Exercise: Preparing data to be mapped

What we want to do:

- Normalise various date formats and texts in columns (Exercise A)
- Make sure our dataset is complete by locating blank fields and fill them in (Exercise B)
- Add a new column of data (Exercise C)
- Extracting a particular type of data from a longer text string (Exercise D)

1. Create your first Open Refine project

To Run Open Refine:

- Locate and open the “OpenRefine 2.6” folder on the desktop
- Click the “  openrefine.exe” file

This will launch the application in your browser (Note: works best in Google Chrome so if it has opened in IE copy and paste the url http://127.0.0.1:3333/ to Chrome)

There are several options for getting your data set into OpenRefine. You can upload or import files in a variety of formats including:

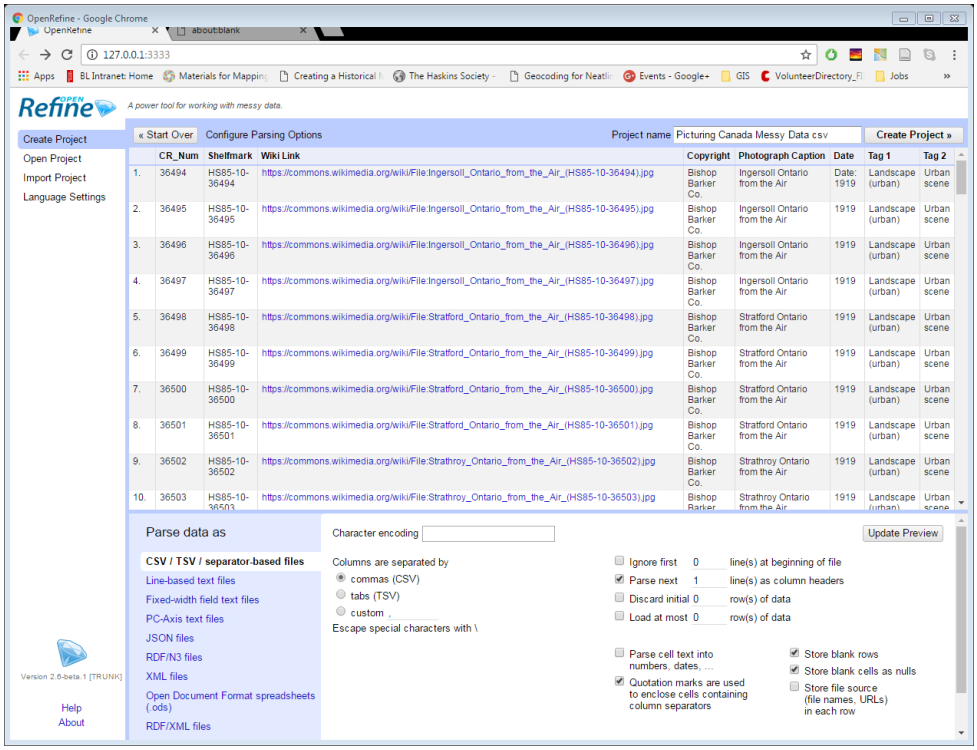
- TSV (tab-separated values)
- CSV (comma-separated values)
- Excel
- JSON (javascript object notation)
- XML
- Google Spreadsheet

Today we will start with a CSV file: “*Picturing Canada Messy Data.csv*”.

CSV is a simple file format used to store tabular data, such as a spreadsheet or database as plain text organised as a series of values (cells) separated by commas (,) in a series of lines (rows). Many applications are capable of reading CSV files, from a simple text editor to Microsoft Excel. XLS files on the other hand contain both content (text and images) and formatting information.

To create a project:

- Click ‘Create Project’
- Choose ‘Get Data from this Computer’
- Click ‘Choose Files’
- Locate the file: “*Picturing Canada Messy Data.csv*”
- Click ‘Next’
- You should see a screen as follows:



This screen gives you some options to ensure that the data gets imported into OpenRefine correctly. The options vary depending on the type of data you are importing. For instance for data with multiple languages you may need to set the Set the ‘Character encoding’ to ‘UTF-8’.

In our particular case you need to:

- Ensure the first row is used to create the column headings
- Make sure OpenRefine doesn’t try to automatically detect numbers and dates

Once you are happy click ‘Create Project >>’

This will create the project and open it for you. Projects are saved as you work on them, there is no need to save copies as you go along.

To open an existing project in OpenRefine you can click ‘Open Project’ from the main OpenRefine screen (in the lefthand menu). When you click this, you will see a list of the existing projects and can click on a project’s name to open it.

OpenRefine displays data in a tabular format. Each row will usually represent a ‘record’ in the data, while each column represents a type of information. This is very similar to how you might view data in a spreadsheet or database.

2. Clean Data Using Facets

Facets “Facets” are one of the most useful features of OpenRefine and can help both get an overview of the data in a project as well as helping you bring more consistency to the data.

A ‘Facet’ groups all the values that appear in a column, and then allows you to filter the data by these values and edit values across many records at the same time.

The simplest type of Facet is called a ‘Text facet’. This groups all the text values in a column and lists each value with the number of records it appears in. The facet information always appears in the left hand panel in the OpenRefine interface.

Exercise A: Normalise various date and text formats in columns using Text Facet

To create a Text Facet for a column, click on the drop down menu at the top of the column and choose **Facet -> Text Facet**. The facet will then appear in the left hand panel.

- Create a text facet for the “Date” column
- Edit the results so that all dates follow either 1919 or 1920
- Create a facet for the “Photograph Caption” column.
- There are two values for “Niagara Falls from the Air”…they look similar but are not….can you see why? Edit these values so they are all the same.

Exercise B: Locate and Fill in blank fields

You can use the Text Facet detailed above to locate blank fields in columns. Another route is through the ‘Facet by blank’ function-useful for quickly checking for missing information in fields of a column.

To try this feature, click on the drop down menu at the top of the column and choose **Facet -> Customized Facets -> Facet by blank**.

- Perform a “Facet by blank” on the column labelled Tag 1.
- Select “true” to view all fields which are blank.
- Edit the values in each field individually.

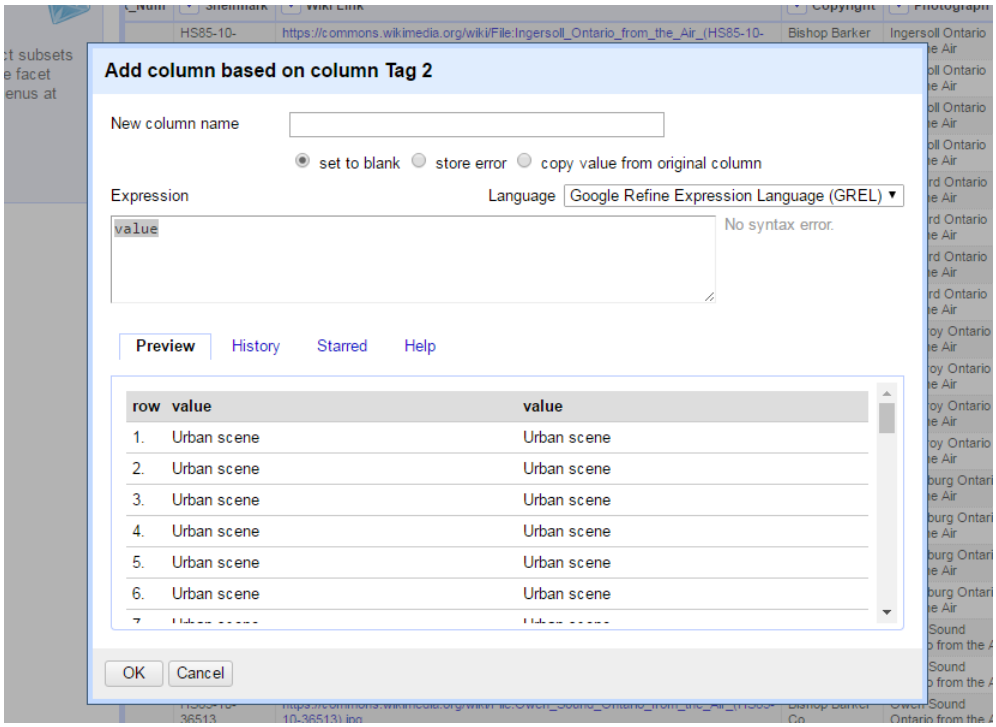
Going further: Do the same with the Tag 2 Column. Compare and contrast with the Text Facet functionality.

Remember you can always Undo/Redo your last steps from the tab on the left-handside!

3. Create a new column

To create new columns in OpenRefine, select any existing column that closely matches, click on the drop down menu at the top of the column and choose **“Edit Column” -> “Add column based on this column…”**

You will to see the following screen:



In this screen you have the ‘Expression’ box which is a place to write what is referred to as a “transformation” on your data, and the ability to Preview the effect the transformation would have on the first rows of your data.

The transformation you type into the ‘Expression’ box has to be a valid GREL expression. The simplest expression is simply the word ‘value’ by itself - which simply means ‘the value that is currently in the column’ - that is, “make no change”.

For example, if I want all the fields in my new column to say “Aerial View” I would replace ‘value’ in the Expression box with **("Aerial View").**

Exercise C: Add a brand new “Aerial Photography” tag

- Create a new column titled “Tag 3”.
- Change the value of all fields in this column to read “Aerial Photography”

4. Extracting a particular type of data from a longer text string (e.g. Location reference in Photo Caption)

The “Photograph Caption” column in our dataset contains references to locations (“Ingersoll Ontario from the Air”) which would be useful in their own column for geocoding. In fact most records in that column follow the same format of location reference followed by the phrase “from the air”.

This means we can use a simple *value.replace* transformation to create a new “Location” column based on the “Photograph Caption” column.

To do this, click on the drop down menu at the top of the “Photograph Caption” column and choose

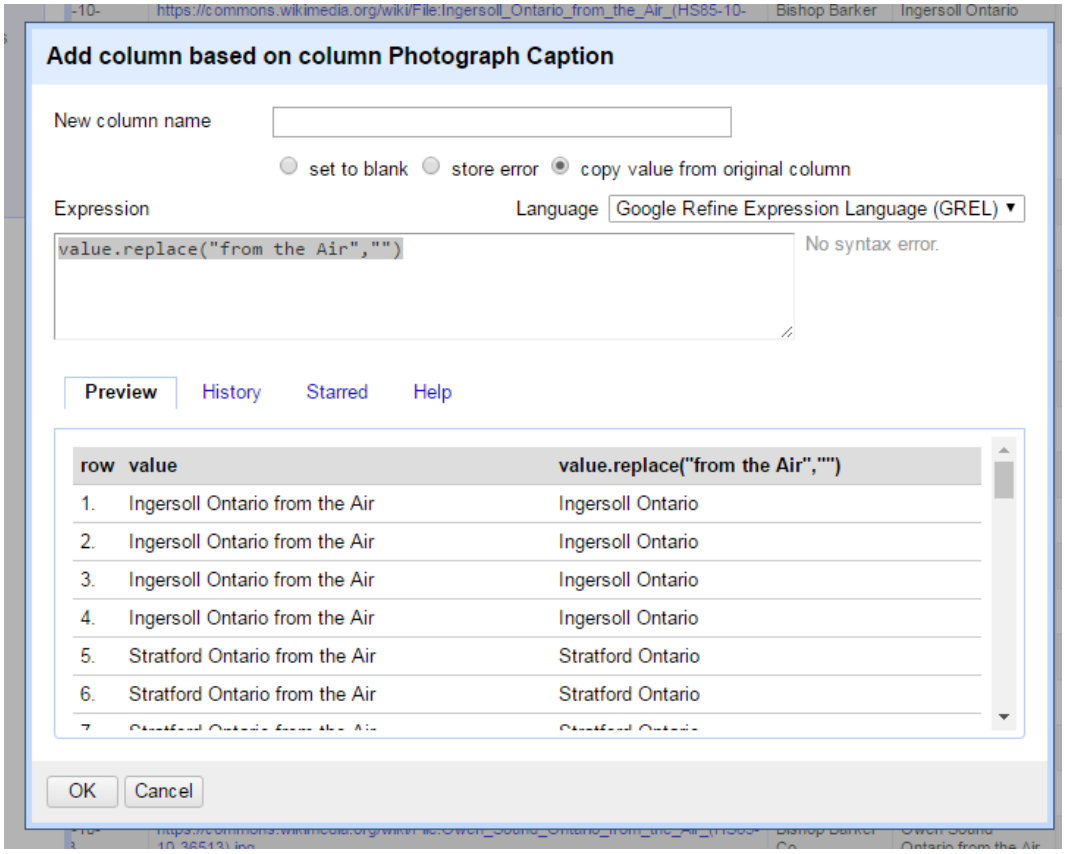
“Edit Column” -> “Add column based on this column…”

Make sure to select “copy value from original column”.

The valid GREL expression syntax for this looks like:

value.replace("existing value","new value")

Note: To denote a blank space simply leave the space in between quotes blank.



Exercise D: Add a new column that contains only the location references within the “Photograph Caption”

- Create a new column titled “Location” based on the existing “Photograph Caption” column.
- Change the value of all fields in this column to include the location data but not ”from the Air”
- Use your Text Facet skills gained earlier to review the contents of this new field and normalise the data to only reflect City and Province (Hint, “Toronto” should be “Toronto Ontario”)

5. Exporting Data

Once you have finished working with a data set in OpenRefine you may wish to export it. The export options are accessed through the ‘Export’ button at the top right of the OpenRefine interface.

Export formats support include HTML, Excel and comma- and tab-separated value (csv and tsv). You can also write a custom export, selecting to export specific fields, adding a header or footer and specifying the exact format.

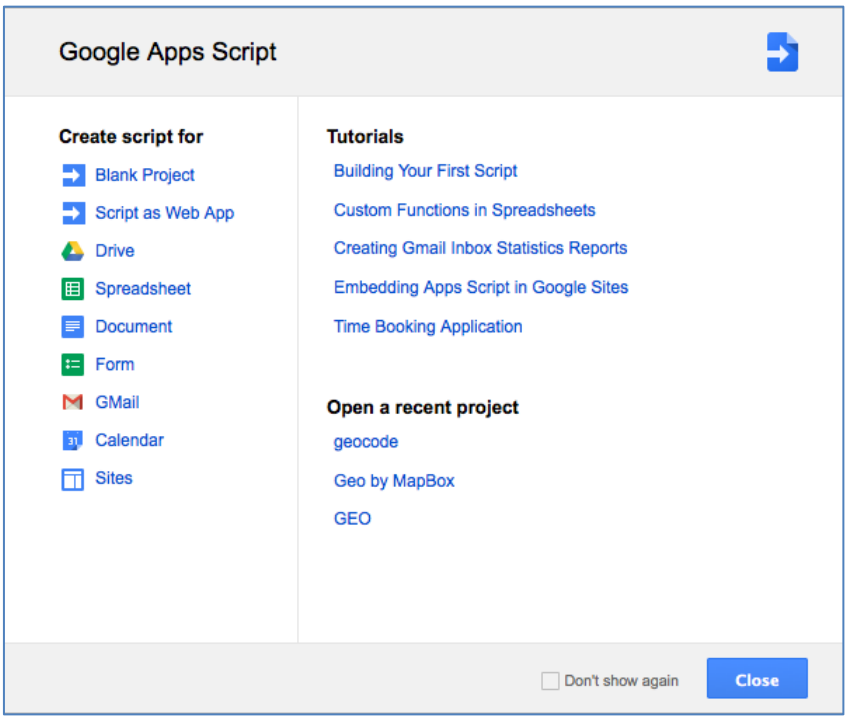
You now have a .csv file with a basic “Location” column that can be used in a variety of mapping programs such as Google Fusion Tables.

6. Using Google Sheets to get Lat/Lon

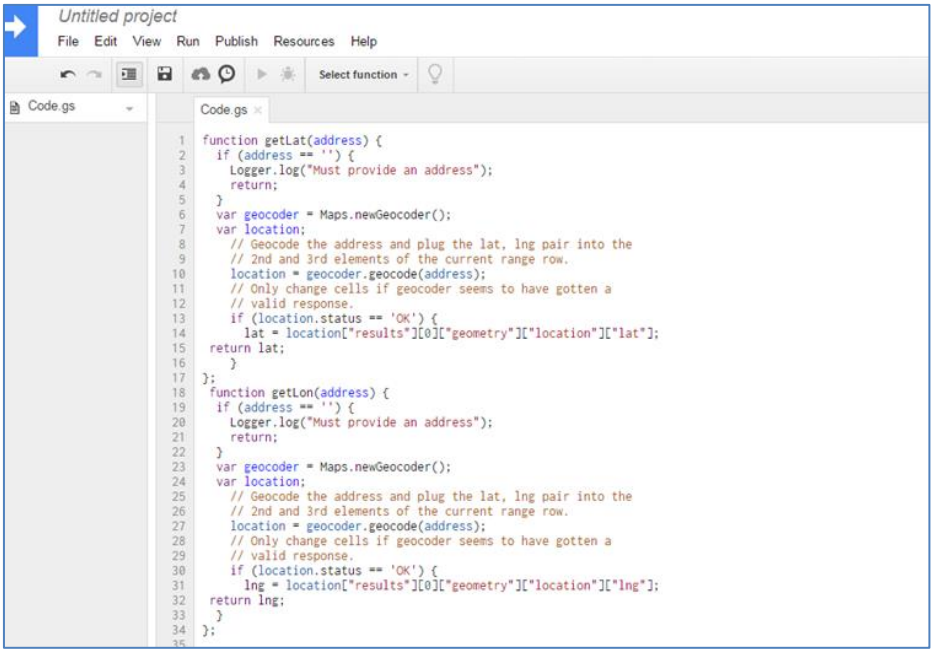
From: http://www.digital-geography.com/geocoding-google-spreadsheets-the-simpler-way/#.WH-litKLTct

Steps:

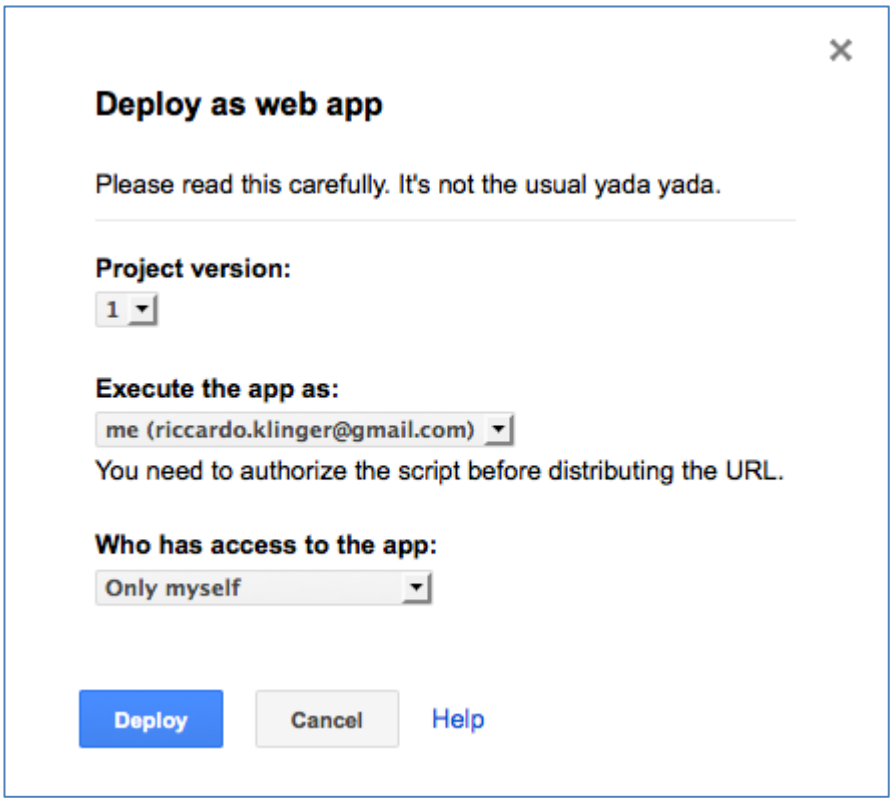
1) Open your .csv file in Google Sheets
2) Add two new columns to your sheet, one for **Lat** and one for **Lon**
3) Go to *Tools->Script Editor* and click on “Create new project” in the new tab of the script editor. In the new dialog choose “Blank Project”:



4) Open **GoogleSheetsGeoCodeScript.txt** in a text editor like NotePad and copy and paste this code into your blank project



5) Save the code and publish it by clicking on *Publish->Deploy as Web App* give it a name and save it as the first version. You should see something similar to this:



6) Press deploy at the end. Press “OK” to close the last dialog you will see which gives you an address for your webapp.
7) Go back to your spreadsheet and refresh it by pressing “F5”
8) In the Lat column add this formula to any cells you’d like to geocode *(Replacing A2 with whatever cell holds the location data in your sheet):*

=getlat(A2)

9) In the Lon column add the formula to any cells you’d like to geocode:

=getlon(A2)

10) You will see a nice and easy set of coordinates which can be used in a webmapping application by publishing your spreadsheet as a csv

WARNING: Google limits the amount of calls you can make to it daily so you may see lots of errors if you try to do many at a time. To get around this, for any successful coordinates gathered remove the formula from those cells so it doesn’t continue to re-process everytime you refresh the sheet.

To remove a formula, right click on the cell or cell, select “Copy” and “Paste Special->Values Only”

7. Going further: Other ways to get Latitude/Longitude

There are a variety of different approaches and applications you can use to further transform text location data into latitude and longitude.

The below are just a few useful online tutorials you might explore to do this:

- GeoCoding in Fusion Tables (Easy, but can't export lat/lon): https://opensas.wordpress.com/2013/06/30/using-openrefine-to-geocode-your-data-using-google-and-openstreetmap-api/
- Geocoding in Open Refine (More difficult): https://opensas.wordpress.com/2013/06/30/using-openrefine-to-geocode-your-data-using-google-and-openstreetmap-api/
- Geocoding in Tableau: https://onlinehelp.tableau.com/current/pro/desktop/en-us/maps_build.html