Formatting Financial Data with Google Refine

Many problems in formatting financial data can be time-consuming to solve. Luckily, there is a very powerful tool - *Google Refine* - which can drastically speed up data cleansing. This tutorial shows you how to use Google Refine to clear up key issues with the data. Following the steps outlined here will help you to easily transform data according to the data formatting guidelines.

Getting set up

Install Refine

First step is to install Google Refine by following the instructions here:

http://code.google.com/p/google-refine/wiki/Downloads

Google Refine will be an application on your computer that will open up and run in your web browser, so you will need to be connected to the internet to use it.

Note: Many users who have Windows as an operating system run Google Refine directly from a Zip File. We strongly advise against this and would suggest extracting the application into a dedicated directory.

Create Project

Now you need to upload your data to Refine. After you start Refine, it will open a browser and present its landing page. If the page does not appear automatically, you can try to open it by accessing http://localhost:3333 in your browser. On the landing page, create a project, choose files and hit '*Next*'.

Before you start editing there are a couple of changes we suggest you make to your data:

- Character encoding - select either "ISO 8859-1" or 'UTF-8" - this ensures any special characters or diacritics will be displayed correctly.
- Parse cell text into numbers, dates, … - we suggest de-selecting this option as it can often cause errors to occur (e.g. confusion between American and British date formats). Remember, the details of how dates should be formatted are contained in the Data Formatting Guide.

Tricks and Tips

Google Refine is a spreadsheet editor built for bulk data analysis and processing. It takes a bit of getting used to and unfortunately does not share many commands with familiar programmes such as Excel, however, certain elements are very simple to use.

Getting Familiar with Facets and Filters

You will use both facets and filters very often in Refine. Try creating a text facet to understand what they do:

Click on the dropdown arrow in the column header > Facet > Text Facet

You will see a box appear which groups all identical cell contents and provides a count for the number of times they appear in that column in your dataset. This is useful for several reasons:

- Spotting typos - for example, creating a text facet gives you an overview of all the unique cells in a column. This means you can easily scroll through them to review. E.g. it could show that some of your cells contain “Rroceeds from global taxes” rather than “Proceeds from global taxes” - you want to correct the former, so click edit by the facet result and edit it directly. This changes all of the cells with typos.
- Spotting blank columns. Think a column is blank and preparing to delete it? Check quickly that there is nothing in it by performing a text facet. If the column is empty, you should get only one result (blank). You can then delete this column by clicking the *Dropdown > Edit Column > Remove this Column*.

Note: Facets only work up to a few thousand unique entries, so if you have a very large dataset and want to find specific values in a column with many distinct values, it may be best to use a filter to search for that item individually. Select this from the column dropdown menu as before.

Fill Down

It is not uncommon for data to be produced in a way which is easy for humans to read, but not for machines. Look at the ‘Head-Account’ Column in the example below:

	Column	Head-account	Column2	Sub-account	Column3	Sub-account Description	2008 Budget	2008 Actual	2009 Budget	2009 Actual	Column4
1.				110.101		Reserves for overheads	2,000,000	-	-	175,000	REVENUE
2.		710.xxx	Fiscal revenue (1)	Fiscal revenue (1)							REVENUE
3.				710.1		Rroceeds from global taxes	6,847,200	3,127,600	6,847,200	6,293,198	REVENUE
4.				710.101		Proceeds from business	4,166,100	1,528,655	4,166,100	4,733,829	REVENUE
5.				710.102		Liquor license	60,750	-	60,750	20,500	REVENUE
6.				710.103		Cattle taxes	3,118,600	1,028,100	5,118,600	1,232,690	REVENUE
7.				710.107		Sanitary taxes	9,647,875	5,006,000	11,647,875	6,391,200	REVENUE
8.				710.109		Land taxes	100,000	-	100,000	-	REVENUE

We can see that there is a relationship between the Head-account column and Sub-account. The head account value which is present in the second row should also ‘fill-down’ numerous rows, as all the sub-accounts fall under this category. Google Refine has a tool to copy the results of a cell down until it meets another entry, in this case, the next value for head account.

Dropdown > Edit Cells > Fill Down.

Check your results have provided the right results by performing some text facets. In this case, you will see that the cell in the top row is blank as there was nothing to fill down from, this will have to be corrected manually.

Result:

All		Column	Head-account	Column2	Sub-account	Column3	Sub-account De	2008 Budget	2008 Actual
1.			110.xxx		110.101		Reserves for overheads	2,000,000	- edit
2.			710.xxx	Fiscal revenue (1)	Fiscal revenue (1)				
3.			710.xxx		710.100		Rroceeds from global taxes	6,847,200	3,127,600
4.			710.xxx		710.101		Proceeds from business	4,166,100	1,528,655
5.			710.xxx		710.102		Liquor license	60,750	-
6.			710.xxx		710.103		Cattle taxes	3,118,600	1,028,100
7.			710.xxx		710.107		Sanitary taxes	9,647,875	5,006,000
8.			710.xxx		710.109		Land taxes	100,000	-
9.			710.xxx		710.110		Other fiscal revenues	2,000,000	513,514
10.			711.xxx	Council additional taxes for levies	711.100		Council additional taxes for levies	40,000,000	24,737,422

Delete Empty Columns

Check ‘Getting Familiar with Facets and Filters’ for techniques to show a column is genuinely

empty, then:

- *Dropdown > Edit Column > Remove this Column*.

Rename Columns

- *Dropdown > Edit Column > Rename this Column*

Removing Pseudo Rows

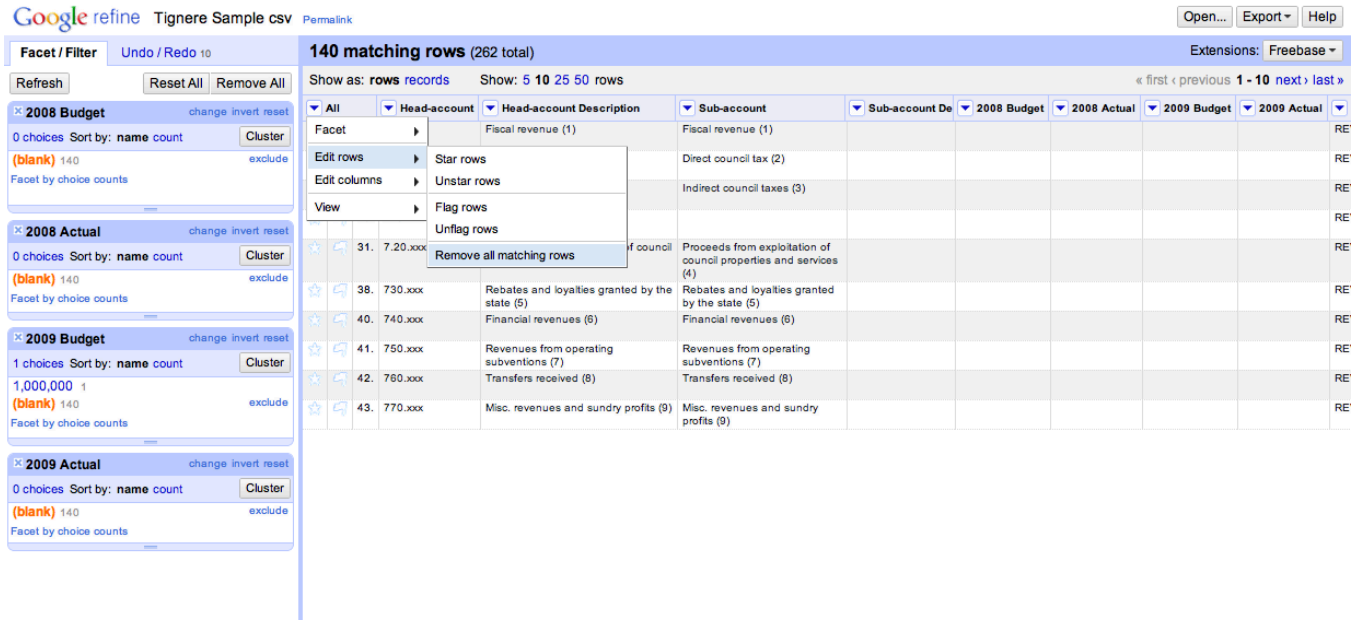
You will notice that some of the rows in the data do not actually contain any data. See row 2 in the example below which contains no data for budgeted / actual amounts for either 2008 or 2009:

All		Head-account	Head-account D	Sub-account	Sub-account De	2008 Budget	2008 Actual	2009 Budget	2009 Actual	Revenue/Expen	Recurrent
1.		110.xxx	Reserves for overheads	110.101	Reserves for overheads	2,000,000	-	-	175,000	REVENUE	RECURRENT
2.		710.xxx	Fiscal revenue (1)	Fiscal revenue (1)						REVENUE	RECURRENT
3.		710.xxx	Fiscal revenue (1)	710.100	Rroceeds from global taxes	6,847,200	3,127,600	6,847,200	6,293,198	REVENUE	RECURRENT
4.		710.xxx	Fiscal revenue (1)	710.101	Proceeds from business	4,166,100	1,528,655	4,166,100	4,733,829	REVENUE	RECURRENT
5.		710.xxx	Fiscal revenue (1)	710.102	Liquor license	60,750	-	60,750	20,500	REVENUE	RECURRENT
6.		710.xxx	Fiscal revenue (1)	710.103	Cattle taxes	3,118,600	1,028,100	5,118,600	1,232,690	REVENUE	RECURRENT
7.		710.xxx	Fiscal revenue (1)	710.107	Sanitary taxes	9,647,875	5,006,000	11,647,875	6,391,200	REVENUE	RECURRENT
8.		710.xxx	Fiscal revenue (1)	710.109	Land taxes	100,000	-	100,000	-	REVENUE	RECURRENT
9.		710.xxx	Fiscal revenue (1)	710.110	Other fiscal revenues	2,000,000	513,514	1,500,000	160,390	REVENUE	RECURRENT
10.		711.xxx	Council additional taxes for levies	711.100	Council additional taxes for levies	40,000,000	24,737,422	35,000,000	27,341,392	REVENUE	RECURRENT

This is because it was simply a placeholder row in the original document. There are many like this in the data. To find these, we perform a text facet on the four columns 2008 Budget, 2008 Actual, 2009 Budget and 2009 Actual and in each one, select only the blank cells.

When you are done, go to the dropdown in the *All* column:

Dropdown > Edit rows > Remove all matching rows

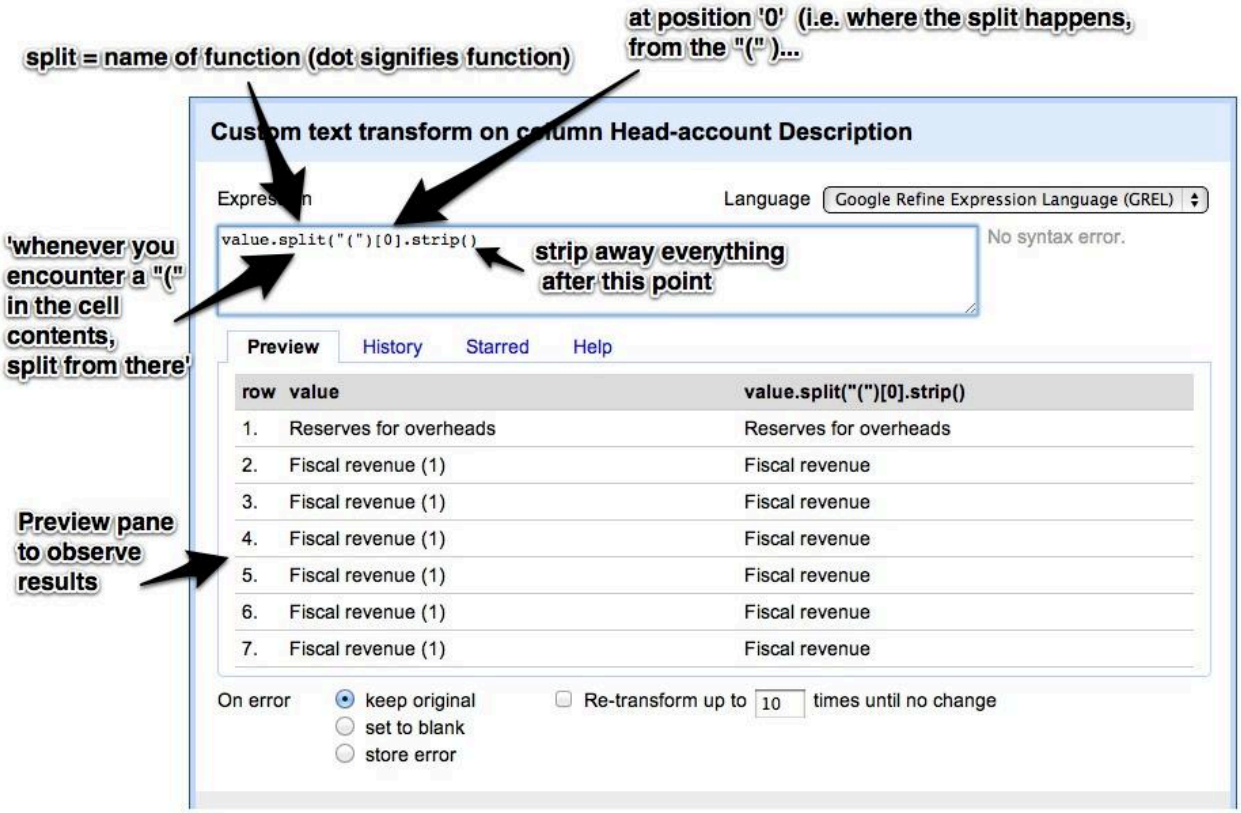


Removing numbers in brackets

You will notice that in the Head Account and Sub-Account columns, a number appears after the Description. If these do not add any additional value over the head-account description, you can remove them to tidy up the appearance.

Dropdown > Edit cells > Transform

You will be taken to a screen which will ask you to input some functions in Google Refine code. You can refer to the *Help* section of the dialogue box for more functions and transformations, we cover just the necessary here:



Don't worry if you don't understand exactly what is going on here, if you are just trying to tackle exactly the same issue as here, you can simply copy the code here.

Transposing columns

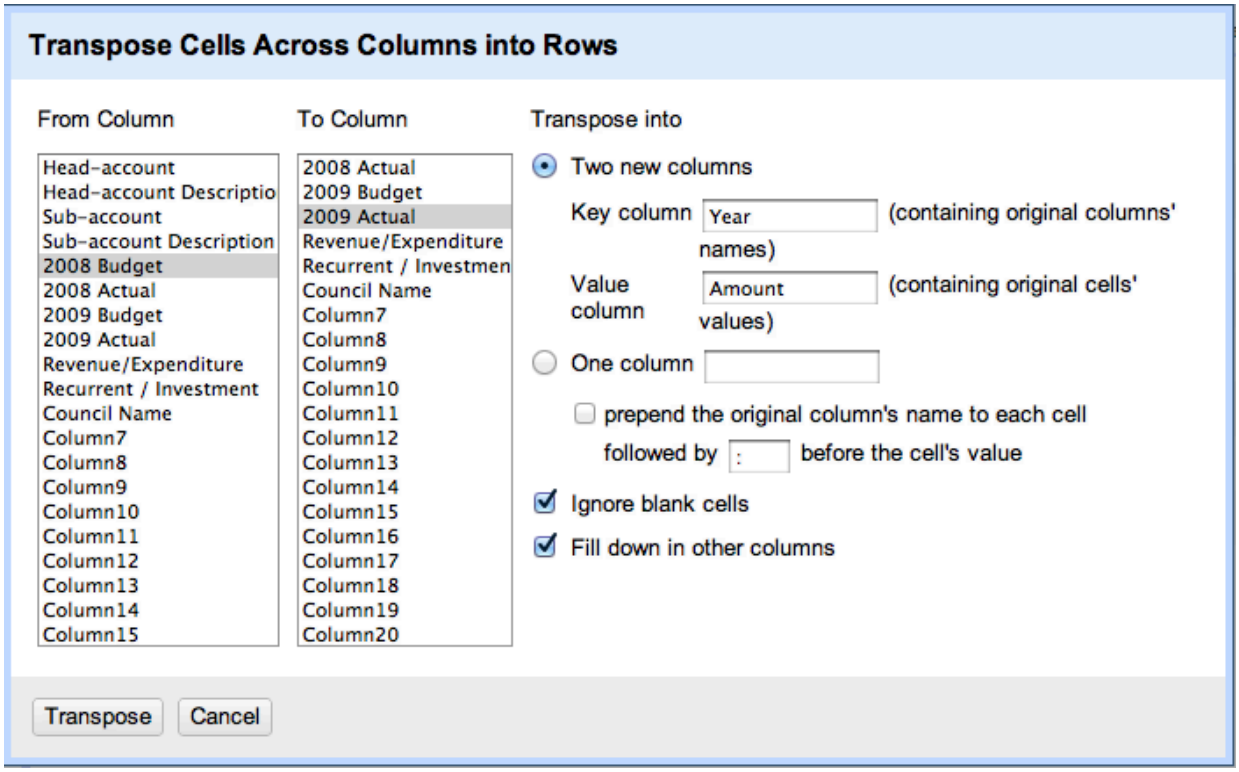
As you will remember from the documentation on how to format your data, one row must contain *one logical piece of information*. As you can see from this data, we have 4 columns which correspond to *time*

2008 Budget	**2008 Actual**	**2009 Budget**	**2009 Actual**

This is additionally complicated as each of these column headers contains multiple pieces of information - budget vs actual spending. We also need to split these elements out.

Step 1 - Transpose

Dropdown > Transpose
You will then be presented with a dialogue box which will look something like this:



1. In the *From Column* and *To Column* selectors, you need to select the range of the columns you would like to transpose (literally - flip by 90 degrees). *From* is the furthest column left of the range you are selecting, *To* is the furthest right.
2. In this case you are interested in two new columns, one which will contain what were previously the contents of the header row and another to contain the contents of the cells (the amount). Put a name to describe the original column headers in the *Key column* field, and a name to describe the original cell contents in the *Value column* field.
3. You should also select '*Fill down in other columns*' to ensure that the data from the existing rows is correctly replicated down the table.

As a result, you should end up with something like this:

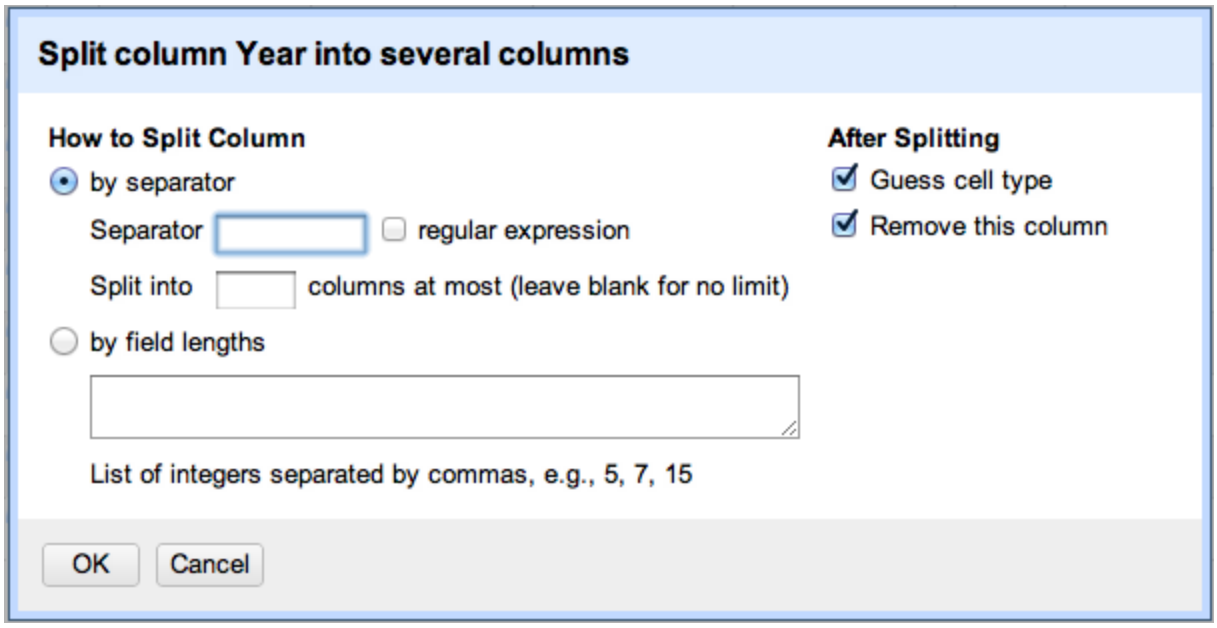
All		Head-account	Head-account D	Sub-account	Sub-account De	Year	Amount	Revenue/Expenc
1.	110.xxx	Reserves for overheads	110.101	Reserves for overheads	2008 Budget	2,000,000	REVENUE
2.	110.xxx	Reserves for overheads	110.101	Reserves for overheads	2008 Actual	-	REVENUE
3.	110.xxx	Reserves for overheads	110.101	Reserves for overheads	2009 Budget	-	REVENUE
4.	110.xxx	Reserves for overheads	110.101	Reserves for overheads	2009 Actual	175,000	REVENUE
5.	710.xxx	Fiscal revenue	710.100	Rroceeds from global taxes	2008 Budget	6,847,200	REVENUE
6.	710.xxx	Fiscal revenue	710.100	Rroceeds from global taxes	2008 Actual	3,127,600	REVENUE

As you can see, the cells in the year column still contain information for both a) the year and b) whether the amount is budgeted or actual, so we need to split these out.

Splitting one column into multiple columns

Dropdown > Edit column > Split into several columns

In the Dialogue box, you will be asked where the column should split, i.e. what the separator is. The column cell contents are all of the format: [YYYY][Space][Budget/Actual], so you want to simply hit the spacebar to enter that as the character you want to separate by (obviously, you can't see that here).



You may choose to deselect '*Guess Cell Type*' but it's not critical here. Then hit *OK*.

You will end up with something which looks like this. You will need to rename the new column.

All		Head-account	Head-account D	Sub-account	Sub-account De	Year 1	Year 2	Amount
1.		110.xxx	Reserves for overheads	110.101	Reserves for overheads	2008	Budget	2,000,000
2.		110.xxx	Reserves for overheads	110.101	Reserves for overheads	2008	Actual	-
3.		110.xxx	Reserves for overheads	110.101	Reserves for overheads	2009	Budget	-
4.		110.xxx	Reserves for overheads	110.101	Reserves for overheads	2009	Actual	175,000
5.		710.xxx	Fiscal revenue	710.100	Rroceeds from global taxes	2008	Budget	6,847,200
6.		710.xxx	Fiscal revenue	710.100	Rroceeds from global taxes	2008	Actual	3,127,600
7.		710.xxx	Fiscal revenue	710.100	Rroceeds from global taxes	2009	Budget	6,847,200

Again, it is a good idea to run a text facet over the new cells just to check that nothing has gone wrong.

Removing blank cells

As you can see - some cells still contain dashes as there is no data for that year. OpenSpending will not accept these, so they must be removed.

Simply filter the column for dashes:

Dropdown > Text Filter

Enter your search term. This will bring up all of the empty columns. Remove them by selecting the dropdown in the *All* column.

Dropdown > Edit rows > Remove all matching rows

Clear your filter ([x]) and you will see your cleaned data.

Removing commas in numbers

OpenSpending requires numbers to not have any delimiters besides a dot to designate decimals and optionally a minus sign. In many datasets, however commas or spaces are present in numbers as separators. With Google Refine, these are easy to remove. This uses an approach very similar to *'Removing numbers in brackets'* however, where before, the command was *split*, here the function we run is *replace*.

Dropdown > Edit cells > Transform

In the input screen:

```
value.replace(",","")
```

The contents of the first double quotation marks tell the program what is being replaced, while the contents of the second tell it what to replace it with.

Stripping Whitespace

The final step will not show any results which are immediately visible to the human eye in Refine, however it is important to strip off any remaining spaces from the ends of cells. Here's why:

OpenSpending groups identical items and produces aggregates, so:

Fiscal revenue

Fiscal revenue[space]

could be perceived as different things and grouped separately. We remove the whitespace on all of the columns as a precaution.

Dropdown > Edit cells > Common transforms > Trim leading and trailing whitespace

Results

When you've finished, you should end up with something like this:

All		Head-account	Head-account D	Sub-account	Sub-account De	Year 1	Year 2	Amount	Revenue/Expen	Recurrent / Inve	Council Name
1.		110.xxx	Reserves for overheads	110.101	Reserves for overheads	2008	Budget	2000000	REVENUE	RECURRENT	TIGNERE COUNCIL
2.		110.xxx	Reserves for overheads	110.101	Reserves for overheads	2009	Actual	175000	REVENUE	RECURRENT	TIGNERE COUNCIL
3.		710.xxx	Fiscal revenue	710.100	Rroceeds from global taxes	2008	Budget	6847200	REVENUE	RECURRENT	TIGNERE COUNCIL
4.		710.xxx	Fiscal revenue	710.100	Rroceeds from global taxes	2008	Actual	3127600	REVENUE	RECURRENT	TIGNERE COUNCIL
5.		710.xxx	Fiscal revenue	710.100	Rroceeds from global taxes	2009	Budget	6847200	REVENUE	RECURRENT	TIGNERE COUNCIL

Merging in data from another project/spreadsheet

One of the most powerful functions of Google Refine is the option to combine information from

multiple Refine Projects[1]. Given a shared set of values in a particular column, attributes from one table can be imported into the other.

Assume, for example, that one project contains information about investment projects and has a column “Chapter”, which contains only a numeric identifier for the budget chapter the investment was allocated under. At the same time, another project (we assume it’s called “Cameroon Budget Codes”, actual data for testing) may contain more information about each chapter, such as the full title (the column is named “en” in the source data) and its association to other classification schemes (named “Focus Sector Codes” in the data).

The merge function, called cross in Refine, is somewhat complex to use since it must be scripted as a command and the code involves both the concept of a cell and various rows. It’s documentation may give further guidance. To run it, open the “Chapter” column dropdown in your investment data and select “Edit column”, then “Add column based on this column...”. This function is also useful when applying general transformation where you want both the original and transformed data to remain available.

In the transformation code box, type the following:

```
cell.cross(“Cameroon Budget Codes”, “code_category”).cells[“en”].values[0]
```

This command, when executed, will pull in the chapter titles from the budget codes project. As may be obvious, the first argument to cross, “Cameroon Budget Codes” is the project name of the project from which we’ll pull in our data. The next argument “code_category” is the name of the column which contains the chapter codes in that project. Each value in the investment data “Chapter” column will thus be searched in the “code_category” column.

If a match is found, we will receive a reference to the row in which it occurred. This reference can be used to look up a specific column in the budget codes project: cells[“en”] will pick the value column called “en”. Finally, any such linkage may yield multiple results: a given chapter code may occur not once but many times in the project we’re crossing with. In this example, we’re using .values[0] to select the first match, irrespective of the total number of possible links (which will always be one, as the budget codes spreadsheet has no duplicates).

After verifying the result in the preview and adding a name for the new column, pressing “OK” will add the desired values. You can repeat this for each column you want to import, e.g. for the focus sector or COFOG codes.

Further Reading

[1] Users of Microsoft Excel might know a similar function called VLOOKUP, while users of relational databases will be familiar with the idea of a JOIN.

The operations explained in this tutorial are the most common actions needed when cleaning up data for OpenSpending. The software offers a much larger set of functions for data cleansing, so it is worthwhile to browse the documentation at:

http://code.google.com/p/google-refine/wiki/DocumentationForUsers

In particular, we recommend using the clustering functions on manually created and messy datasets as well as the web retrieval options to add further attributes to a table from an external source.