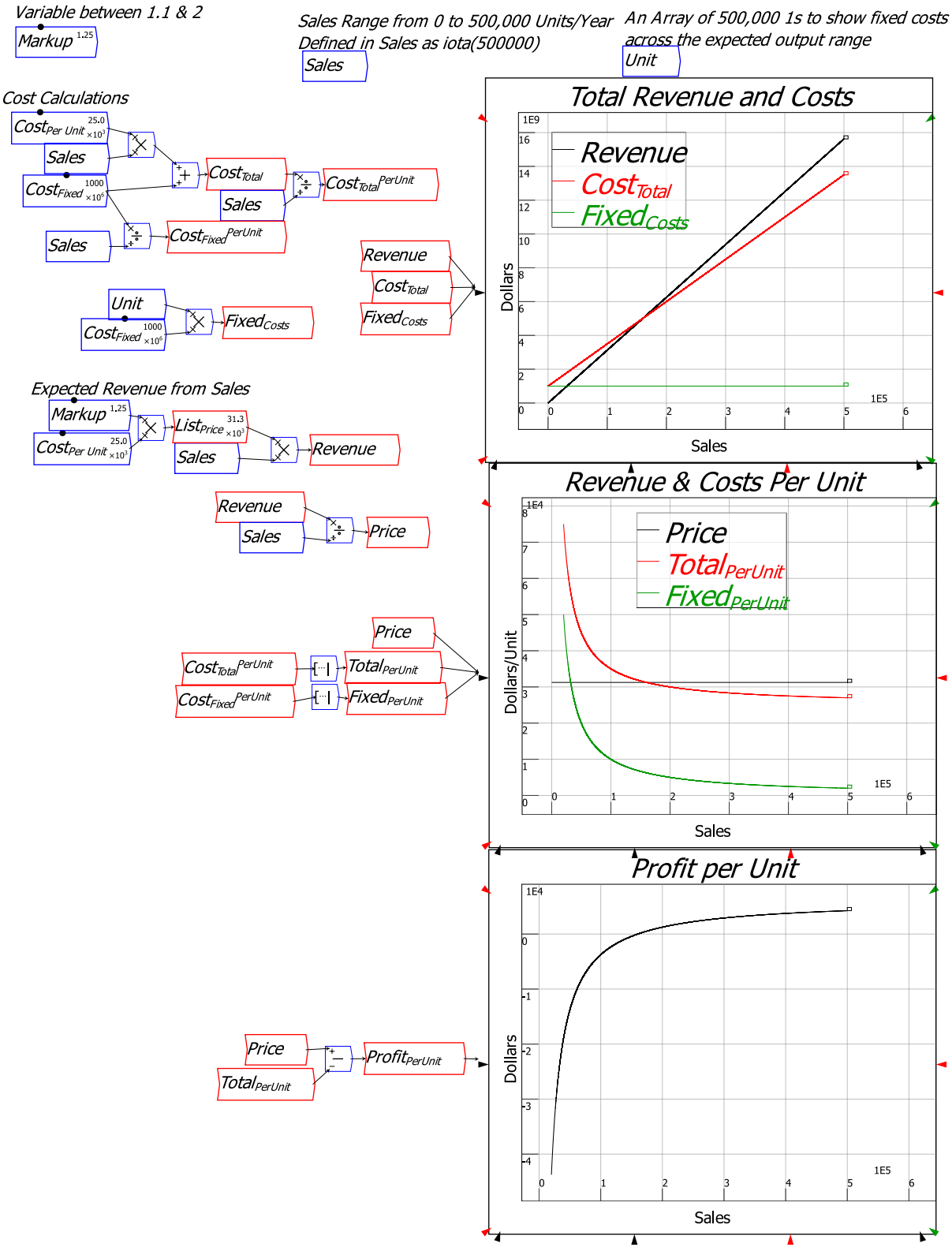


Figure 48: Breakeven analysis with Ravel

The next two figures show the use of the iota function to build a typical economics textbook models of perfect competition, where price is unaffected by the firm's sales (Figure 49), and of monopoly, where price has to fall to allow more sales (Figure 50).

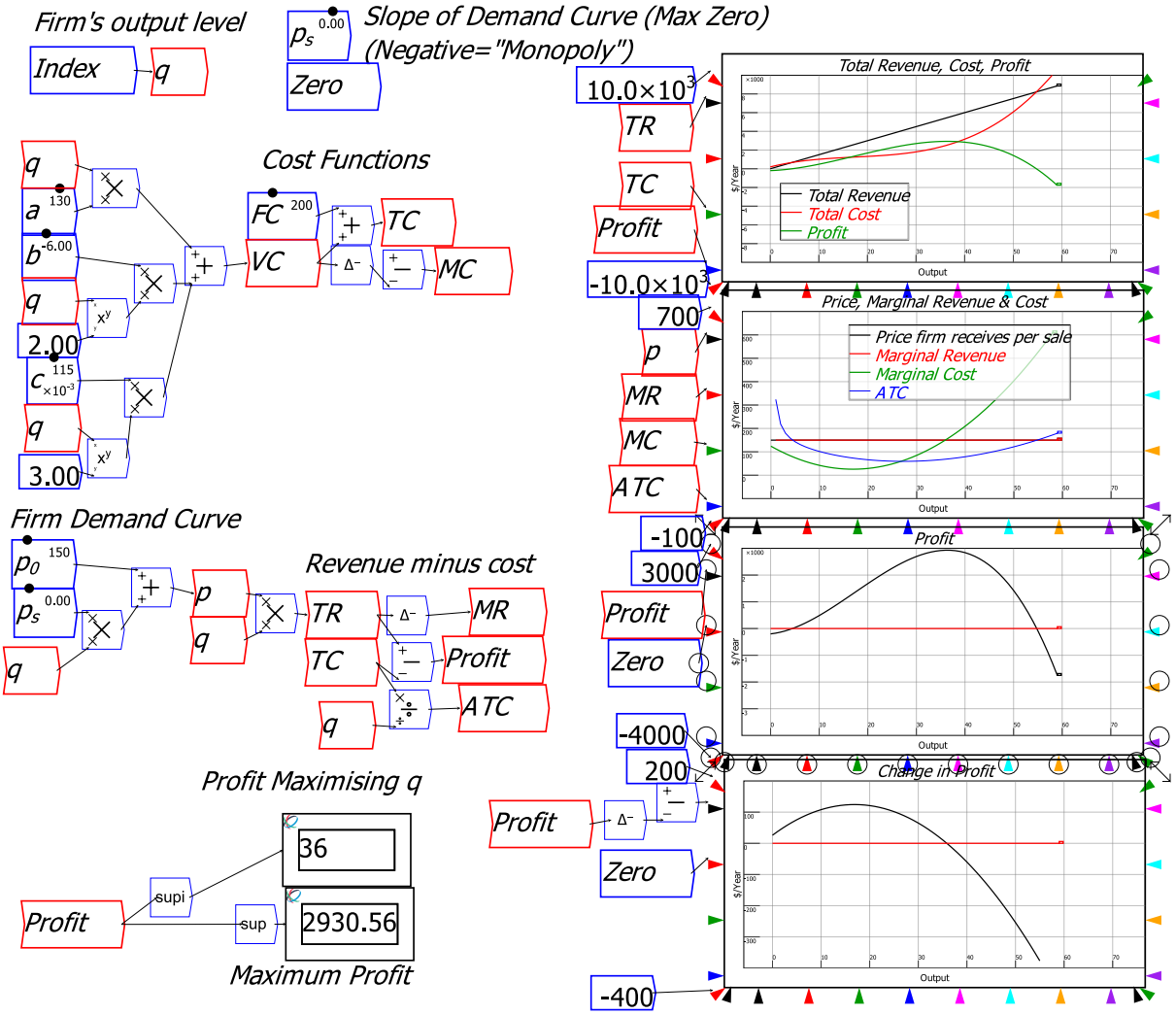


Figure 49: The economics textbook model of a competitive firm where price doesn't vary with output

Ravel's English-language formulas make it much easier to follow the analysis than it would be with spreadsheet versions of the same models.

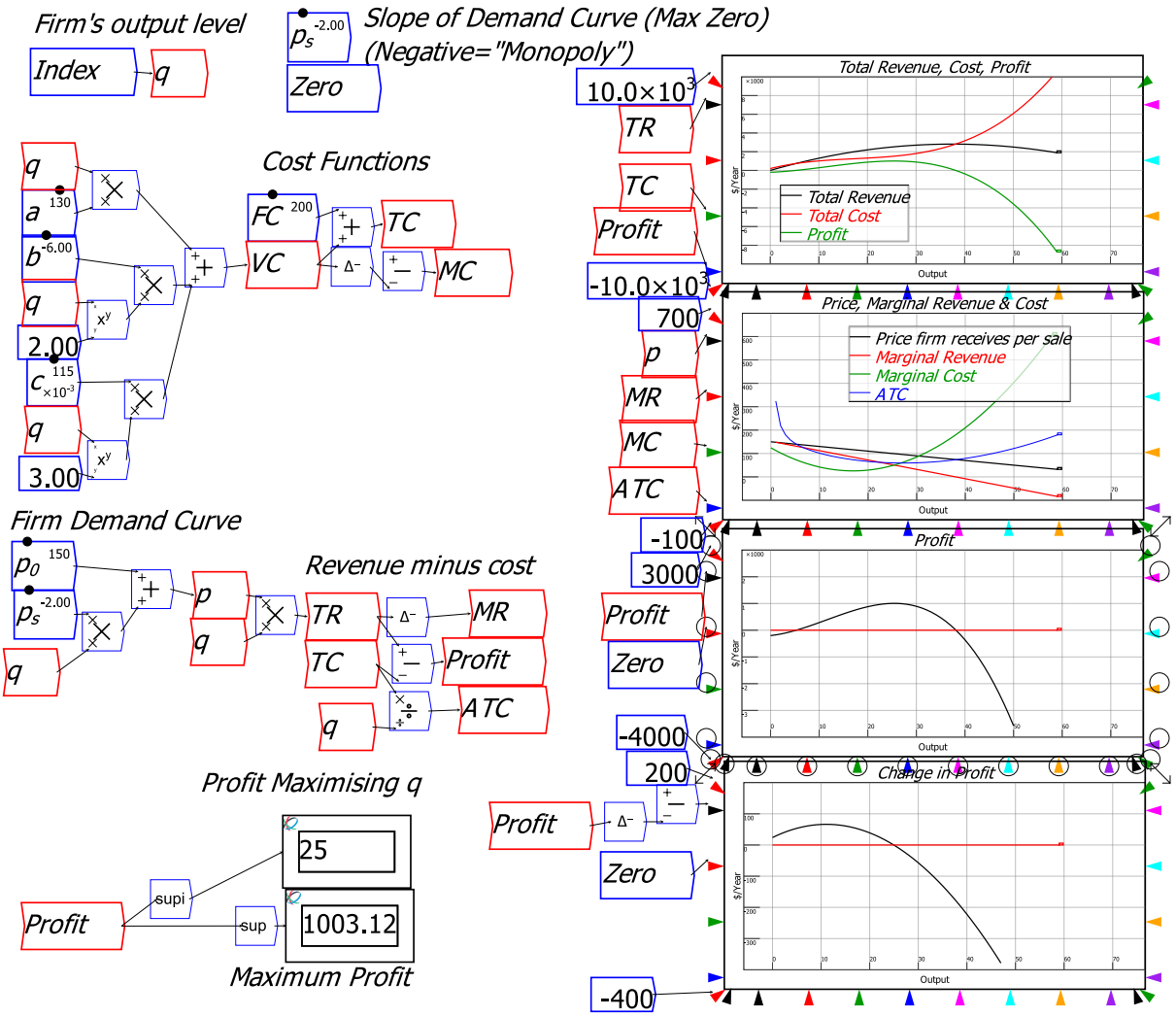


Figure 50: The textbook model of monopoly, with the slope of price set to minus 2

Statistical Analysis

Ravel supports a range of statistical functions (the range will expand significantly in future releases), the widgets for which are on the <x> menu, and are explained in Table 6. The relationship between change in household credit and change in house prices is an obvious candidate for correlation analysis—see the highlighted additional widgets in Figure 51.

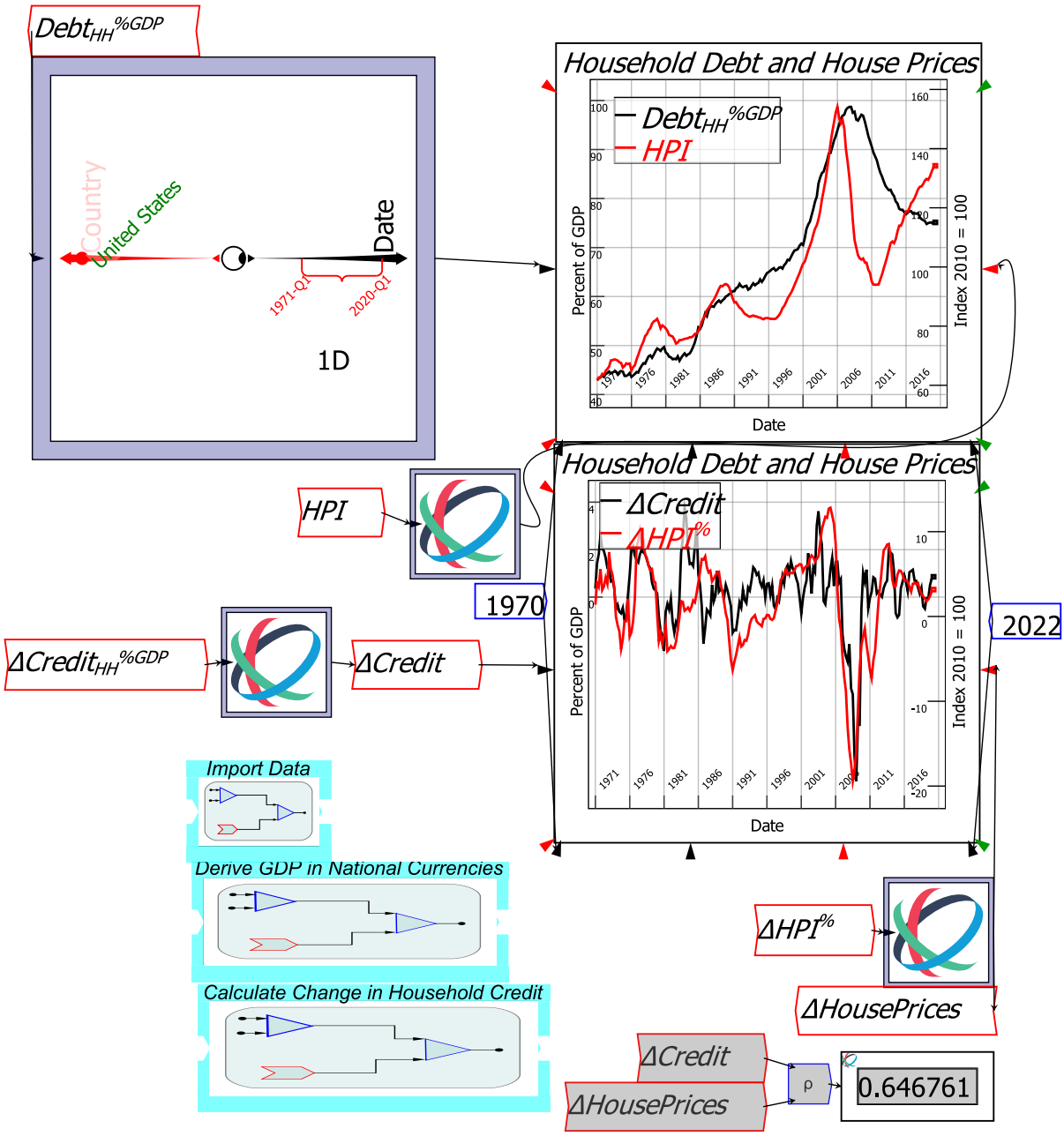
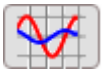


Figure 51: The correlation coefficient for change in household credit and change in house prices between 1970 and 2022 for the USA is 0.65

Plots and Sheets

There are two main ways to display your results in *Ravel*: Plots  and Sheets  .

Plots

Figure 52 shows the default Plot. It has ten inputs, though only one needs to be connected to populate a plot:

- One input for data for each of the 2 Y-axes;

- Two inputs for data for the X-axis, which are used to create X-Y plots but are otherwise left blank; and
- 6 angled inputs which set the range of values for each of the 3 axes.

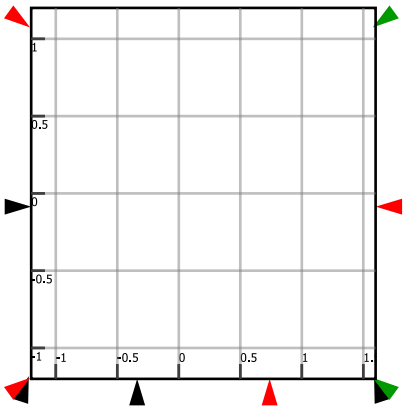


Figure 52: The Plot

Later versions of *Ravel* will add 3 more inputs: "Markers" for the Y and X axes. These draw a dotted line on the plot at a specified value, which is useful when you want to compare a plot to a reference value. Currently markers for the Y1 and X axis are controlled by drop-down menus on the Options form.

As with mathematical operators like add and multiply, the input ports on the two Y-axes can have multiple inputs, which allows multiple series to be plotted on one Plot. A Ravel with multiple values—say the rate of inflation in European countries—will also generate a multi-series plot.

The appearance of a Plot is controlled by two forms, one for Legends etc (Options), and the other for the colour and style of lines and bars (Pen Styles)—see Figure 53. Both are right-click menu choices.

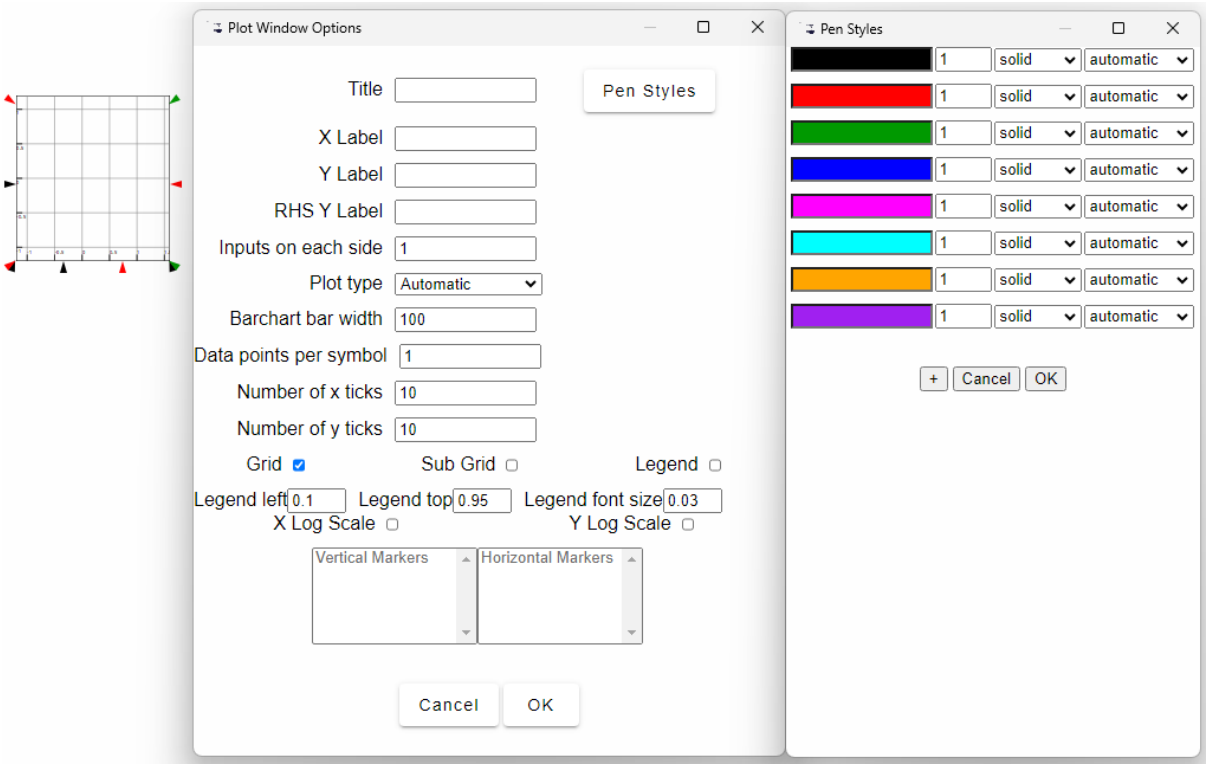


Figure 53: Controls for the appearance of Plots

Figure 54 shows the use of several of these features:

- The top two Plots have inputs on both Y axes, which enables the comparison of two data series over time;
- They also have values (the dates 1970 and 2022) on the X-axis range inputs;
- The middle plot has “No Change” as a horizontal marker;
- Its y-axes ranges are determined by a set of parameters g_t^l, g_b^l and $scale_{lr}$, which make it easy to align the scales on both axes; and
- The bottom plot has an input on the X axis, which enables a scatter-plot comparison of change in household credit with change in house prices.

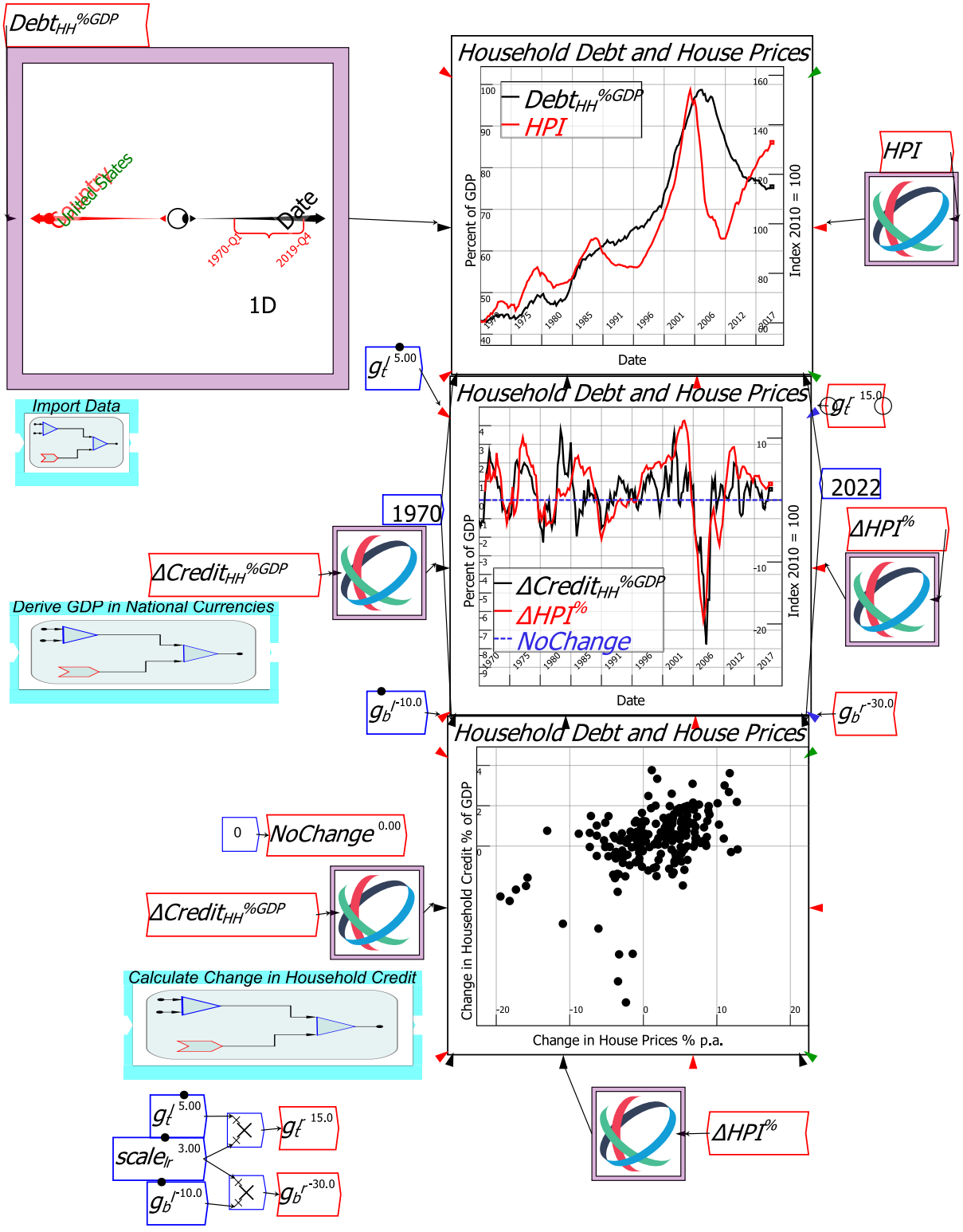


Figure 54: Wiring canvas showing use of the different Plot input ports

Figure 55 shows the same plots on a publication tab. The settings on the Ravel can be altered—say, moving the country selector from the USA to Japan, or changing the date range—which in turn alters what the plots display.

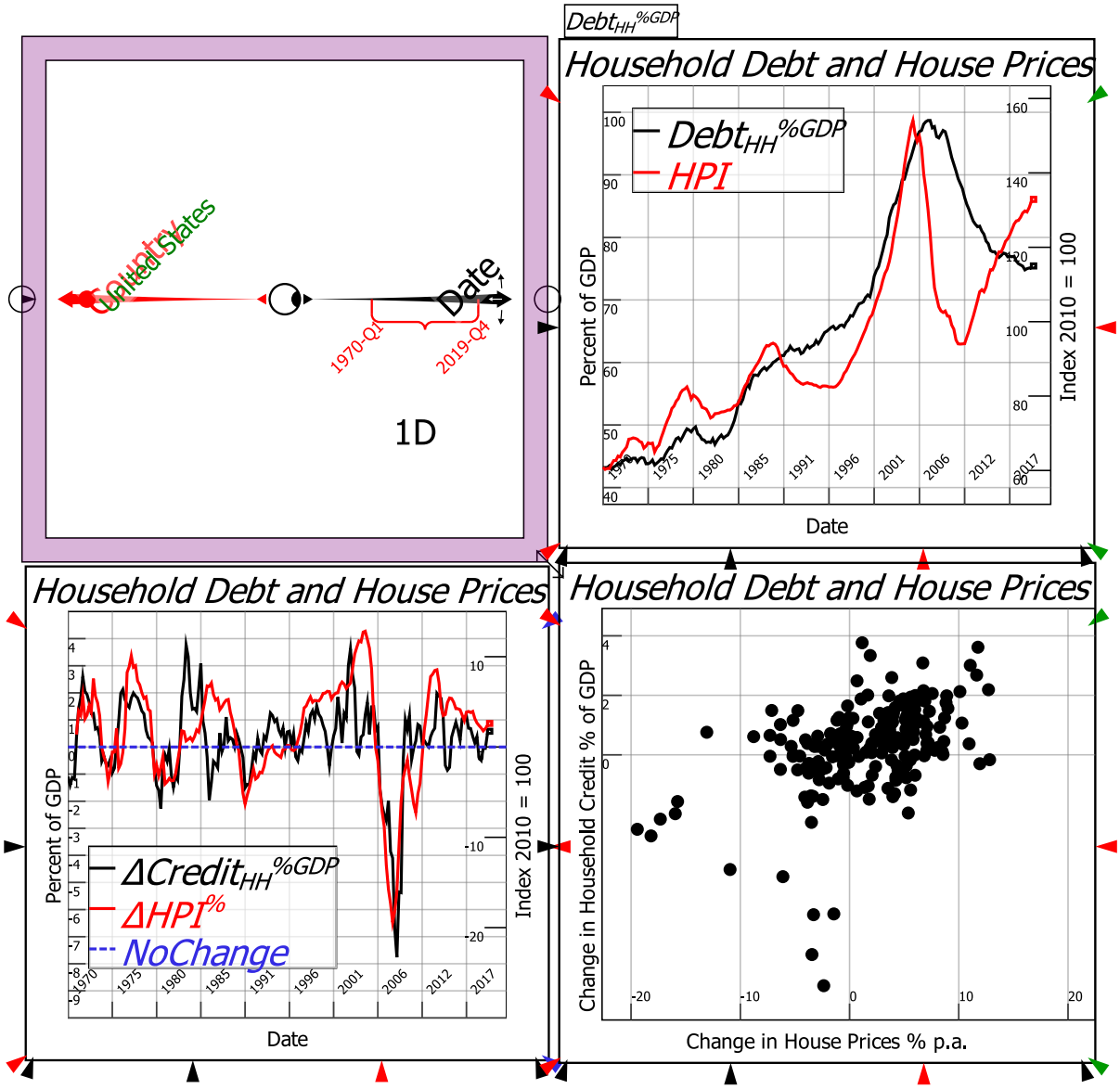


Figure 55: Ravel Publication Tab with several plots

There will be many improvements to Plots in the near future. We have focused on the core aspects of Ravel in its pre-release development; now that we are ready to release Ravel commercially, the "bells and whistles" are receiving attention.

Sheets

Figure 56 shows the default sheet. It is big enough to display a couple of rows of data, but it can be resized using its corner arrows to show a larger amount of data.

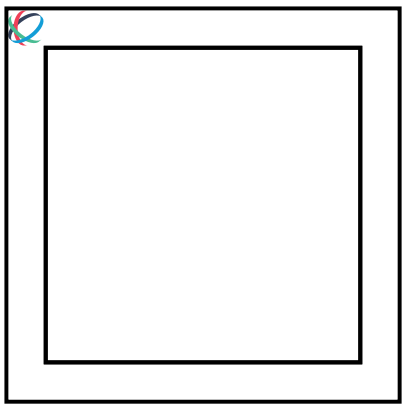


Figure 56: The default Sheet

At present, sheets can only take one input (unlike Plots), which restricts them to showing only the contents of one Ravel or variable, as with Figure 57—but there is a sophisticated workaround to this, as explained below.

The main control over the appearance of a Sheet is the "Row Slices" command, which determines whether a Sheet displays the first entries in a variable, the last, or a mix of the two.

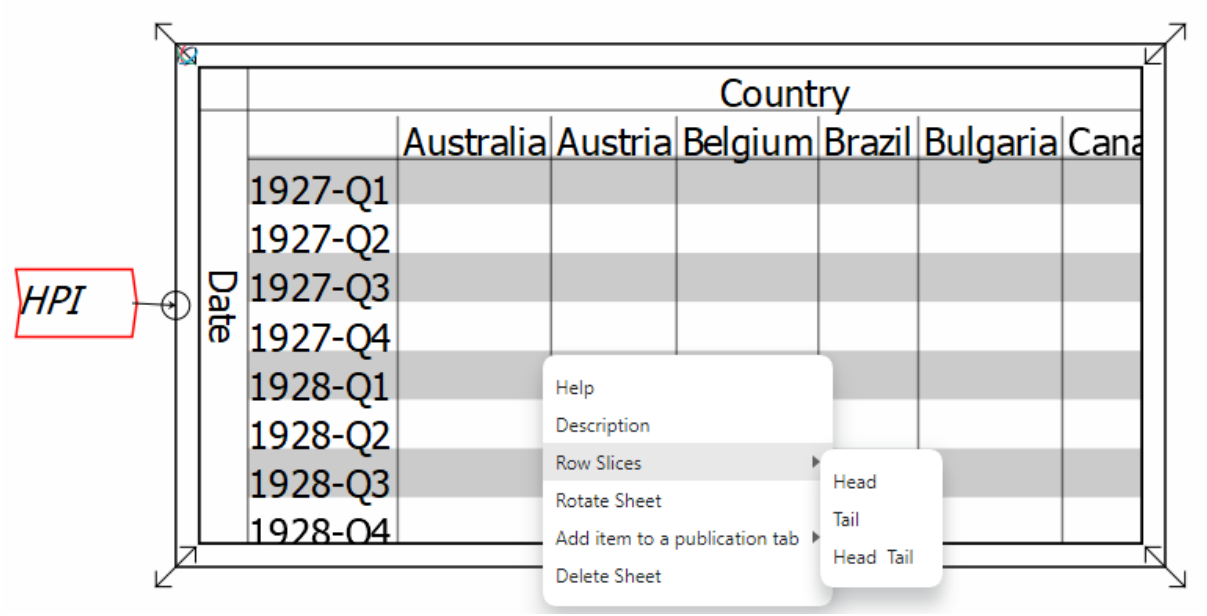
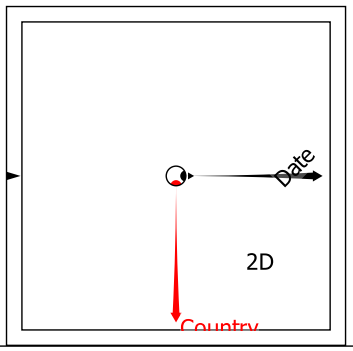


Figure 57: The default sheet expanded and with a variable attached

Given the limited controls over a Sheet at present, it's best to attach a Ravel to control what a Sheet displays (in a future release, Sheets and Plots will be integrated into Ravels, making this a one-step process).



	Country										
Date	Austria	Belgium	France	Germany	Greece	Ireland	Italy	Luxembourg	Netherlands	Portugal	Spain
2022-Q1	165.122	111.619	111.81	158.423	76.7863	133.223	76.1252	186.012	129.222	147.651	86.1696
2022-Q2	165.239	111.175	111.114	157.012	75.4158	130.944	76.4551	185.401	129.861	144.995	85.0242
2022-Q3	161.276	111.01	112.826	153.409	78.5657	132.354	74.0326	188.729	125.394	147.705	85.4797
2022-Q4	154.552	107.246	111.062	142.622	80.0765	131.487	70.9075	183.997	120.476	146.321	84.8562
2023-Q1	151.104	108.754	108.526	136.421	83.6631	129.69	70.6296	175.444	121.02	148.612	84.9326
2023-Q2	148.381	107.737	106.362	133.157	84.3882	126.126	71.6514	168.583	117.591	151.066	85.5698
2023-Q3	146.612	108.465	106.175	130.412	86.1663	126.36	71.3728	156.725	117.646	153.573	86.9031
2023-Q4	143.144				86.6337						

Figure 58: A sheet on a Publication Tab with a linked Ravel attached and the Tail of the row data displayed

To display the contents of more than one variable on a Sheet, use the Merge  command. This Ravel-specific widget takes in two or more variables which share dimensions, and adds a new axis to the resulting variable (or Ravel) with the two (or more) variables as entries on that axis. In Figure 59, the two variables $\Delta HPI^{\%}$ and $\Delta Credit_{HH}{}^{\%GDP}$, which share the dimensions Date and Country, are melded into a new axis called Variables. These two data series can now be displayed in the one Sheet.

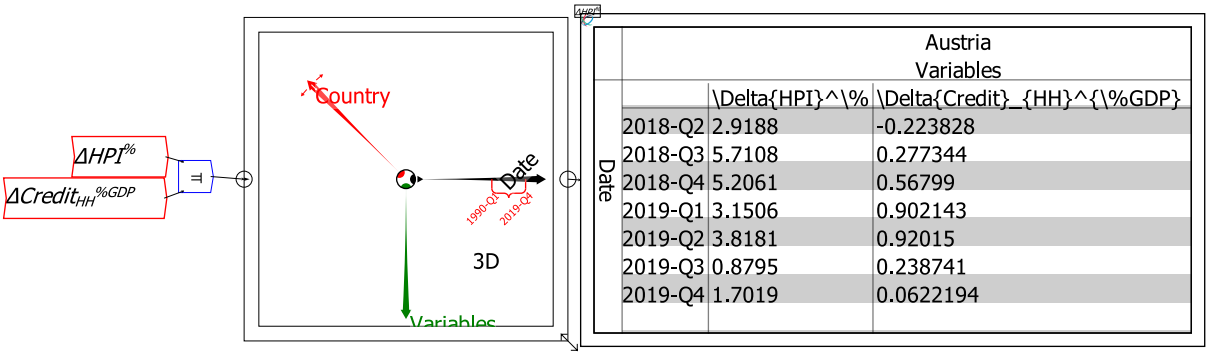


Figure 59: Merging two variables to create a new Ravel with an extra dimension "Variables", set in the Edit menu for Merge 

Organizing your Data

The *Ravel* wiring canvas is essentially unlimited, which means you need tools to organise and navigate it as your model grows. *Ravel* provides two main tools for this:

- Bookmarks, which store a location (and a zoom scale) into a Bookmark menu; and
- Groups, which store a set of operations to reduce clutter on the canvas.

Bookmarks

Bookmarks can be created in two ways:

- By storing the current wiring tab view—both the xy location of the top-left point visible on your monitor, and the zoom scaling in use—via the "Bookmarks/Bookmark this position" menu; and
- By entering a text block and clicking the checkbox to make it into a Bookmark

Figure 60 shows the latter operation. The Bookmark "GDP Calculations" is added to the Bookmarks menu, and you can navigate to that spot by clicking on its entry.

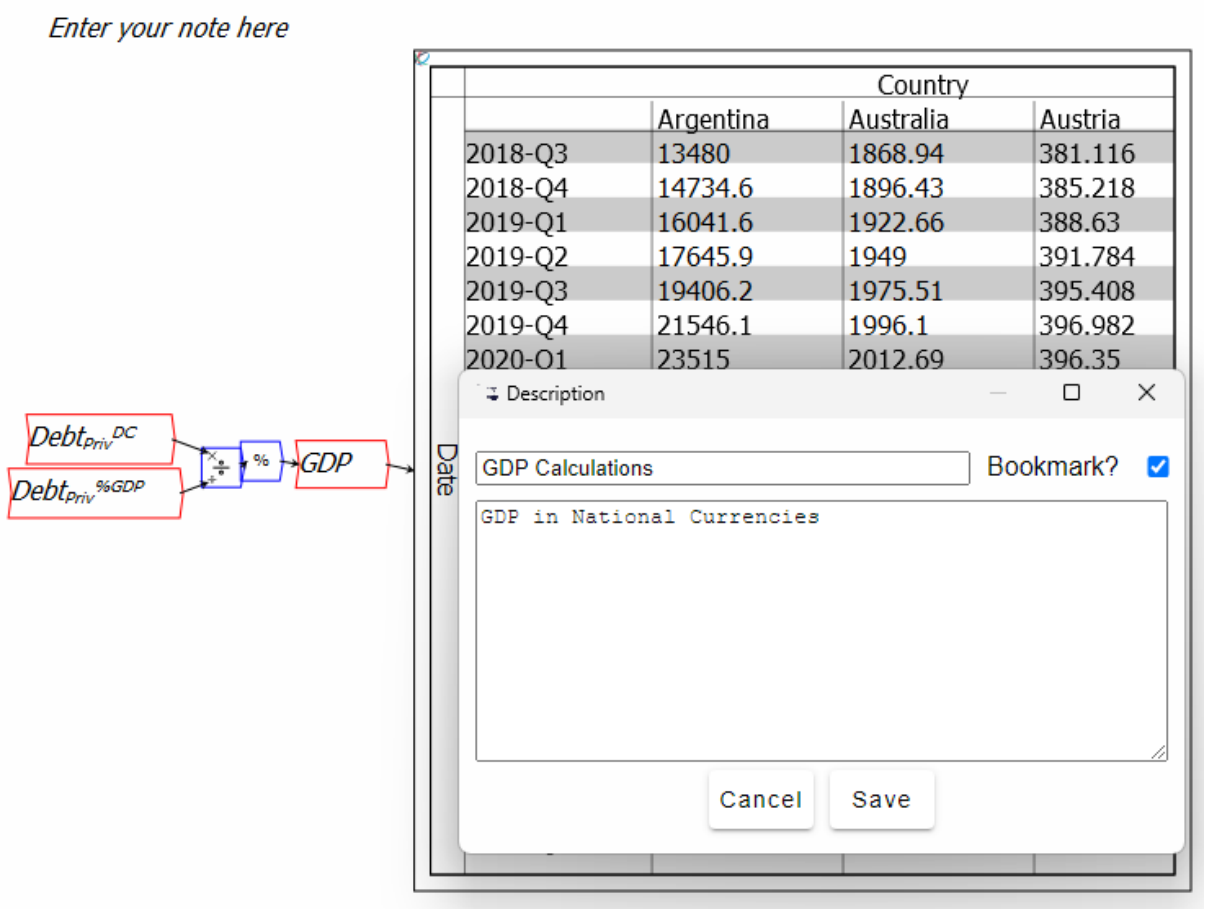


Figure 60: Creating a bookmark from a text entry

Groups

A group is created by clicking on the wiring canvas and dragging the mouse to surround a set of widgets, then right-clicking on the greyed-out area and choosing "Group" from the context menu—see Figure 61.

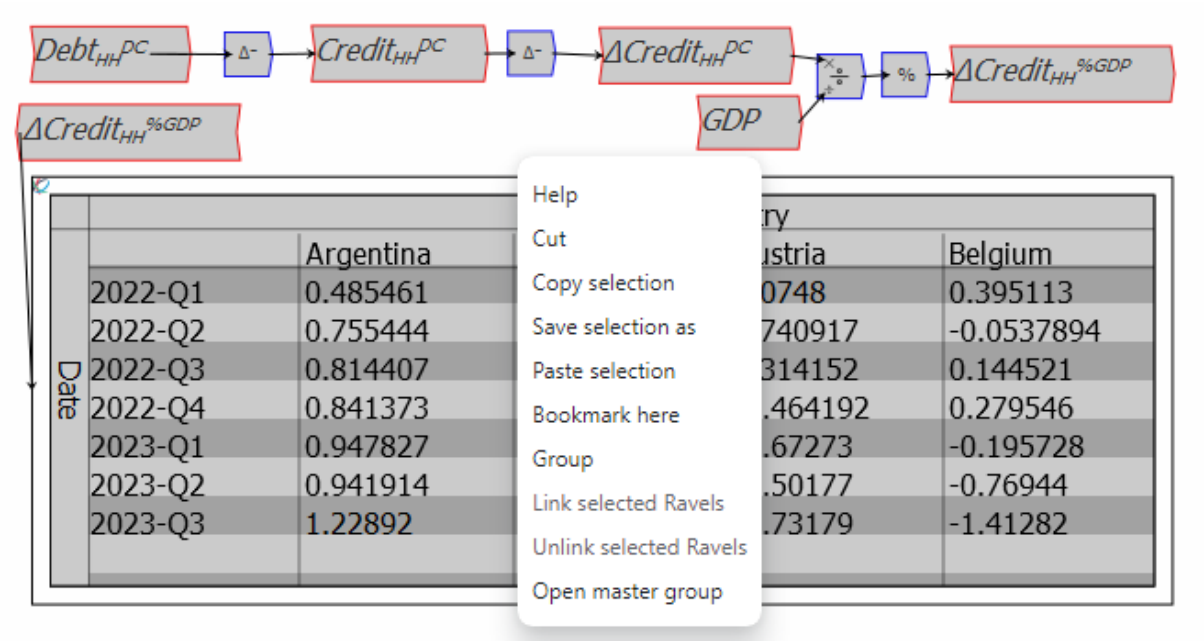


Figure 61: Creating a group

A Group can be given a name via the right-click/context menu "Edit" command when the mouse is hovering over a group.

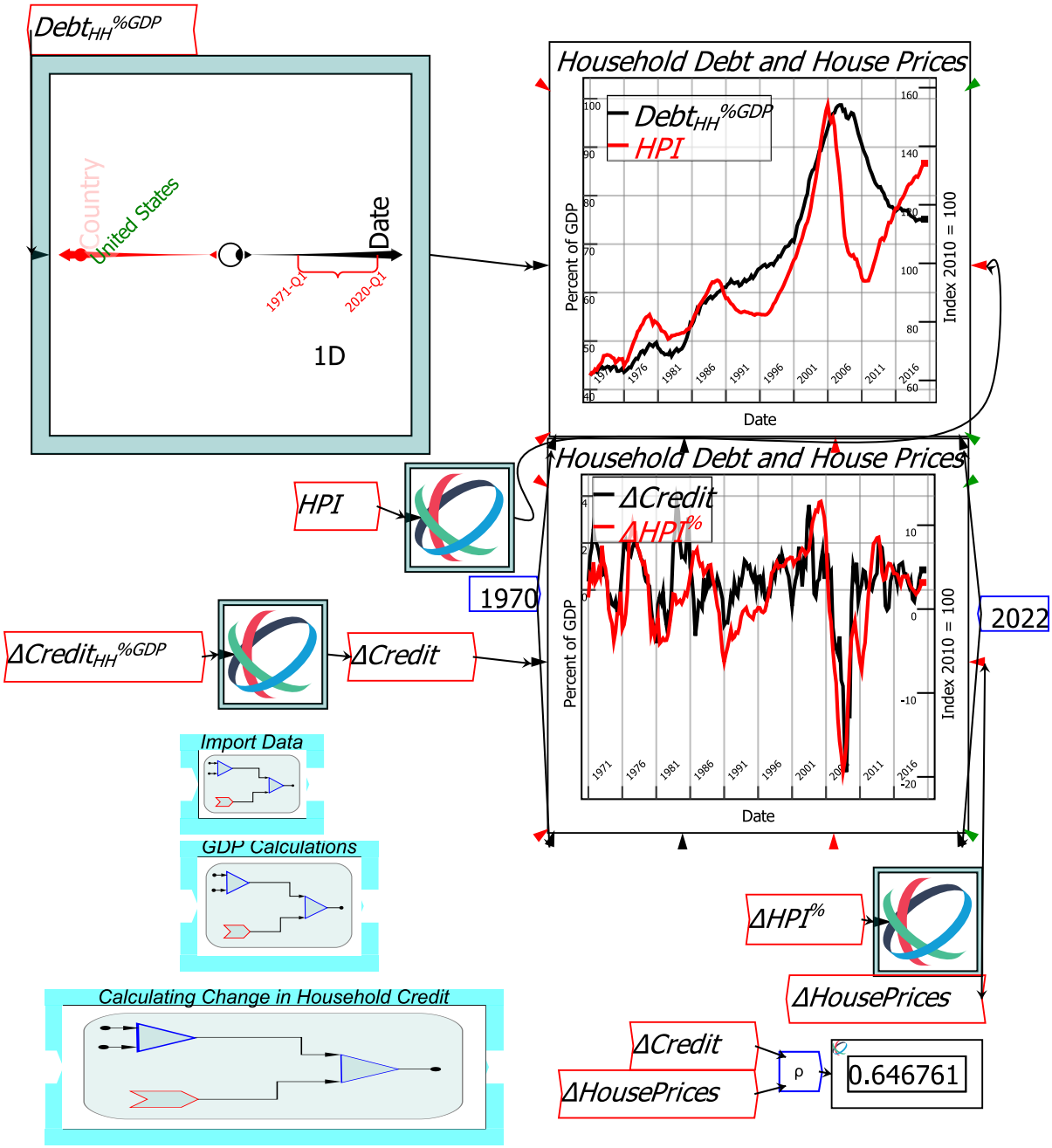


Figure 62: The operations shown in Figure 61 are grouped in "Calculating Change in Household Credit"

A group can be resized using the arrows on its corners. This can reduce clutter on the wiring canvas, while still providing access to the operations within the group if required.

Group Plot

By default, the group display uses a stylized circuit diagram. But you can make the group much more informative by creating a plot within the Group and using the context menu's "Make Group Plot" to identify that as the Group['s plot.

Then that chart will display as the group's icon, as shown in Figure 63.

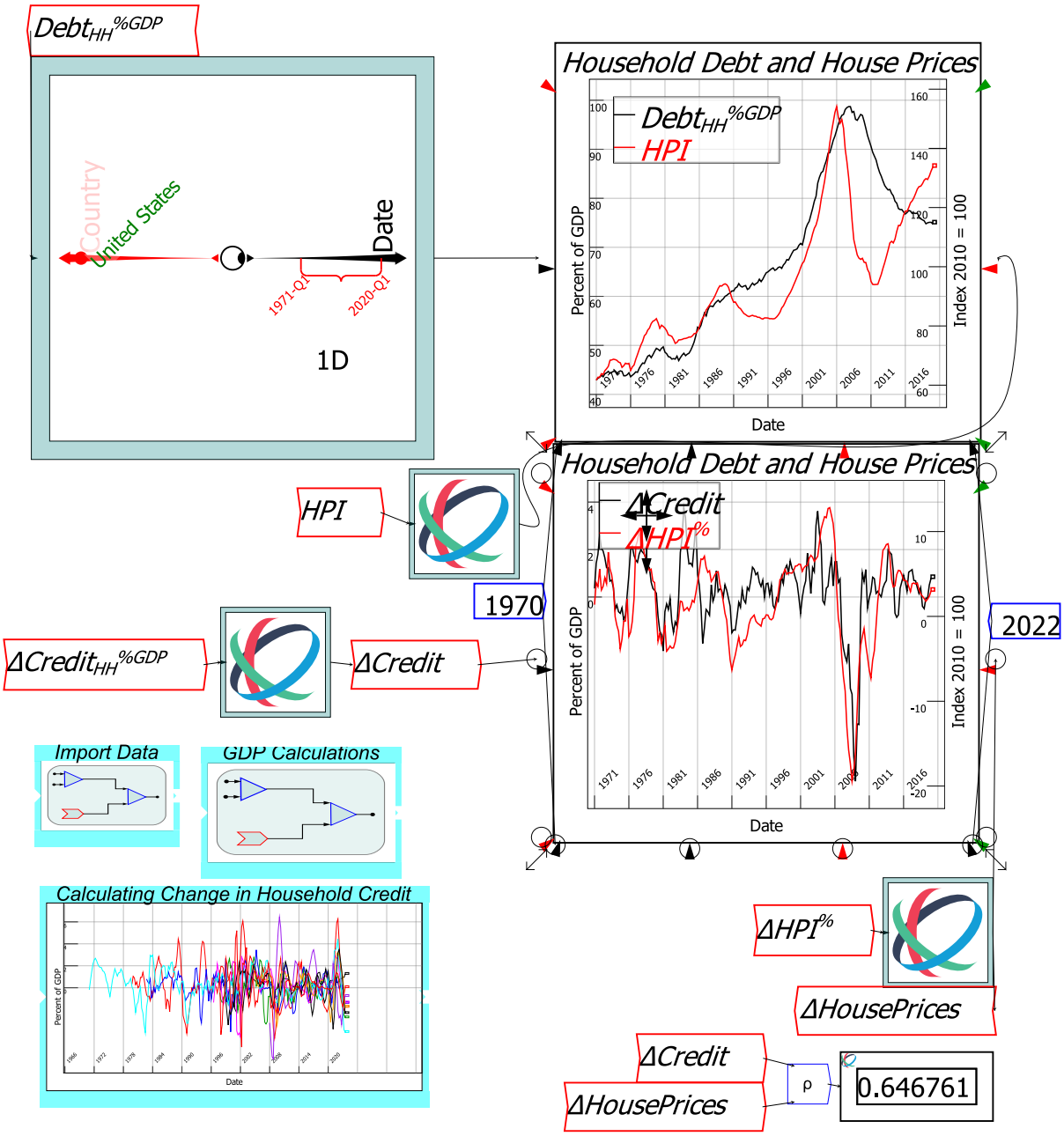


Figure 63: Using a plot as a Group's icon

Editing a Group, and Group Transparency

There are 2 ways to edit the contents of a group:

- By hovering over a group and choosing "Open in Canvas" from the context menu; and
- By zooming in on the group.

Once the Zoom is sufficient to identify the components within the Group, the group icon will become transparent, and you can edit items within the group without having to open it on the canvas.

The Browser Window

A large document can have many variables and parameters defined in it. The Browser window, which is accessible via the Var  widget, or via the Insert menu, makes it easy to access these variables and parameters so that you can work with them to do your analysis.

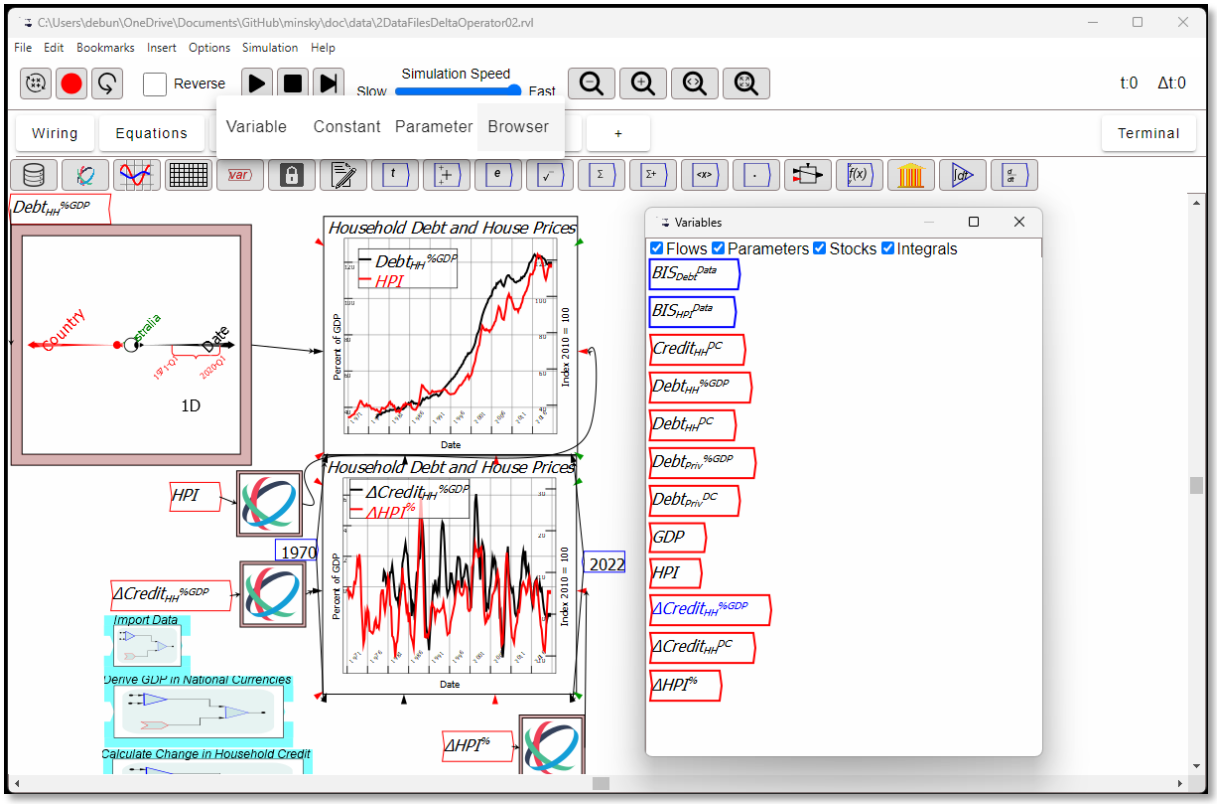


Figure 64: The Browser window

The Browser window is an "Always on Top" window that you can place anywhere on screen. When you click on an object in the Browser window, that object is attached to your cursor so that you can place it where you want it on the canvas.

Model Documentation

Adding text

Ravel's text tool  enables you to place text anywhere on the wiring canvas. There are two ways to enter text:

- Click on this tool, which attaches the default text string "Enter your note here" to your cursor; or
- Type the # key anywhere on the wiring canvas. That brings up the Text Input form shown in Figure 65, which is also used for defining Variables and Parameters. When you press Enter or click OK, this text string will be attached to your cursor.

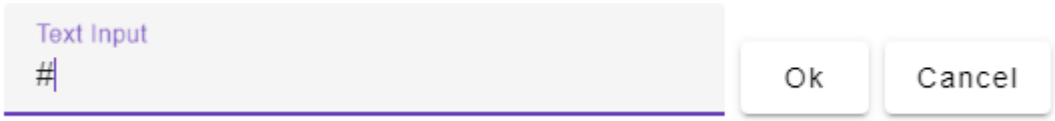


Figure 65: The Text Input form opened by typing # on the canvas

Once you have clicked to place the string where you want it on the canvas, you can then edit it further to, if you wish, include paragraphs of text. There are two ways to commence the editing process:

- Double-click on the text string; or
- Right-click and choose "Description" from the menu

Either action will bring up the text editing window shown in Figure 66. You can type entire paragraphs of text into this window, and the text formatting capabilities that LaTeX enables for variable and parameter names in Ravel are available here as well.

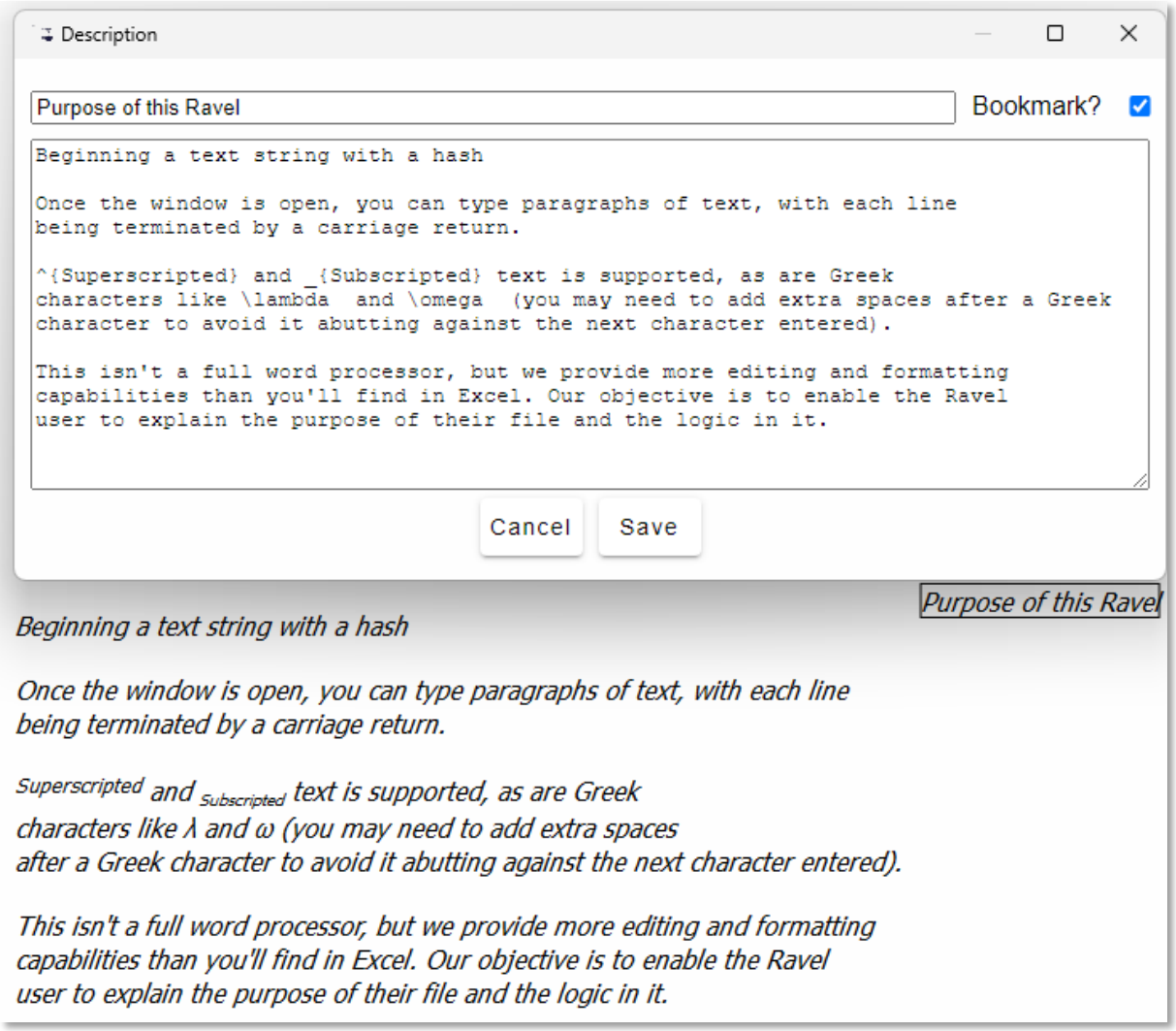


Figure 66: The text editing window and its result on the canvas

Two other methods to document a *Ravel* file are the Summary Tab, and the Publication Tabs

The Summary Tab

Ravel models are far easier to audit and understand than spreadsheet models, where fundamentally only the person who designs a sheet can understand the logic contained in cell reference formulas. In contrast, anyone who can read a flowchart can understand the logic behind a *Ravel* model. In addition, *Ravel* is self-documenting via the Equations and Summary Tabs.

The Equation Tab shows you a (bitmapped) image of the equations in your model. These equations can also be exported to word processing applications via the *File/Export Canvas as/LaTeX* command.

The Summary Tab gives you a well-formatted and structured view of the components of a *Ravel* model, including the dimensions of each variable and the number of values on each dimension.

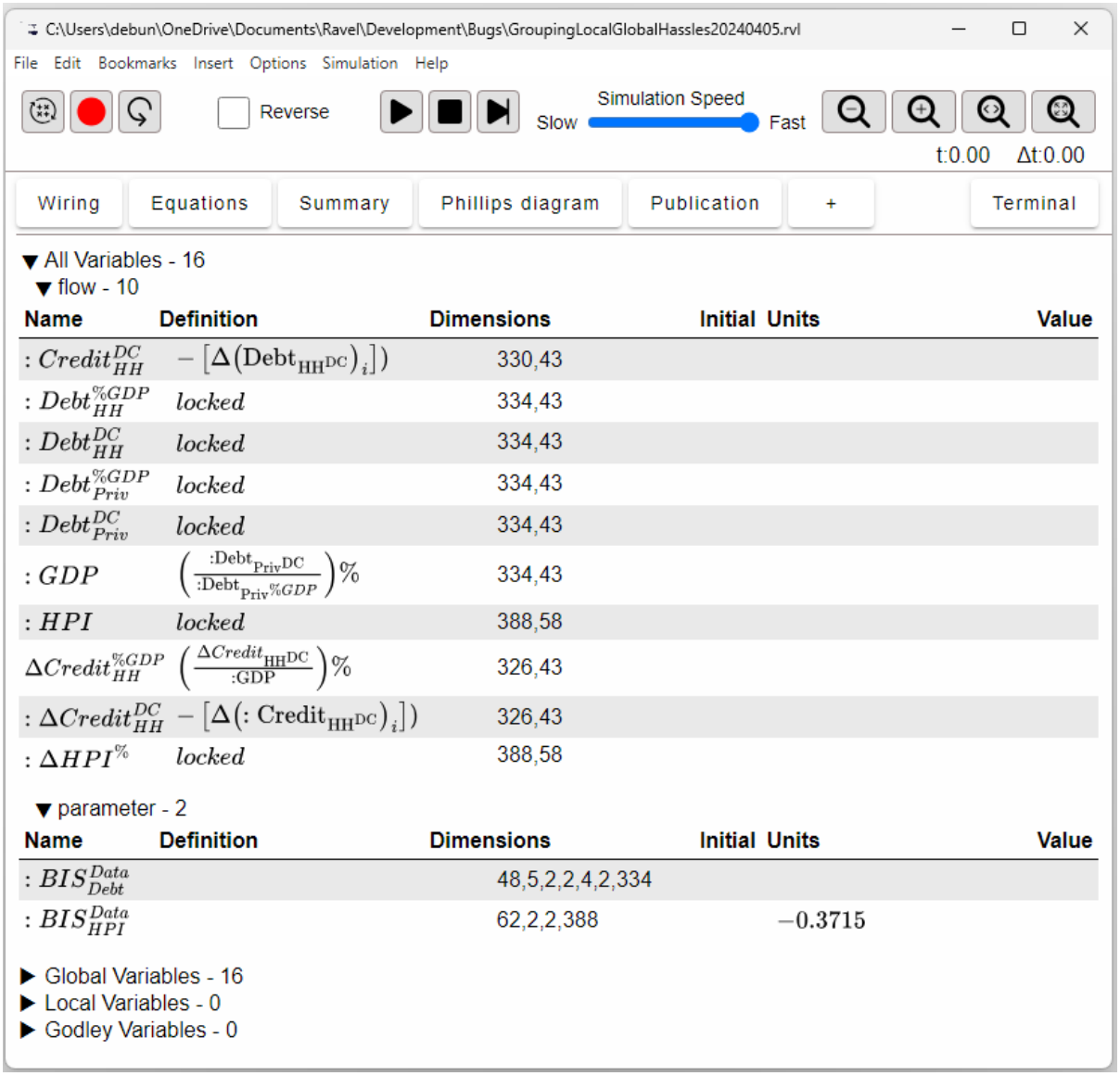


Figure 67: Ravel auto-documenting a model using the Summary Tab

Publication Tabs

Publication Tabs let you show the results of your calculations, without necessarily showing the workings behind them. Items on the Wiring canvas can be placed on a Publication Tab via the—you guessed it—right-click menu. Put the mouse over an object, click on the right-mouse button, choose "Add item to a Publication Tab", and select one of the Tabs in the menu.

Items are first placed in the upper left corner of the Tab. You then click on them and drag them to where you want them on the Tab. A Ravel is dragged by clicking on its bounding box, as for the Wiring Tab (clicking inside the Ravel itself can change settings on the Ravel, rather than moving it on the canvas).

The figure below shows the key Ravel and graphs from this document on the Publication Tab (more Publication Tabs can be created by clicking on the + Tab). The components on the Tab are editable, so that you can alter the country being displayed and the date range on the Ravel, and the graphs will automatically update. This figure shows that the relationship found between credit acceleration and house price change for America applies even to Australia, which was one of the few countries not to experience a recession during the Global Financial Crisis.

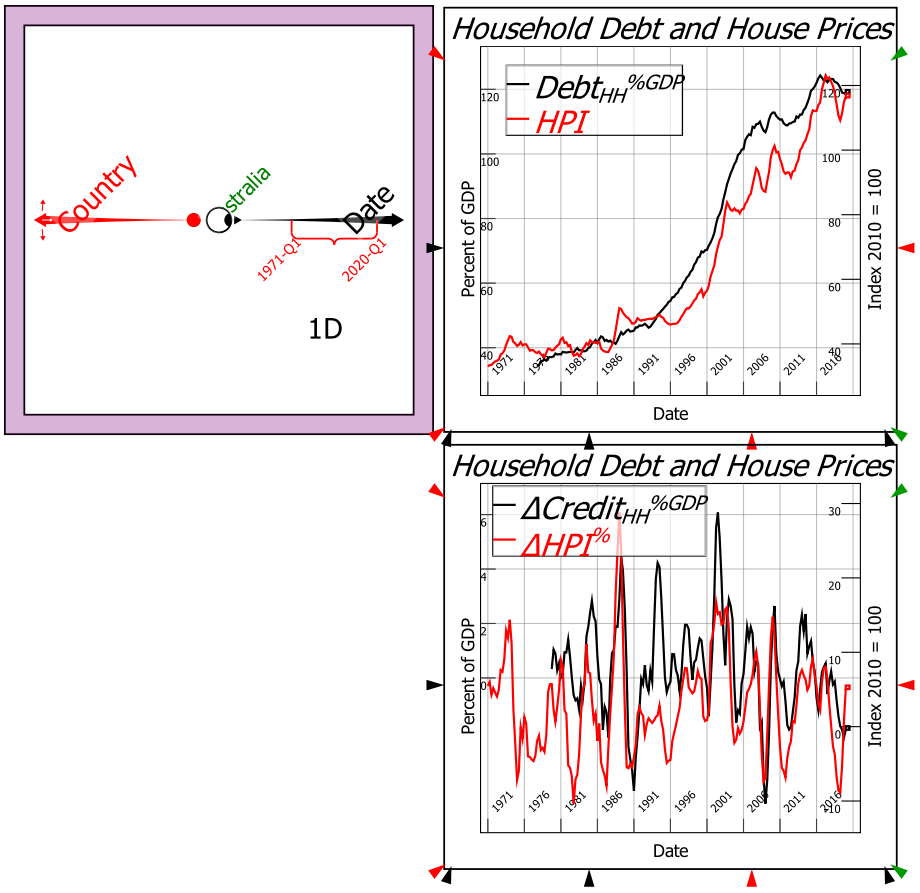


Figure 68: Using Publication Tabs to show results without the clutter of equations

Working with Other Programs

Ravel can export any canvas to a number of other file formats, using the "File/Export Canvas as..." command.

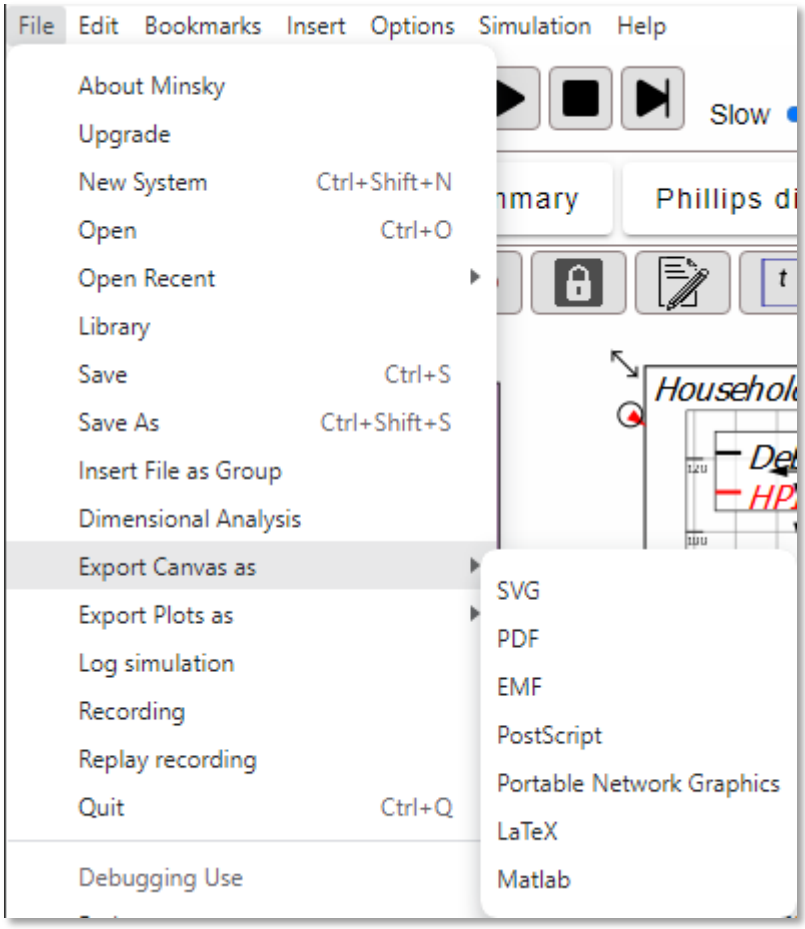


Figure 69: The different export formats in Ravel

The file types and their primary functions are detailed in Table 9 below.

Table 9: File export options

File Type	Suffix	Function
Scalable Vector Graphics	SVG	Import into word processor, graphics programs etc.
Portable Document Format	PDF	A document anyone can read with Adobe Acrobat
Enhanced Metafile	EMF	A Windows-specific vector graphics file
Postscript	EPS	Encapsulated version of PDF
Portable Network Graphics	PNG	Bitmapped format for paint programs etc.
LaTeX	TeX	Equation-oriented word processor text-format
Matlab	M	The Matlab numerical analysis program

Importing Data into Ravel

Any data can be loaded into a spreadsheet, because the rows and columns in a spreadsheet are just placeholders. *Ravel* is pickier, since its dimensions—the equivalent of a spreadsheet's rows and columns—specify a unique piece of information in your data. Therefore, every cell must have a unique identifier in terms of the dimensions of the data. The first task in using *Ravel* is to choose the columns in a CSV file that give you a unique specification of your data.

This first release of *Ravel* only imports CSV—"Comma Separated Value"—files. This is advantageous, in that virtually every program can output CSV, and also problematic, because

there is no pre-defined format, and no error correction for bad data, in CSV files. A file can have no header at all, or many lines of header. What is supposed to be a date field can have characters accidentally stored in them—a "B" rather than an "8" from a file created by an OCR program, for example—or the formatting can change from one date format to another without warning. Some knowledge of the source data is therefore necessary when importing a file.

The figure below shows the BIS debt data in *Excel*. Clearly this file has been exported from a transactional database, since there are numerous columns—such as *BORROWERS_CTY*—which takes codes—such as *HU*—then generate the full text country names—*Hungary*—in the *Borrowers' country* column. Only one of these columns-- *BORROWERS_CTY* or *Borrowers' country*—is needed to uniquely specify the Country, so it is sensible to import one column and ignore the other.

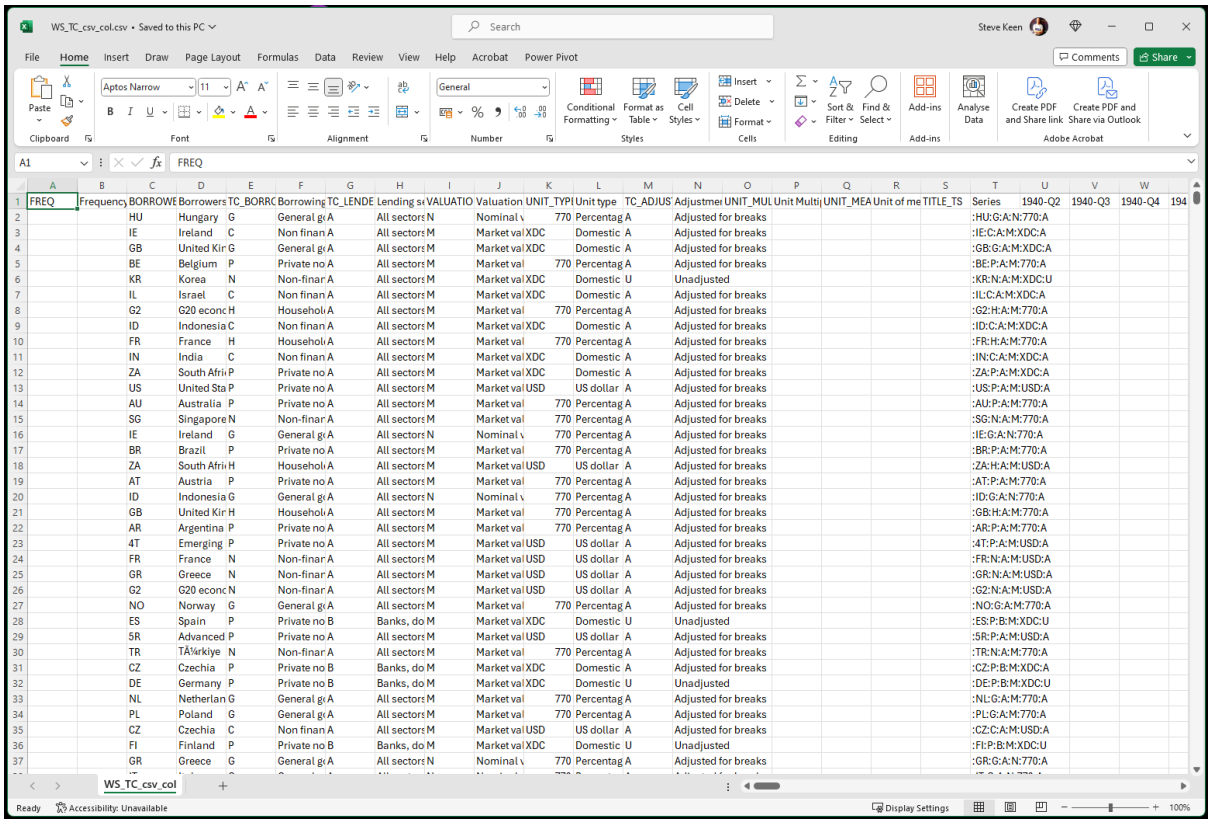


Figure 70: The BIS data in Excel

Choosing which columns to import and which to ignore, until a unique key is generated for each data point, is the role of the Import Data form.

You access this form by clicking on the first widget in the widget bar, the Import Data widget  . You can also activate this function by choosing "Import Data" from the File menu. Either action will (a) attach a parameter named *dataImport* on your mouse pointer, and (b) call up the Import Data form shown below (the precise layout of this form may change as we improve the process in future releases).

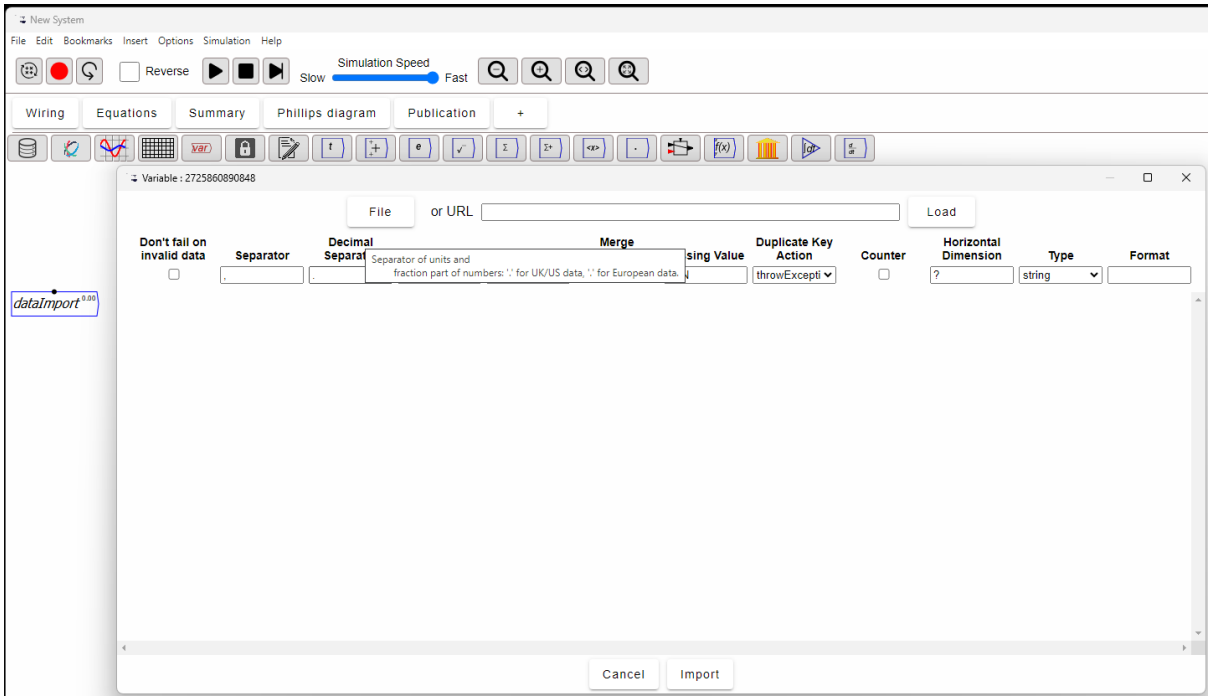


Figure 71: The data importing form

The first step in using this form is to click on the File button and specify the file you want to import (alternately, you can enter an URL into the next text window and Ravel will download that data file from the web). Then click Load and the form will load the file for import processing. The figure below shows the BIS file loaded.

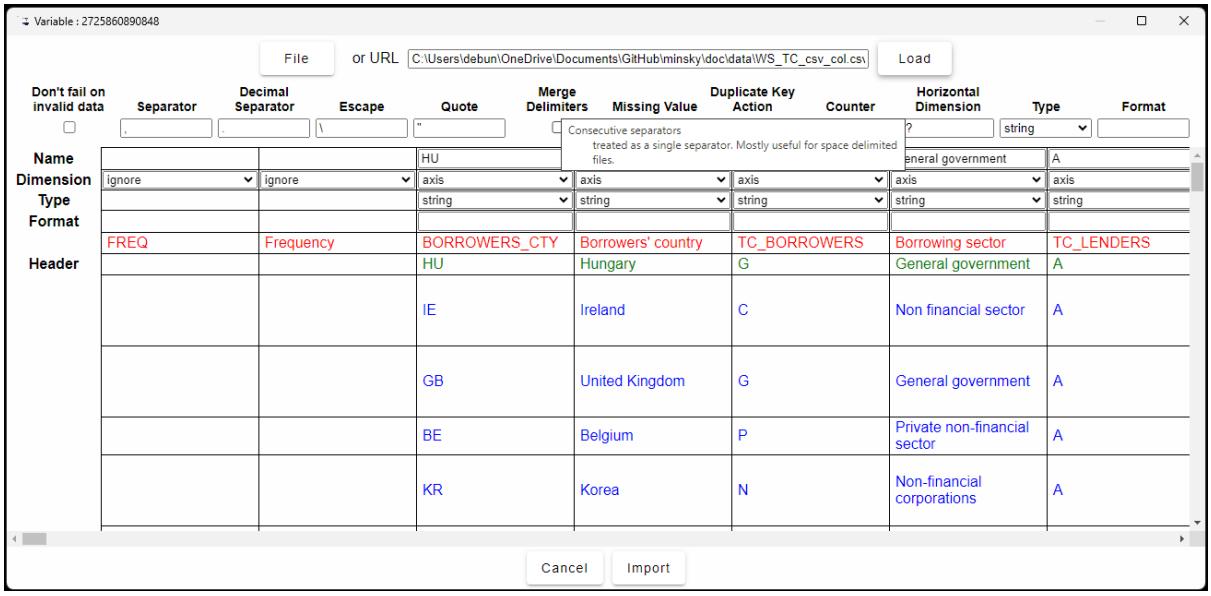


Figure 72: The data importing form

Ravel does some preliminary data analysis as it loads a file. Most of the time, *Ravel*'s interpretation of a file will be accurate, but sometimes it will, for example, identify the wrong row

as having the field labels for this data—as it has done here. Notice that it has identified the first data entry in the *BORROWERS_CTY* field as the name of the field (HU, the code for Hungary).

As you might expect by now, there are several right-click menu items that help you correct these errors. Right click on the row where *BORROWERS_CTY* appears in the file, and the first option is "Set as header row".

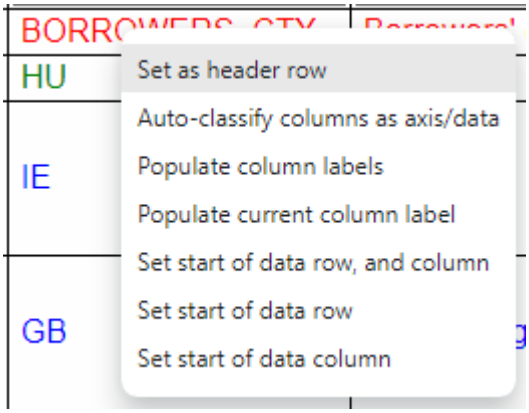
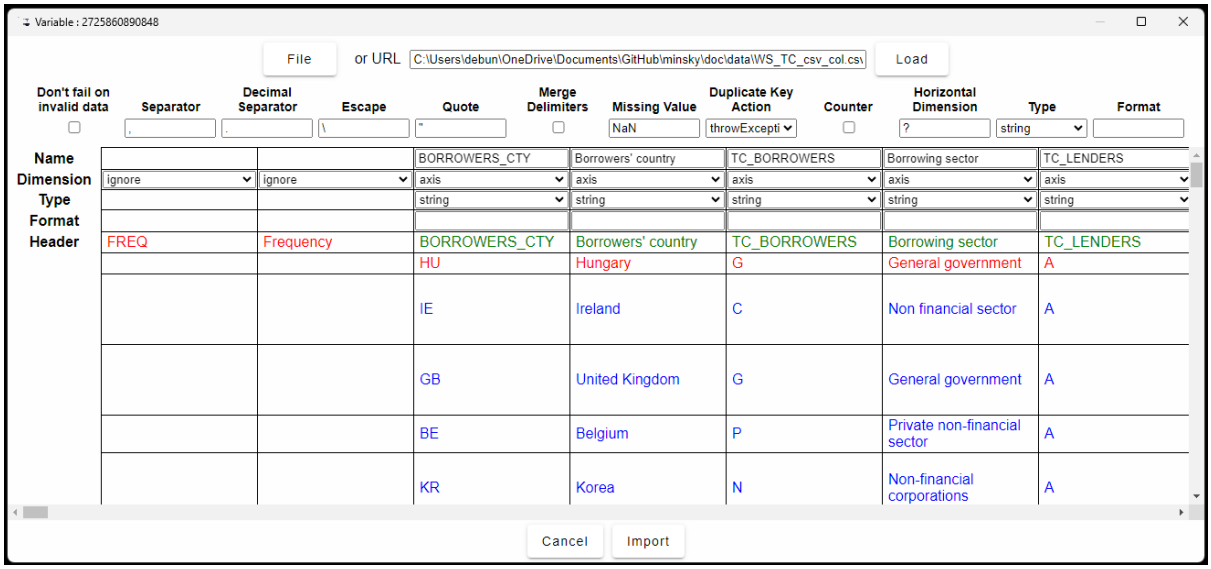


Figure 73: Right-mouse button menu items in the data importing form

The colour of the row will change from red to green; if you now follow up with "Populate column labels", all entries for that row will become the Name of the relevant column. There's still an error here—check the figure below to see if you can spot it—but for now we'll move on to the next stage of importing data, deciding which columns to convert into Dimensions, and which column (or columns) to identify as data.



Don't fail on invalid data	Separator	Decimal Separator	Escape	Quote	Merge Delimiters	Missing Value	Duplicate Key Action	Counter	Horizontal Dimension	Type	Format
	,	.	\	"		NaN	throwExcepti		?	string	

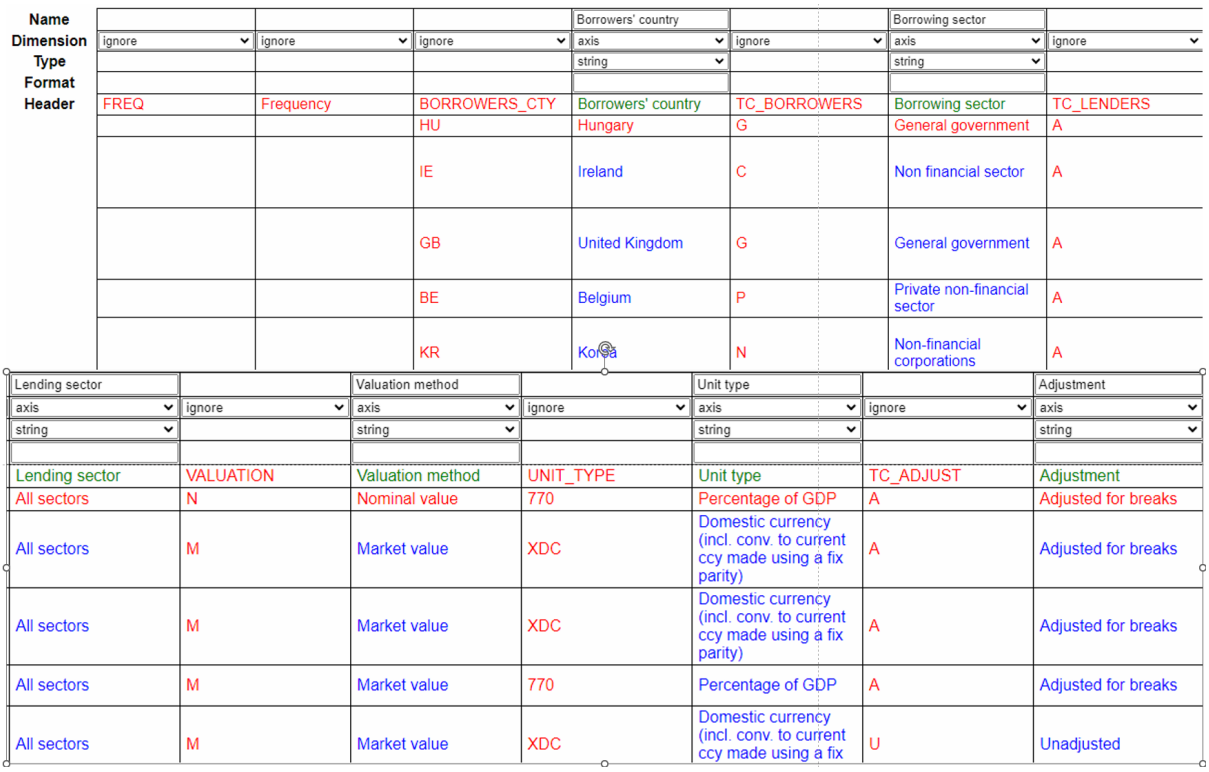
Name			BORROWERS_CTY	Borrowers' country	TC_BORROWERS	Borrowing sector	TC_LENDERS
Dimension	ignore	ignore	axis	axis	axis	axis	axis
Type			string	string	string	string	string
Format							
Header	FREQ	Frequency	BORROWERS_CTY	Borrowers' country	TC_BORROWERS	Borrowing sector	TC_LENDERS
			HU	Hungary	G	General government	A
			IE	Ireland	C	Non financial sector	A
			GB	United Kingdom	G	General government	A
			BE	Belgium	P	Private non-financial sector	A
			KR	Korea	N	Non-financial corporations	A

Figure 74: Steps in importing data

Since the first two columns were blank for quite a few rows, the import form guessed that they should be ignored, while it has interpreted the other fields visible here as axes. We know that the *BORROWERS_CTY* and *Borrowers' country* fields are mirror images of each other, so one should be ignored.

There are many such fields in this database, so you have to choose "ignore" for one and axis for the "other" multiple times. The next figure shows the end result of this process: the blue-

coloured columns will be imported as Dimensions, while the data in the red-coloured columns will be ignored.



Name				Borrowers' country		Borrowing sector	
Dimension	ignore	ignore	ignore	axis	ignore	axis	ignore
Type				string		string	
Format							
Header	FREQ	Frequency	BORROWERS_CTY	Borrowers' country	TC_BORROWERS	Borrowing sector	TC_LENDERS
			HU	Hungary	G	General government	A
			IE	Ireland	C	Non financial sector	A
			GB	United Kingdom	G	General government	A
			BE	Belgium	P	Private non-financial sector	A
			KR	Korea	N	Non-financial corporations	A

Lending sector		Valuation method		Unit type		Adjustment
axis	ignore	axis	ignore	axis	ignore	axis
string		string		string		string
Lending sector	VALUATION	Valuation method	UNIT_TYPE	Unit type	TC_ADJUST	Adjustment
All sectors	N	Nominal value	770	Percentage of GDP	A	Adjusted for breaks
All sectors	M	Market value	XDC	Domestic currency (incl. conv. to current ccy made using a fix parity)	A	Adjusted for breaks
All sectors	M	Market value	XDC	Domestic currency (incl. conv. to current ccy made using a fix parity)	A	Adjusted for breaks
All sectors	M	Market value	770	Percentage of GDP	A	Adjusted for breaks
All sectors	M	Market value	XDC	Domestic currency (incl. conv. to current ccy made using a fix	U	Unadjusted

Figure 75: Steps in importing data

Having identified which columns to ignore and which ones to treat as axes (or Dimensions), the final step is to specify which columns contain data. Often that will be a single column, but sometimes—as in this case—there will be multiple columns.

Ravel calls these the "Horizontal Dimension". Often, they will be unrelated to each other, but in this case, the Horizontal Dimension is the time dimension of this data, which is Quarterly, starting with 1940-Q2. In this import, we named the Horizontal Dimension "Date", specified that it was a time dimension, and specified its format as 1999-Q4 from the top-down menu on the format field: the year with 4 digits, followed by a dash – and the letter Q, and then the number of the quarter.

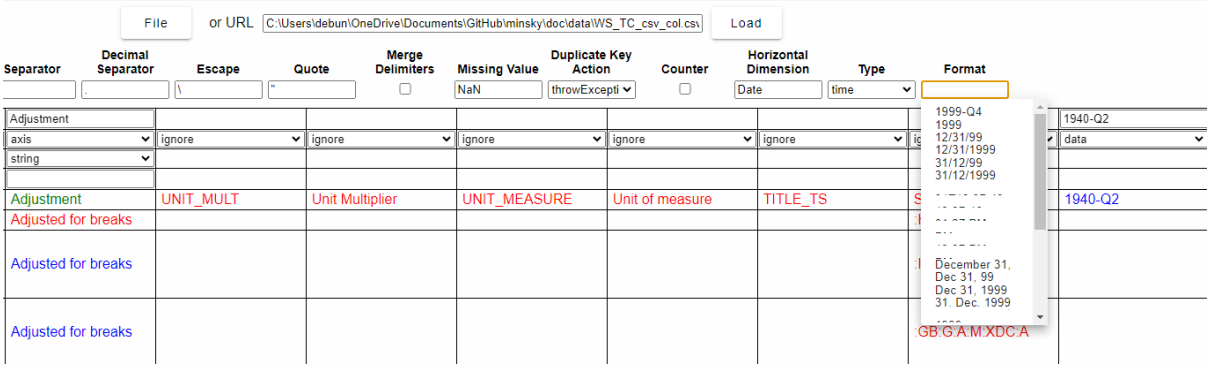


Figure 76: Steps in importing data

There is just one more thing to fix up here, and that's indicated by the first row of the data being red—see "Adjusted for breaks" in the first column. To show that that row is in fact the first row of

data, and that the date data starts where 1940-Q2, right click on that line in the cell below 1940-Q2 and choose “Set start of data row, and column”.

We are now ready to import the data, so click on “Import”. If this process succeeds (we’ll cover what to do if it doesn’t succeed shortly), the form will close, and the parameter dataImport will be renamed to that of the file being imported—in this case, WS_TC_csv_col, which Ravel formats as $WS_TC_csv_col because of LaTeX’s formatting rules:

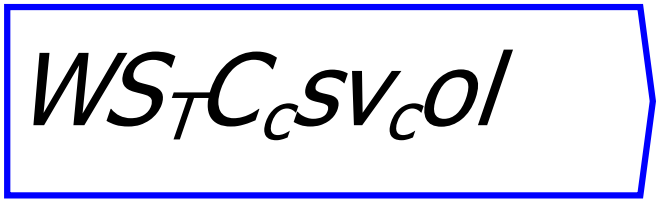


Figure 77: A Ravel data-storing parameter

Aggregating data on import

The BIS file has a unique record per row: statisticians have already aggregated every recorded loan in each quarter into a unique number for that quarter. But if you are loading a transactional database, like the sales data shown in Figure 3 and reproduced here in Figure 78, then you will want to aggregate sales for each day when you import.

Date	Salesperson	Source	Suburb	Quote	Discount	Price
06/04/2024	Ford Prefect	Bartercard	Wyoming	8967.21	876.3	8090.91
07/04/2024	Slartibartfast	Business Referral	Not Available	227.27	0	227.27
09/04/2024	Arthur Dent	Business Referral	Bateau Bay	3766.6	174.84	3591.76
10/04/2024	Ford Prefect	Business Referral	Gosford	1285.57	0	1285.57
11/04/2024	Arthur Dent	Business Referral	Wyoming	568.9	0	568.9
12/04/2024	Ford Prefect	Business Referral	Bateau Bay	300.98	0	300.98
12/04/2024	Arthur Dent	Business Referral	Yattalunga	470.35	0	470.35
14/04/2024	Arthur Dent	Business Referral	Yattalunga	3996.35	178.17	3818.18
30/04/2024	Ford Prefect	Business Referral	Bensville	1902.5	0	1902.5
02/04/2024	Arthur Dent	Car Signage	Narara	6733.2	641.38	6091.82
02/04/2024	Ford Prefect	Car Signage	Wamberal	2508.6	190.42	2318.18
04/04/2024	Trillian	Car Signage	Wyoming	309.09	0	309.09
09/04/2024	Ford Prefect	Car Signage	Wyoming	2752	161.09	2590.91
11/04/2024	Ford Prefect	Car Signage	Wyoming	6881.2	381.05	6500.15
15/04/2024	Arthur Dent	Car Signage	Narara	4881.4	154.13	4727.27
19/04/2024	Ford Prefect	Car Signage	Booker Bay	2722.76	86.4	2636.36
19/04/2024	Arthur Dent	Car Signage	Not Available	1720.55	0	1720.55
24/04/2024	Ford Prefect	Car Signage	Chain Valley Bay	3289.72	585.63	2704.09
01/04/2024	Ford Prefect	Drive/Walking Past	Aberglasslyn	1455.36	0	1455.36

Figure 78: Transactional data has several entries per day

This is handled by altering the default “Duplicate Key Action” from “throw Exception” to “sum”, as shown in Figure 79. This will then add up any entries with duplicated entries for the data’s four dimensions.

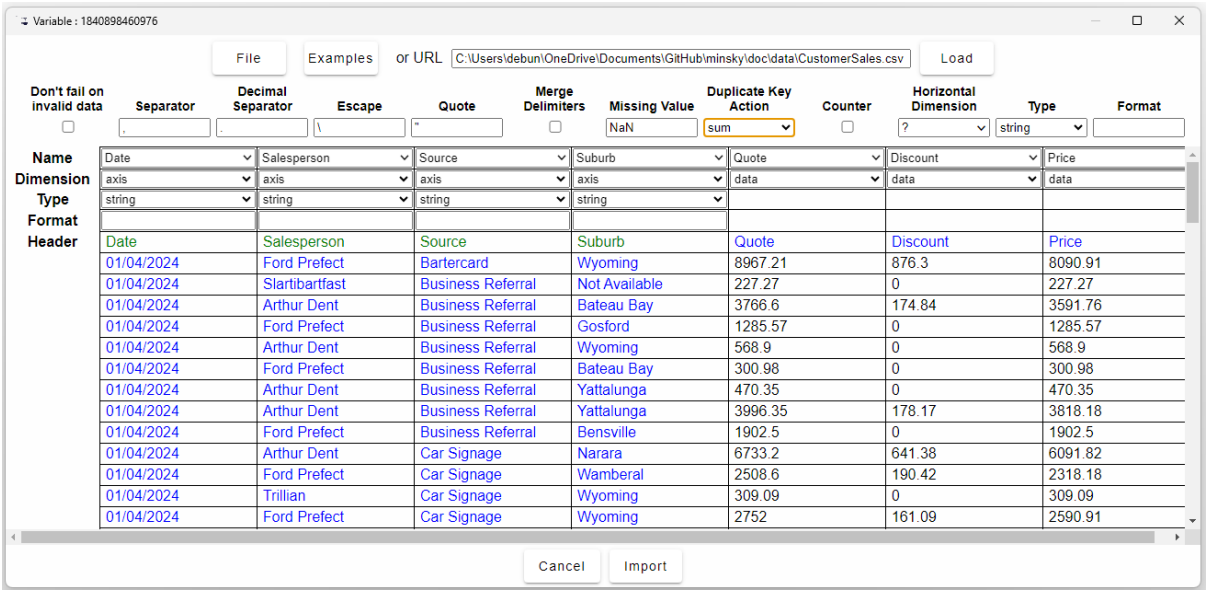


Figure 79: Aggregating data on import using "sum" as the action to perform if there are duplicate keys

Other options include returning the product of duplicated entries, the minimum, maximum, and average.

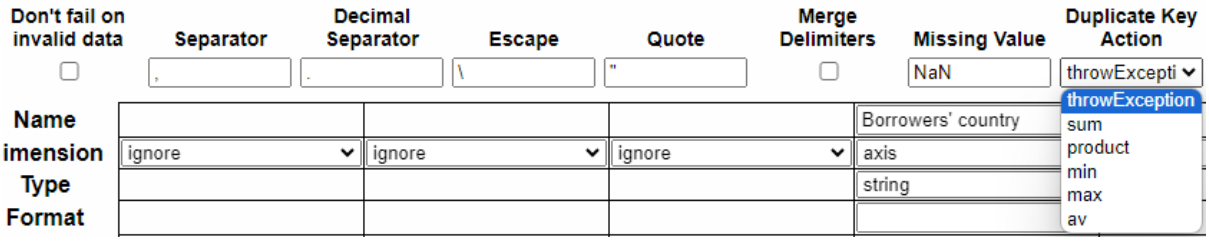


Figure 80: Control options on the data importing form

Data Reduction with Ravel

To analyse your newly-loaded data, you have to connect this parameter to a Ravel. Once you do this, you will see something like the figure below (this Ravel has 7 dimensions, and at present *Ravel* hides the axis names with more than 6 dimensions; to see the name, you hover the mouse over an axis).

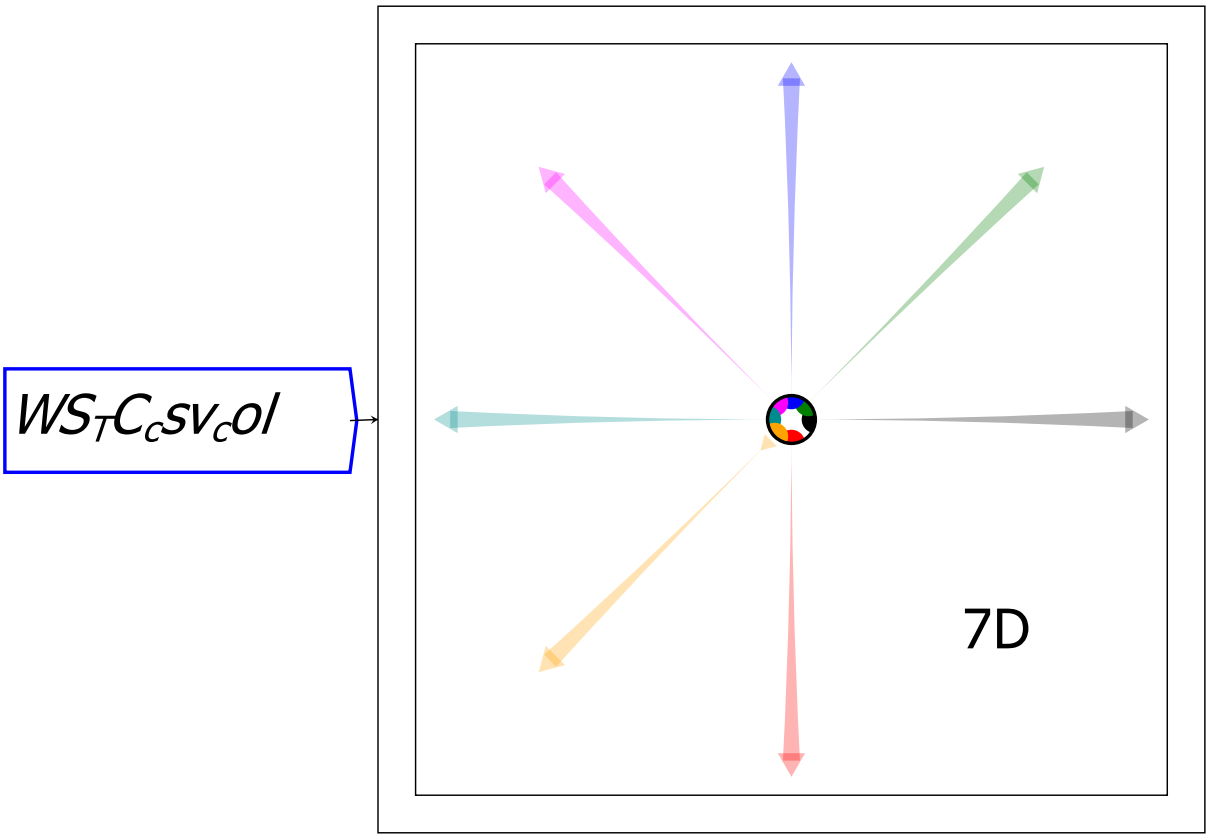


Figure 81: Attaching a data-storing parameter to a Ravel

At this point, the data may need further reduction. Sometimes this will be because something unimportant is coded that was still necessary to get a unique key for each element in the Ravel—like, for example, whether the data was collected by a government agency or a private body. Other times it will be because data on an axis needs to be aggregated to get a complete file for analysis.

The latter is the case for this Ravel. There is a dimension called *Valuation method* which has two entries, Market Value and Nominal Value—the former being the market price of debt and bonds etc., the latter being the face value of bonds. Every country in the database reports government debt data in Nominal Value terms—*except South Korea*. So, to include South Korean government debt in the analysis, we collapse this axis with the aggregation set to Sum (Max or Min would also have worked). There is also data which has not been adjusted for structural breaks, and data which has. To use just adjusted data, we move the selector dot to "Adjusted for breaks" on this axis. We also set Lending Sector to "All", since this matches data previously published by the US Federal Reserve. Ravel then outputs a 4D object, as shown in the next figure.

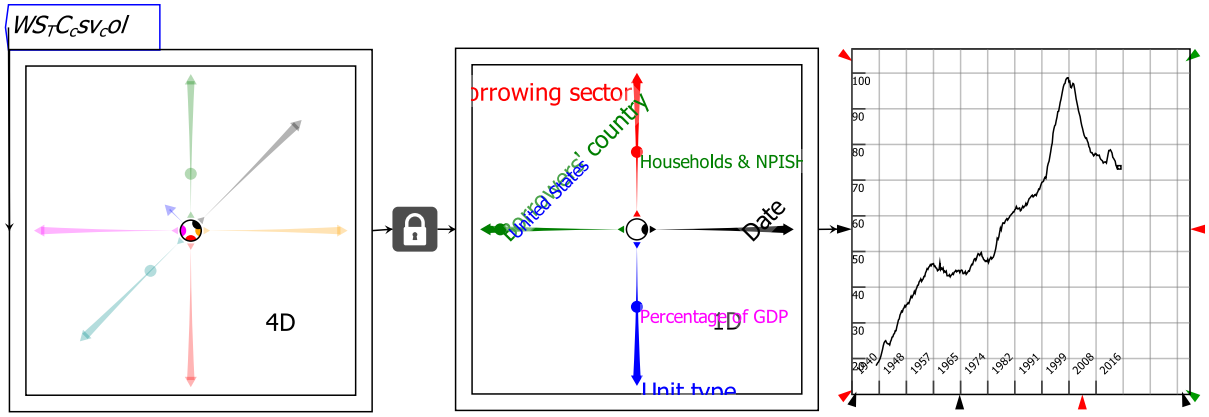


Figure 82: Reducing unneeded detail using Ravel

If Data Importing Fails

It is possible that a CSV file may contain flaws that prevent data being loaded into Ravel. This can include non-numeric data in fields that are supposed to control only numbers, date formats that alter within a date field, and extraneous information stored at the end (or sometimes the beginning) of a CSV file, and so on...

Ravel can identify problematic rows and columns in a CSV vile, and produce an error-report version of that file for examination in a text or spreadsheet editor. It is also possible to import valid data from a file and exclude invalid data, by ticking on the "Don't fail on invalid data" option on the Import form. That will load valid data and exclude invalid data.

Miscellaneous Features

Resizing objects in *Ravel*

All *Ravel* objects have either a single arrow, or an arrow on each corner, which can be used to resize the object. Click and drag on any object to alter its size.

For More Information

Ravel has a context-sensitive help system. For help on any aspect of *Ravel*, hover over the relevant operator and choose Help from the context menu.

Acquiring *Ravel*

Currently *Ravel* is being distributed via the Patreon page www.patreon.com/Ravelation. If you sign up to this page for A$10 per month (roughly US$6.50), you will have access to posts containing new releases of Ravel. Figure 83 shows the pre-release version of this process, using the Patreon page for Minsky, the Open-Source system dynamics program on which *Ravel* is built.

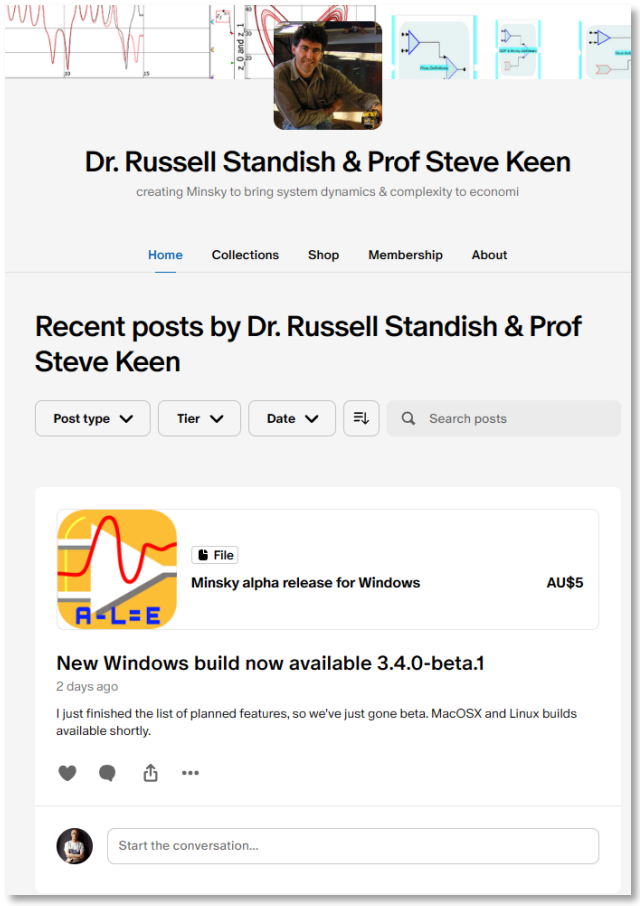


Figure 83: The Ravel installation process starts with joining the Ravel page and downloading from Patreon

Once you have signed up to the Patreon page, you will have access to posts there that release new versions. Figure 84 shows one such post.

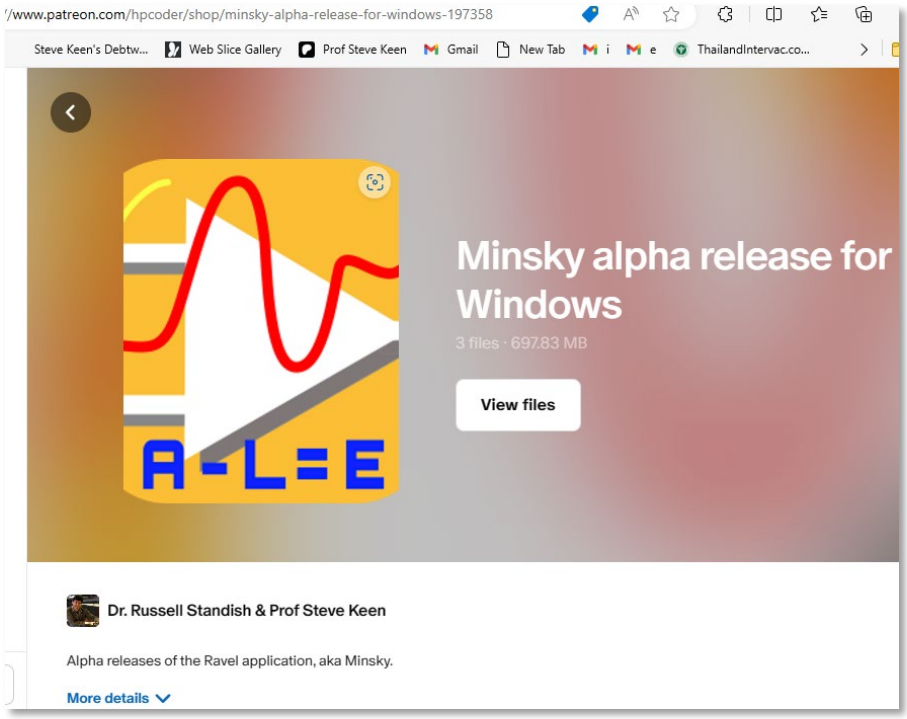


Figure 84: A post containing the latest releases of Ravel

Click on "View Files" and the recent releases of Ravel will appear in chronological order—see Figure 85

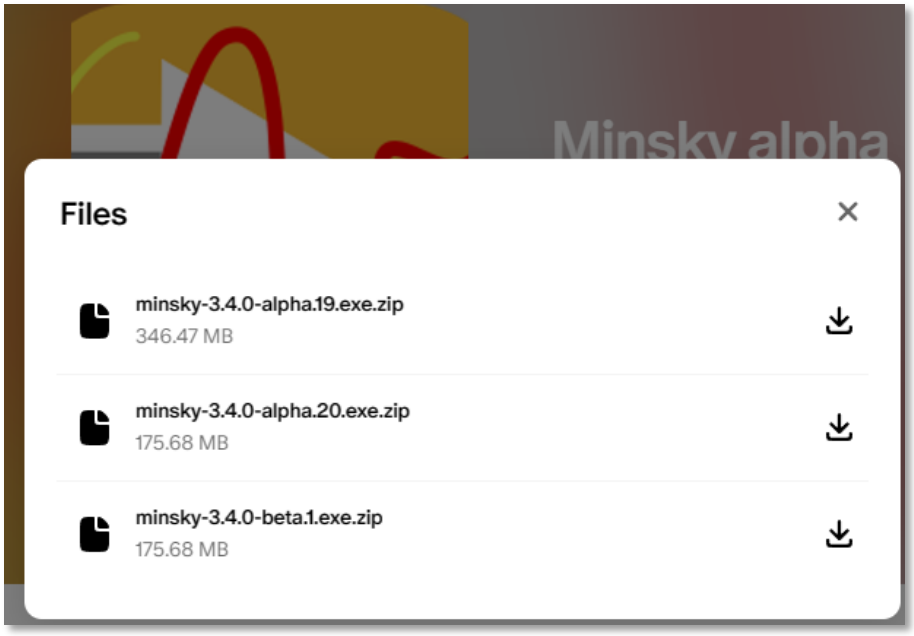


Figure 85: The most recent releases of Minsky/Ravel

Click on the final version shown—minsky-3.3.0-beta.1.exe.zip in this case—and it will be downloaded to your computer. Install it from there and you will have Minsky, but not yet have *Ravel*.

You acquire *Ravel* using "Upgrade" from the File menu. That will bring up the *RavelUpgrader* form shown in Figure 86.

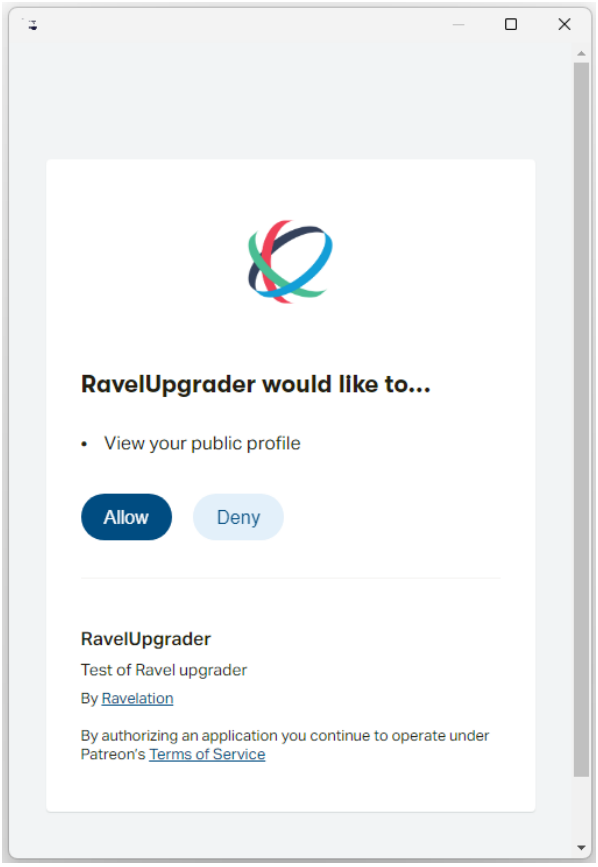


Figure 86: The Ravel upgrade form

Click on "Allow", and the program will confirm that you are a current subscriber, and download and install *Ravel* into your copy of *Minsky*. Once finished, the program will close and restart, and you will have the latest version of *Ravel*.

We expect to release a new version of *Ravel* at least once a month, and as often as once a week. This process will slow down when we release Ravel on its own commercial website towards the end of 2024.