Intersect Australia Ltd
PO Box H58, Australia Square
Sydney NSW 1215

T +61 2 8079 2500
F +61 2 9262 3040

ABN 67 131 752 657

Google Refine for Humanities Datasets

An Introduction

Dr Luc Small | 22 February 2012 | 1.2

1 Overview

Google Refine is a powerful free tool for exploring, normalising and cleaning up datasets. In this tutorial we'll work through the various features of Refine, including importing data, faceting, clustering, and calling into remote APIs, by working on a fictional but plausible humanities research project. We'll start with a research question in mind and use the features of Refine to gain insights and find answers.

The research question relates to NSW police stations — finding out what we can about where they are located, their heritage status, and the kinds of archival records State Records NSW holds on them.

2 Resources

Google Refine:

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

Google Refine Documentation:

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

The Google Geolocation API:

- http://code.google.com/apis/gears/api_geolocation.html

Source of the original dataset as hosted by the NSW Office of Environment and Heritage:

- http://www.heritage.nsw.gov.au/07_subnav_04.cfm (then Basic Search for "police station")

The NSW State Records API:

- http://api.records.nsw.gov.au/

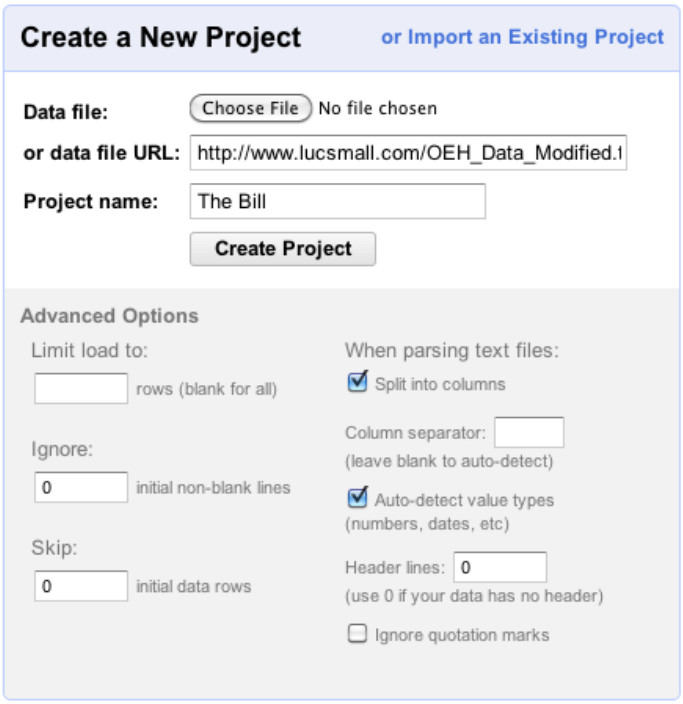
3 Installing Google Refine

To install Google Refine:

1. Go to the main Google Refine website:
   - http://code.google.com/p/google-refine/
2. Select the **Downloads** tab.
3. Choose the appropriate download for your operating system. Windows, Mac and Linux are all supported.
4. Follow the installation procedures for your operating system.

4 Starting a Project

1. Launch Google Refine. It will open in your default web browser. Note: Google Refine is not a cloud application; it runs locally, using your web browser as its primary interface.
2. Enter `http://bit.ly/zonJkP` (or, alternatively, `http://www.intersect.org.au/docs/OEH Data Modified.txt`) in the **Data file URL** field.
3. Set **Project name** to *`The Bill`*.
4. Set **Header lines** to *`0`*.
5. Click **Create Project**.

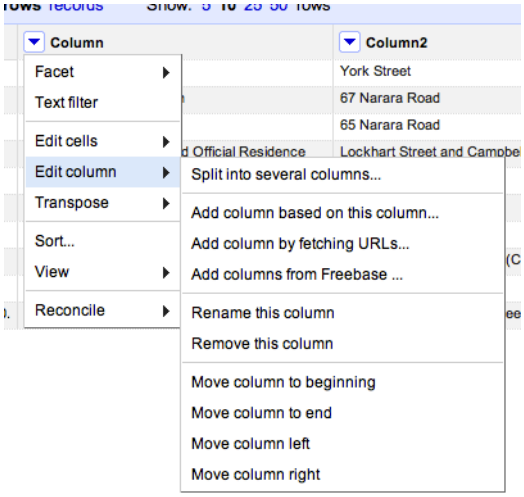


6. The project will open with 391 rows:

5 Getting Organised

Renaming columns

1. Select **Column menu > Edit column > Rename this column**.



2. Rename to *Station Name*. Click **OK**
3. Rename *Column2* to *Address* as above.
4. Rename *Column3* to *Suburb* as above.
5. Rename *Column4* to *LGA* as above.

Splitting columns

1. Select **Column5 menu > Edit column > Add column based on this column...**
2. Set **New column name** to *Heritage Listed*.
3. Set **Expression** to:

   ```
   if(value == "Yes", "Yes", "No")
   ```

4. Select **Column5 menu > Edit column > Add column based on this column...**
5. Set **New column name** to *Source*.
6. Set **Expression** to:

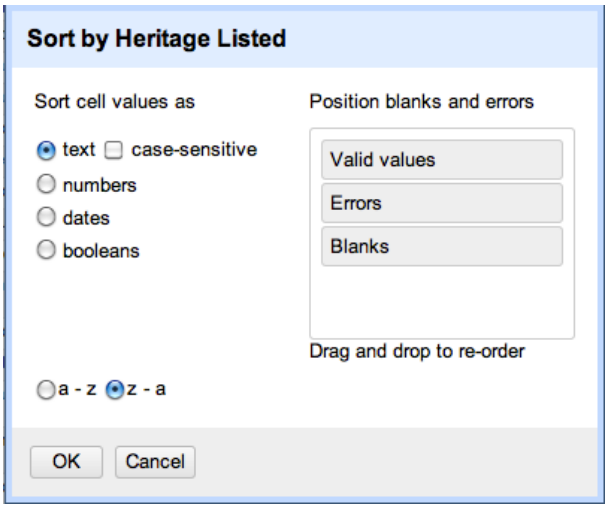
   ```
   if(value != "Yes", value, "")
   ```

7. Select **Column5 menu > Edit column > Remove this column**

6 Exploring the data

Sorting Columns

1. Select **Heritage Listed menu > Sort**
2. Select **text** and **z-a** and click **OK**. All the heritage-listed police stations will appear first.



Facet and cluster on Suburbs

1. Select **Suburb menu > Facet > Text Facet**. Note that a **Suburb** facet will appear on the left hand side of the screen. This shows a list of unique suburbs in the data.
2. Click on **265 choices**. A text box will appear so we can copy and paste our list of unique suburbs into, say, a document.
3. Click **count** to order to list the most frequently occurring suburbs first.
4. Click **Cluster** to reveal and fix some consistency issues with the dataset. Select, for instance **nearest neighbor** as the method. You'll see that Refine finds some near matches. Now try some of the other methods.
5. You can make your data more consistent by typing the correct value into **New Cell Value** and ensuring the **Merge?** checkbox is selected. Use the **Merge Selected & Re-Cluster** function to actually modify the dataset.

Find duplicate addresses

1. Take the same approach as above to create a text facet on the **Address** column.
2. Sort by **count**.
3. You can click on any given address to view only the records matching the address. For instance, click on *281 Clarence Street*.
4. Click on **Reset All** to restore the listing to display all 391 rows.

7 Undo/Redo History

Google Refine has an infinite undo history. To access:

1. Click on the **Undo/Redo** tab. You'll see every action you've done since creating the project.
2. You can undo to any step by clicking on the step you want to revert back to. Similarly, you can redo every step.

8 Calling into an API

It's relatively straightforward to draw in data from an external API with Google Refine. In this case we'll call in to Google's Geolocation API to get the longitude and latitude of all our police stations.

1. Ensure all 391 rows are displayed by selecting **Reset All**.
2. Select **Suburb menu > Edit column > Add column by fetching URLs**.
3. Type *Geocoding Response* into **New column name**.
4. Type *50* in **Throttle delay**.
5. Type the following exactly into the **Expression** box:

   ```
   "http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address="+escape(value + ", New South Wales", "url")
   ```

6. Click **OK**. Google Refine will query the API for each row in the dataset.
7. Select **Geocoding Response menu > Edit column > Move column to end**.
8. Select **Geocoding Response menu > Edit column > Add column based on this column**.
9. Type *Lat* into **New column name**.
10. Type *parseJson(value).results[0].geometry.location.lat* into **Value**.
11. Click **OK**.
12. Select **Geocoding Response menu > Edit column > Add column** based on this column.
13. Type *Long* into **New column name**.
14. Type *parseJson(value).results[0].geometry.location.lng* into **Value**.
15. Click **OK**.
16. Select **Geocoding Response menu > Edit column > Remove this column**.

Scatterplot Facet

We now have the coordinates of the suburb of every police station in the dataset. We can use the scatterplot facet to hone in on a subset of these:

1. Select **Long menu > Facet > Scatterplot Facet**.
2. Click the highlighted facet area. The scatterplot facet will appear on the left-hand side. Notice it looks uncannily like a map of NSW.
3. Use click-and-drag to select an area of NSW (say the Sydney Basin). Observe the subset of results you get. Note that the **Suburb** and **Address** facets are narrowed down to the matching area.
4. Click **Reset All** to display the entire set of 391 records.

9 Supplementing the data by calling into another API

Searching the State Records NSW Archives

1. Select **Suburb menu > Edit column > Add column by fetching URLs**.
2. Type `Search Response` into **New column name**.
3. Type `200` in **Throttle delay**.
4. Type the following exactly into the **Expression** box:

   ```
   "http://api.records.nsw.gov.au/search.json?entities=Agency&q="+escape(value +" Police Station", "url")
   ```

5. Click **OK**. Google Refine will query the API for each row in the dataset.
6. Select **Search Response menu > Edit column > Move column to end**.
7. Select **Search Response menu > Edit column > Add column based on this column**.
8. Type `State Records Title` into **New column name**.
9. Type `parseJson(value).StateRecordsSearch.Results[0].Result.title` in **Expression**. Click **OK**.
10. Select **Search Response menu > Edit column > Add column based on this column**.
11. Type `Agency URL` into **New column name**.
12. Type `parseJson(value).StateRecordsSearch.Results[0].Result.href` into Expression. Click **OK**.
13. Select **Search Response menu > Edit column > Remove this column**.

Digging Deeper

14. Select **Agency URL menu > Edit column > Add column by fetching URLs**.
15. Type `Agency Response` in **New column name**.
16. Set **Throttle Delay** to `50`.
17. Type `value + ".json"` in **Expression**. Click **OK**.
18. Select **Agency Response menu > Add column based on this column**.
19. Type `History Note` in **New column name**.
20. Type `parseJson(value).agency.Administrative_history_note` in **Expression**. Click **OK**.
21. Select **Agency Response menu > Add column based on this column**.
22. Type `Start Year` in **New column name**.
23. Type `parseJson(value).agency.Start_date.substring(0,4)` in **Expression**. Click **OK**.
24. Select **Agency Response menu > Add column based on this column**.

25. Type *End Year* in **New column name**.
26. Type *parseJson(value).agency.End_date.substring(0,4)* in **Expression**. Click **OK**.
27. Select **Agency Response menu > Edit column > Remove this column**.

10 Exporting the dataset

1. Click **Export** in the top right-hand corner.
2. Select **Comma-separated variable** from the drop-down menu. A CSV dump of your data will be downloaded.