

Cleaning data with OpenRefine

By MaryJo Webster

Mjwebster71@gmail.com

@MaryJoWebster

Created: March 2015

We're going to clean up the campaign finance data from the Minnesota governor's races we used in an earlier exercise. As we saw in that exercise, there's huge inconsistencies in the donor names, city names and employer names.

We'll need to start by exporting the data from Access (or using the .CSV file provided with this exercise). To export from Access, right-mouse click on the table name and choose "Export" and choose "text file." Follow the steps in the wizard – set it as delimited by a comma, include the field names in the first row and use double quotes as the text qualifier.

You'll need to have Open Refine installed on your computer: <http://openrefine.org/>

Launch Refine. This will open up a new window in your web browser

A little housekeeping, first:

We need to do one housekeeping task because the data we're working with is rather large. Notice in your address bar it should say "127.0.0.1:3333" or something like that. Edit that by adding "/preferences" to the end of the URL.

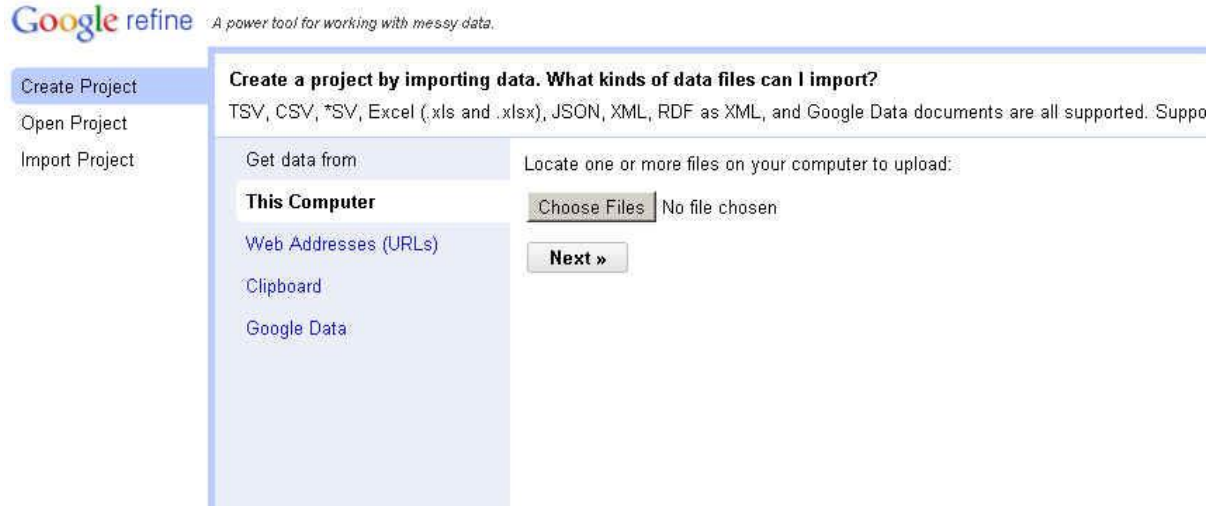
Click the button that says "Add preference."

In the dialog box that comes up type: "ui.browsing.listFacet.limit"

In the next dialog box, it wants a value to set the new facet limit to. I'm going to choose 5,000 for this exercise.

Now go back to the original page by deleting "/preferences" from the URL

You'll see, on the left: Create Project, Open Project, Import Project



Choose "Create Project". Then click "Choose Files" button and navigate to the "governor.txt" data file. Then hit "next"

It will give you a preview of your data and (at the bottom of the page), various options. We're going to leave the defaults.

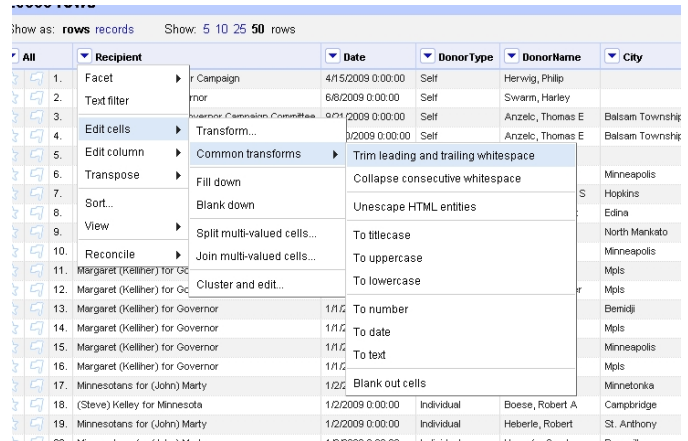
In the upper right corner, you can name your project. Then click "Create Project"

Now you'll see your data – at least the first 10 rows. You can adjust that where it says "Show 5 10 25 50 rows"

BASIC CLEANUP:

Let's start with some basic cleanup. Run the following transformations on each of the text fields (you can skip the date and the amount)

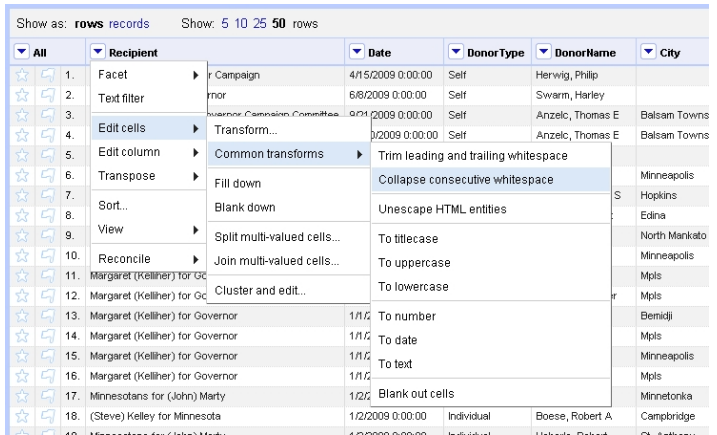
Trim leading and trailing spaces



show as: rows records Show: 5 10 25 50 rows

	All	Recipient	Date	Donor Type	DonorName	City
1.	Facet	Campaign	4/15/2009 0:00:00	Self	Herwig, Philip	
2.	Text filter	mor	6/8/2009 0:00:00	Self	Swarm, Harley	
3.		Donor Campaign Committee	9/21/2009 0:00:00	Self	Anzels, Thomas E	Balsam Townsh
4.	Edit cells	Transform...	0/2009 0:00:00	Self	Anzels, Thomas E	Balsam Townsh
5.	Edit column	Common transforms				
6.	Transpose	Fill down				Minneapolis
7.	Sort...	Blank down				Hopkins
8.	View	Split multi-valued cells...				Edina
9.		Join multi-valued cells...				North Mankato
10.	Reconcile	Cluster and edit...				Minneapolis
11.	Margaret (Kellner) for G					Mpls
12.	Margaret (Kellner) for G					Mpls
13.	Margaret (Kellner) for Governor		1/1/	To number		Bemidji
14.	Margaret (Kellner) for Governor		1/1/	To date		Mpls
15.	Margaret (Kellner) for Governor		1/1/	To text		Minneapolis
16.	Margaret (Kellner) for Governor		1/1/			Mpls
17.	Minnesotans for (John) Marty		1/2/	Blank out cells		Minnetonka
18.	(Steve) Kelley for Minnesota		1/2/2009 0:00:00	Individual	Boese, Robert A	Cambridge
19.	Minnesotans for (John) Marty		1/2/2009 0:00:00	Individual	Heberle, Robert	St. Anthony

Collapse consecutive whitespace:



show as: rows records Show: 5 10 25 50 rows

	All	Recipient	Date	Donor Type	DonorName	City
1.	Facet	Campaign	4/15/2009 0:00:00	Self	Herwig, Philip	
2.	Text filter	mor	6/8/2009 0:00:00	Self	Swarm, Harley	
3.		Donor Campaign Committee	9/21/2009 0:00:00	Self	Anzels, Thomas E	Balsam Towns
4.	Edit cells	Transform...	0/2009 0:00:00	Self	Anzels, Thomas E	Balsam Towns
5.	Edit column	Common transforms				
6.	Transpose	Fill down				Minneapolis
7.	Sort...	Blank down				Hopkins
8.	View	Split multi-valued cells...				Edina
9.		Join multi-valued cells...				North Mankato
10.	Reconcile	Cluster and edit...				Minneapolis
11.	Margaret (Kellner) for G					Mpls
12.	Margaret (Kellner) for G					Mpls
13.	Margaret (Kellner) for Governor		1/1/	To number		Bemidji
14.	Margaret (Kellner) for Governor		1/1/	To date		Mpls
15.	Margaret (Kellner) for Governor		1/1/	To text		Minneapolis
16.	Margaret (Kellner) for Governor		1/1/			Mpls
17.	Minnesotans for (John) Marty		1/2/	Blank out cells		Minnetonka
18.	(Steve) Kelley for Minnesota		1/2/2009 0:00:00	Individual	Boese, Robert A	Cambridge
19.	Minnesotans for (John) Marty		1/2/2009 0:00:00	Individual	Heberle, Robert	St. Anthony

Convert to uppercase

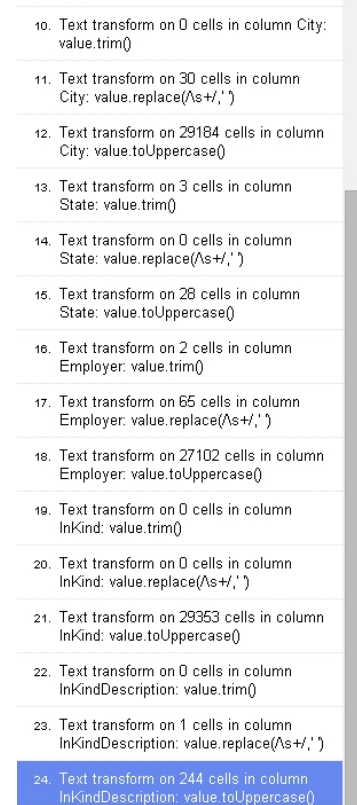
	▼ Recipient	▼ Date	▼ Donor Type	▼ DonorName	▼ City	
1.	Facet	Campaign	4/15/2009 0:00:00	Self	Herwig, Philip	
2.	Text filter	mor	6/8/2009 0:00:00	Self	Swarm, Harley	
3.		Donor Campaign Committee	9/21/2009 0:00:00	Self	Anzels, Thomas E	Balsam Townst
4.	Edit cells	Transform...	0/2009 0:00:00	Self	Anzels, Thomas E	Balsam Townst
5.	Edit column	Common transforms				
6.	Transpose	Fill down				Minneapolis
7.	Sort...	Blank down				Hopkins
8.	View	Split multi-valued cells...				Edina
9.		Join multi-valued cells...				North Mankato
10.	Reconcile	Cluster and edit...				Minneapolis
11.	Margaret (Kellner) for G					Mpls
12.	Margaret (Kellner) for G					Mpls
13.	Margaret (Kellner) for Governor	1/1/	To number			Bemidji
14.	Margaret (Kellner) for Governor	1/1/	To date			Mpls
15.	Margaret (Kellner) for Governor	1/1/	To text			Minneapolis
16.	Margaret (Kellner) for Governor	1/1/				Mpls
17.	Minnesotans for (John) Marty	1/2/	Blank out cells			Minnetonka
18.	(Steve) Kelley for Minnesota	1/2/2009 0:00:00	Individual	Boese, Robert A		Cambridge
19.	Minnesotans for (John) Marty	1/2/2009 0:00:00	Individual	Heberle, Robert		St. Anthony

As you go through, it will tell you how many cells were “fixed” on each one.

Important note: Everything you do in Refine can be undone, very easily. Go to the upper left and look for “Undo/Redo.” Click on that and you’ll get a list like this, showing you everything you’ve done so far.

To undo something, simply click on the item above it.

Let’s say I realize I screwed up on item 17 (in the picture to the right). If I click on #16, it will undo all the items below that.



10.	Text transform on 0 cells in column City: value.trim()
11.	Text transform on 30 cells in column City: value.replace(/s+/, ' ')
12.	Text transform on 29184 cells in column City: value.toUpperCase()
13.	Text transform on 3 cells in column State: value.trim()
14.	Text transform on 0 cells in column State: value.replace(/s+/, ' ')
15.	Text transform on 28 cells in column State: value.toUpperCase()
16.	Text transform on 2 cells in column Employer: value.trim()
17.	Text transform on 65 cells in column Employer: value.replace(/s+/, ' ')
18.	Text transform on 27102 cells in column Employer: value.toUpperCase()
19.	Text transform on 0 cells in column InKind: value.trim()
20.	Text transform on 0 cells in column InKind: value.replace(/s+/, ' ')
21.	Text transform on 29353 cells in column InKind: value.toUpperCase()
22.	Text transform on 0 cells in column InKindDescription: value.trim()
23.	Text transform on 1 cells in column InKindDescription: value.replace(/s+/, ' ')
24.	Text transform on 244 cells in column InKindDescription: value.toUpperCase()

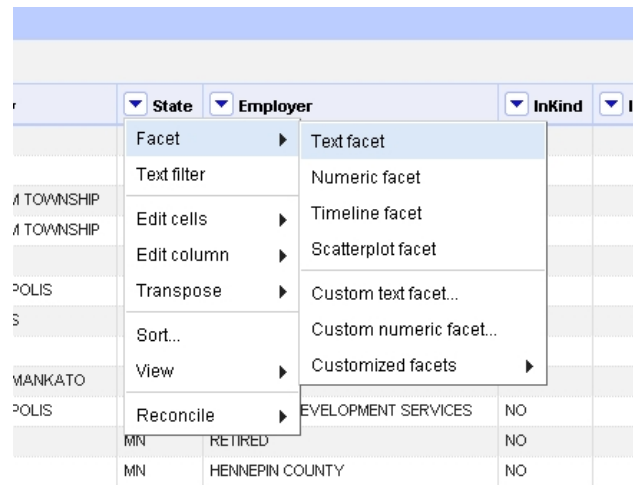
STANDARDIZING VALUES:

The thing that OpenRefine is most known for is standardizing values. So let’s say we have city names spelled various ways (like we do in this dataset), Refine can take an educated “guess” on which ones are the same and it gives you a chance to confirm the results and set the correct name, if necessary. This is much, much faster than any other means of cleaning up data.

The rule of thumb on standardizing data is to only go to this trouble if you need it for your analysis or presenting it online. For this dataset, I want to know who gave the most to the governor campaigns – to do that we need the names standardized. I might also want to put this data online and let people find donors by state or city – that means it might be helpful to have the city names and state abbreviations cleaned up so we can use them for pull-down menus or searches. Then we’ve got the “employer” field. Maybe in this situation, I don’t need that for my analysis so I would leave that one alone.

Let’s start with an easy one – the state field. These are supposed to be postal codes, so how much variation should we expect to find here? Not a lot, usually. But still it’s amazing that you’d find any mistakes here.

Click on the state pulldown and choose Facet >>>
Text Facet



It will bring up a new box on the left side of the screen, listing the variations. You'll see we have "New York," and "MB" and "St Paul" and a few other odd ducks.

Click on "St Paul". It will filter the recordset to just that one record and we can see what's wrong. Clearly here the city landed in the state column. So we can just edit that one by hand. Hover over the city column and you'll see an "EDIT" button appear – push that and type ST PAUL, then push Apply. Repeat that for state field, this time putting MN in as the value.

Do the same for the value in the facet that says "New York."

The push the "Reset All" button above the facet list.

At the top of the Facet box, change the sort to "count" (instead of name)

Scroll to the bottom of the list and you'll see some of the ones where we just have 1 record with that value. For example, "EI" and "SG"

Click on "EI". It looks like this is Hudson – perhaps Hudson, WI?

Click on "MB". This one says Pine Island – perhaps Pine Island, MN?

You would need to do some research on the donors to confirm those are correct before changing them. But usually looking them up in voter registration data or through LexisNexis will square that up for you. For the sake of this lesson, let's leave those values alone.

Click the "x" in the upper left corner of the state facet box – to close the facet box.

Let's repeat that process on the Recipient field. This field SHOULD BE consistent, but let's check. It's hard to tell in the facet box, so let's try clustering.

Push the "Cluster" button at the top of the Facet box.

You'll see that the default clustering method didn't find any potential matches.

You can change the keying function and the method pull-downs to try alternate algorithms.

Cluster & Edit column "Recipient"

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, "york" are very likely to refer to the same concept and just have capitalization differences, and "Gödel" and "Godel" probably refer

Method

Keying Function

No clusters were found with the selected method

Try selecting another method above or changing its parameters

These methods have varying degrees of tightness – in other words, how loose or tight they make their matches. If you leave the method as "key collision" and change keying function to "metaphone3", you'll see that it's trying to match Minnesotans for Matt Entenza and Minnesotans for John Marty --- clearly it's making a very loose match based on both of them starting with "Minnesotans for"

Close this facet.

Let's work on the City field next. Run the facet on City.

Right away at the top we've got a couple things that look like addresses, not cities. Click on each of those and see if you can remedy them.

Try clustering. The default brings up several potential fixes. Clearly one of the problems is that some records have periods and some do not. Let's try fixing those, en masse.

Close the clustering box and go back to the data table.

Go to the city field and choose "Edit Cells" >>> "Transform"

In the box that comes up, we're going to write a little code to tell it to replace any periods with nothing.

Type the expression:

`Value.replace(".", "")`

Then click OK

Custom text transform on column City

Expression Language Google Refine Expression Language (GREL) ▼

`value.replace(".", "")` No syntax error.

Preview History Starred Help

15.	MINNEAPOLIS	MINNEAPOLIS
16.	MPLS	MPLS
17.	MINNETONKA	MINNETONKA
18.	CAMPBRIDGE	CAMPBRIDGE
19.	ST. ANTHONY	ST ANTHONY
20.	ROSEVILLE	ROSEVILLE
21.	MENDOTA HEIGHTS	MENDOTA HEIGHTS
22.	BERKLEY	BERKLEY

On error ☒ keep original ☐ Re-transform up to 10 times until no change
☐ set to blank ☐ store error

OK Cancel

Go back to your facet box and click “Refresh”. Then Cluster it again.

This time it found a few things.

Notice the check box that says “Merge” and the “new Cell value”

Make sure you’re happy with the new cell value and that it’s merging the right things, then click the little “Merge” box next to each one that you want to merge.

Cluster & Edit column “City”

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings “New York” and “new york” are very likely to refer to the same concept and just have capitalization differences, and “Gödel” and “Godel” probably refer to the same person. [Find out more ...](#)

Method key collision ▼ Keying Function fingerprint ▼ 3 clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value	# Rows in Cluster
2	14	<ul style="list-style-type: none">HUGO (13 rows)HUGO* (1 rows)	<input type="checkbox"/>	HUGO	14 — 28
2	28	<ul style="list-style-type: none">BEVERLY HILLS (27 rows)BEVERLY HILLS* (1 rows)	<input type="checkbox"/>	BEVERLY HILLS	Average Length of Choices 4.5 — 13.5
2	21	<ul style="list-style-type: none">LAUDERDALE (19 rows)LAUDERDALE* (2 rows)	<input type="checkbox"/>	LAUDERDALE	

Select All Deselect All Merge Selected & Re-Cluster Merge Selected & Close Close

Then push “merge selected & re-cluster” at the bottom.

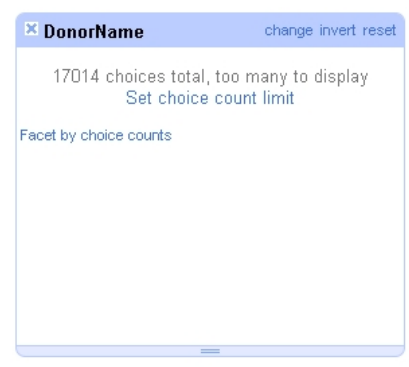
Change the keying function to “ngram-fingerprint” to get a looser match. It looks like there are a bunch more here to cluster. Fix any here, then merge and re-cluster – this time switching to “metaphone3”. Just keep repeating this process until you’ve exhausted all your options. Then you’ll have to look through the facet window for odd balls that the algorithms didn’t catch. For example, go look for “Mpls”

Hover over the right side of the facet window and you'll get an "Edit" link.



Click "Edit" and you can type in the correct spelling of MINNEAPOLIS and hit Apply to fix all 1,284 of them.

Now, it's try to facet the Donor Name field. You'll get an error message like this:



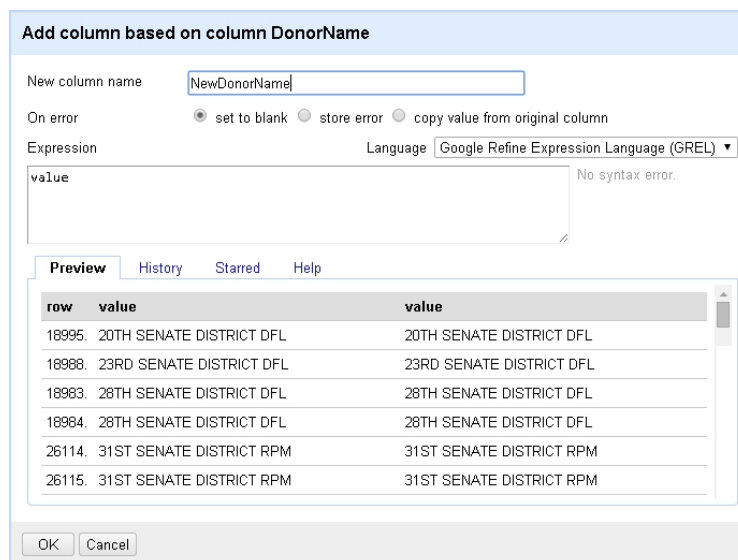
Since we know we need to cluster this data anyway, we can skip the faceting process – for now -- and go straight to clustering.

But first, this time let's put our results into a new column, so that we can preserve the original donorname column – just in case.

Go to Donorname and choose "Edit Column" and "Add column based on this column"

In the dialog box that comes up, the only thing you need to do is name your new column. I named mine "NewDonorName"

(when you get to know Refine better, you can use do all kinds of fancy stuff in the Expression box to put something other than the original value into the new field)



Let's also repeat that process of getting rid of periods (like we did for the city field)

Go to the new donor name field and choose “Edit Cells” >>> “Transform”

In the box that comes up, we’re going to write a little code to tell it to replace any periods with nothing.

Type the expression:

```
Value.replace(".", "")
```

Then click OK

Now go to your new field again and choose “Edit Cells” and “Cluster and Edit”.....and start editing!

Word of wisdom: Cleaning up names of people is a whole order more difficult than cleaning up city names. With people, there could very easily be two people called “Michael Goldner” and if you’ve got one listed as “Michael Goldner” and the other as “Michael D Goldner” – are they the same?

Here’s where you might have to look at the individual records and see if the employer, city and state fields match up.

When you hover over a name in the clustering window it will give you the option to “Browse this cluster.” It will open those records up in a new window, so you can see if they are the same person.

		• OLSON, NEWMAN E (1 rows)		
2	4	• GOLDNER, MICHAEL (2 rows) • GOLDNER, MICHAEL D (2 rows)	<input checked="" type="checkbox"/>	GOLDNER, MICHAEL D
		Browse this cluster		
2	3	• ERICKSON, RONDIC (2 rows) • ERICKSON, RONDIC (1 rows)	<input checked="" type="checkbox"/>	ERICKSON, RONDIC

Once you’re happy with your data cleanup, you can export the data by going to the “Export” button in the upper right corner of Refine, choose the file type you want (I’m going to choose “comma-separated values”)

You can also “export project” to save all your work (and can re-import it later). However, as long as you’re on the same computer, you will find your project the next time you open Refine.

More info on OpenRefine:

<https://github.com/OpenRefine/OpenRefine>

FAQ: <https://github.com/OpenRefine/OpenRefine/wiki/FAQ>

[http://datadrivenjournalism.net/resources/Getting to Know your Dataset with the OpenRefine Facets](http://datadrivenjournalism.net/resources/Getting_to_Know_your_Dataset_with_the_OpenRefine_Facets)

<http://googlerefine.blogspot.com/>

More training materials by MaryJo Webster

[Mjwebster.github.io/DataJ](http://mjwebster.github.io/DataJ)