**Cleaning Data with Refine (v3.0 update)**

**What you’ll need:**

1. Refine – Download it from [openrefine.org](http://openrefine.org)
2. The sample Dataset – Download it from <https://goo.gl/EJxa20> (or, use this URL and the ‘web address’ option to import directly into Refine)

**Step 1: Creating a new Project**

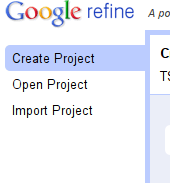
Open Refine (previously Google Refine) is a data cleaning software that uses your web browser as an interface. This means it will look like it runs on the internet but all your data remains on your machine and you do not need internet connection to work with it.

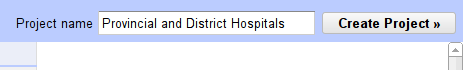
The main aim of Refine is to help you exploring and cleaning your data before you use it further. It is built for large datasets – so don’t worry as long as your spreadsheets can keep the information: Refine can as well.

To work with your data in Refine you need to start a new project:

**Walkthrough:** Creating a Refine project

1. Start Refine – this will open a browser window pointing to <http://127.0.0.1:3333> if this doesn’t happen open the link with your browser directly
2. Create a new project: On the left tab select the “Create Project” tab:



1. Click on “Choose Files” to choose your downloaded file and click on “next” – you can also use the URL to the CSV directly if your data is hosted on the web (e.g. <https://goo.gl/EJxa20> )
2. You will get a preview on how refine will interpret your data – if you have selected a well formatted CSV or other file: this should be pretty automatic.
3. Review the preview carefully to make sure the data looks right. Double check character encoding. Much, but not all data uses UTF-8 these days, but make sure you don’t see any funny characters in preview.
4. Name your project in the box on the top right side and click on “Create Project”
5. The project will open in the project view, this is the basic interface you are going to work with: by default refine shows only 10 rows of data, you can change this on the bar above the data rows. Also you can use the navigation on the right to see the next or previous rows.

You now have successfully created your first Refine project. **Remember:** although it runs in a web-browser, the Refine server is still on your machine – all the data is there (so no worries if you handle sensitive information)

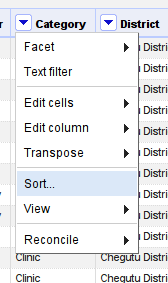
**Step 2: Sorting and Facetting**

Once we created our project, let’s go and explore the data and the Refine interface a bit. Using Refine might be intimidating at first, since it seems so different from spreadsheets, once you get used to it you will notice how easily you can do things with it.

One of the commonly used functions in spreadsheets is sorting and filtering data – to figure out minima, maxima or things about certain categories. Refine can do the same thing.

**Walkthrough:** Sorting rows

1. Refine handles data similar to a spreadsheet: you have rows, columns and cells – a cell is a field defined by a row and a column.
2. To sort your rows based on a specific column click on the small downward triangle next to the column

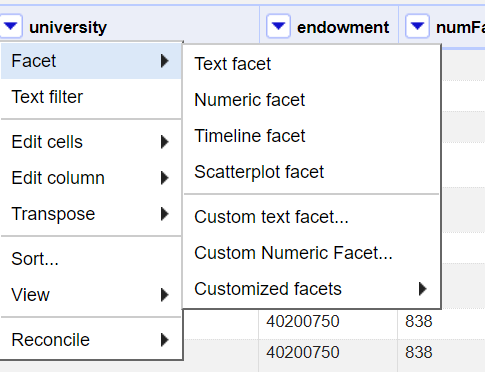


1. Select “Sort…” to open the sorting dialog
2. You can select what to sort the values as and then what order to sort in. (We’ll sort in text, since for now we only have text columns)
3. Click “OK” and your rows will be sorted based on the column
4. To undo the sort, click on the column options again, select “sort” then “remove sort”

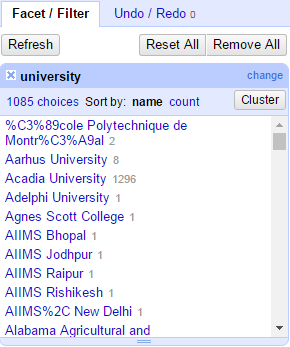
The other frequently used function in Spreadsheets is filtering – in Refine this is called facetting. Facetting in Refine is really powerful – you will see in most of the rest of the Recipe we’ll use facets.

### Walkthrough: Facetting rows based on a column

1. Select the column options for the column you want to facet with
2. Select “Facet” for “university” to start with



1. You can facet differently for text, numbers or dates – let’s facet as text – click on “Text facet”
2. This will open a facet in the left bar

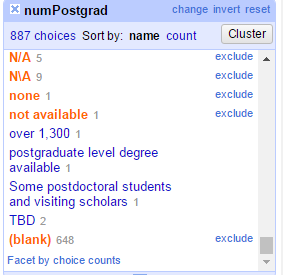


1. Now select one or more of the choices and you’ll see how your data rows are limited to just those selected.
2. Of course you can add more than one facet and thus filter more than once.

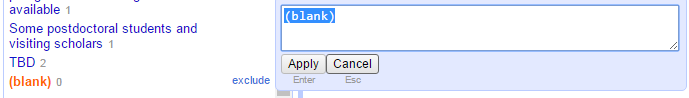
**Step 3 Dealing with Blank Cells and merging**

**Challenge: How many rows should the North American country that isn’t Canada have?**

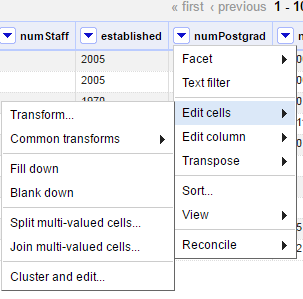
If you look closely at some facets, you’ll notice that on the bottom you have a selector saying “(blank)”. You’ll also see some duplicate options (e.g. N/A, N\A, none, etc.)

**Walkthrough:** Filling in the (blank)s

1. Choose the “(blank)” facet in your “numPostgrad” column (it’s right at the bottom)
2. Hover over ‘edit’ and replace (blank) with NA (or whatever other value), and hit ‘apply’



1. You can also edit individual values within the table on the right http://i0.wp.com/farm9.staticflickr.com/8506/8570737157_b168621858_d.jpg?w=591
2. At the top right of the ‘facet’ option hit ‘cluster’
3. This allows you to merge multiple the multiple NA values using string matching – note that ‘none’ (i.e., 0) is not the same as NA (not available)
4. Play with the different options on the ‘country’ column and merge the values



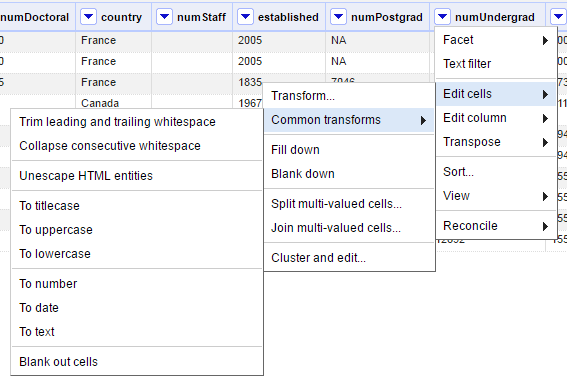
1. You can also access this option through the ‘edit cells’ menu, and navigating to ‘cluster and edit’ (note the other options here are also very useful)

**Step 4: Fighting the Invisible Man**

As illustrated in [The Invisible Man is in your Spreadsheets](http://schoolofdata.org/handbook/courses/data-cleaning-invisible-man-in-spreadsheets/) having spaces or newlines in your datafields is a problem. Since this is a very common problem, Refine has specific functions to remove whitespaces that shouldn’t be there.

**Walkthrough:** Removing hidden whitespaces

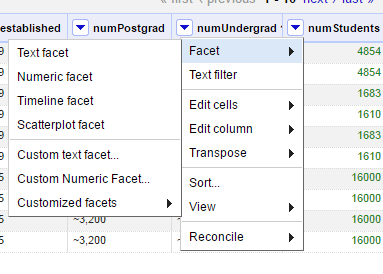
1. An issue you may encounter in your data is that values *look* identical to you, while looking different to the computer
2. A common example of this is in the presence of ‘white space’: in a leading or trailing position (i.e., before or after another string); or with variable space lengths (e.g., a double space “ “ versus a single “ “)
3. Refine has a way to remove these
4. Refine can help you clean this up in an instant – open the column options for the “numUndergrad” column
5. Select “Edit Cells” – “Common Transforms” – “Collapse consecutive whitespace”



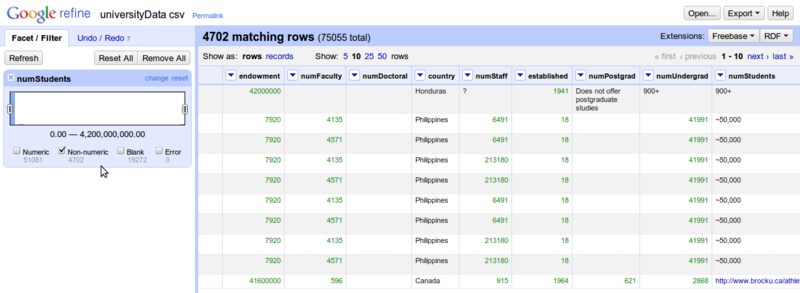
**Step 5: Advanced Clean up values for the number of students and removing bad data**

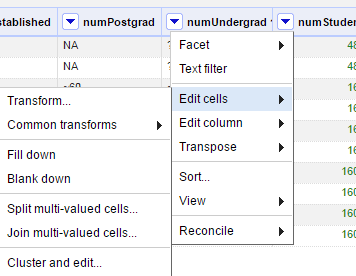
**Challenge: Where has the most students? (Are the first answers you find plausible?)**

We need to clean the data for the number of students. Not all of the values are numeric, and many of them contain bits of text in addition to the actual number of the students.

To figure out which entries need to be fixed, we need to use a Numeric facet (you might need to do “Edit Cells -> Common Transformations -> To number” first):

This shows us a histogram of the values, and also lists the number of entries per type (numeric, non-numeric, blank, error, etc). Make sure that only the non-numeric rows are selected (see below)

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_SelectNonNumericValuesFromNumericFacet.png)



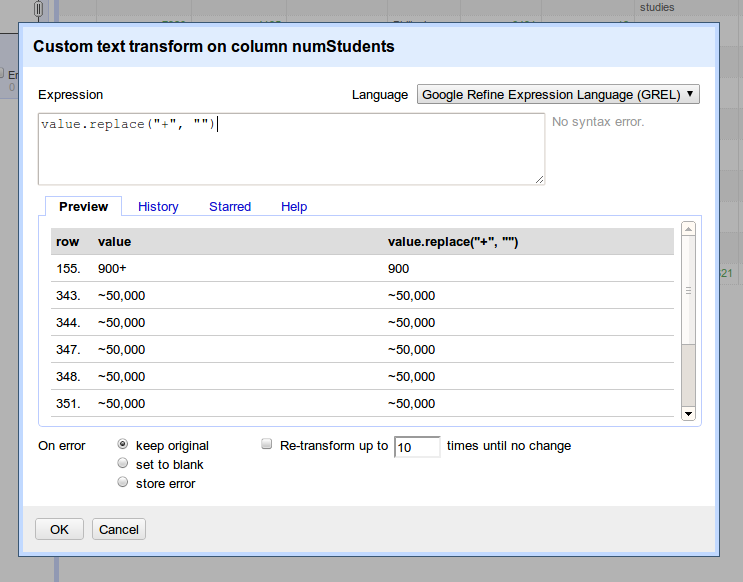
We can see some problems already, as some cells have "+" and "~" in them. To fix this, we need to do **Edit cells -> Transform**

This allows us to now type in commands that can replace sequences of characters:

value.replace("+", "")

Also, if you see entries with strange symbols like "Lumi%C3%A8re University Lyon 2" in the "x" column (should be "Lumière University Lyon 2"), you can fix this via **Edit cells -> Transform** with this command:

value.unescape('url')

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_TextTransform.png)

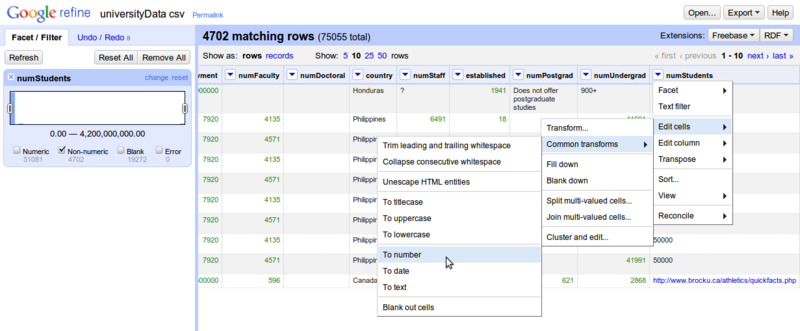
In doing this, you're actually using bits of a programming language. A lot of advanced features are available (not covered in this tutorial), and if you want to understand this further, you can refer to the [Google Refine Expression Language (GREL) reference](http://code.google.com/p/google-refine/wiki/DocumentationForUsers#Reference)

If you find multiple things that need to be replaced, you don't have to keep clicking **Edit cells -> Transform** for every single issue. You can chain these commands together to fix several issues at once: **On numStudents** use **Edit cells -> Transform** to do the following:

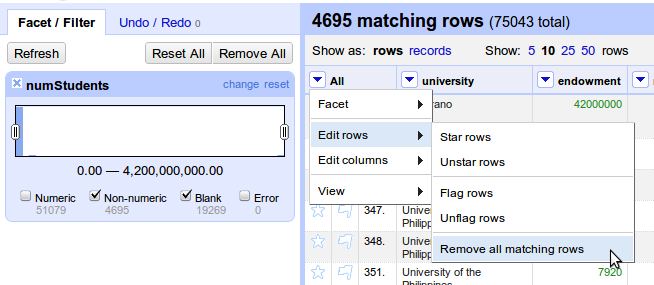
value.replace("~", "").replace(",","")

More issues can be cleaned up via:

value.replace(" total", "").replace(" -", "")

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_ConvertToNumber.png)In order to update the selection of non-numeric values, it's sometimes necessary to convert the values of the columns to numbers - **Edit cells -> Common transforms -> To number**. Once you do this, you should see that there are fewer non-numeric values.

You can continue cleaning up the data, but for this exercise we will move on and remove all the rows that do not have numeric values for the number of students. To do this, use a numeric facet again on numStudents to select only the non-numeric and blank values. Then do **All -> Edit rows -> Remove all matching rows**

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_RemoveAllMatchingRows.png)

**Step 6: Clean up values for the endowment**

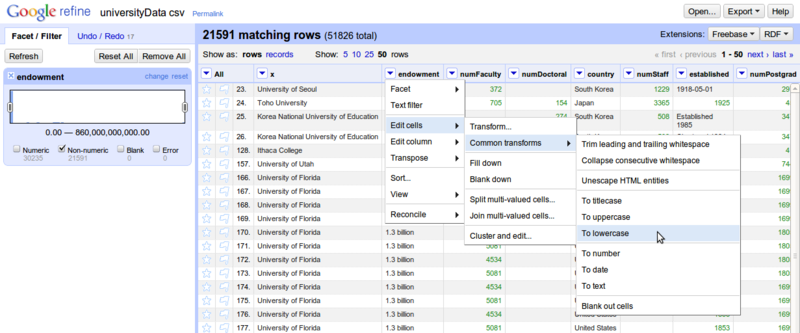
It's possible to have multiple facets in use at once. When you do this, each additional facet makes a sub-selection of the data selected by the previous facet. If you find that the number of rows you have selected and are working with is smaller than expected, then check to see if you still have facets in use which are not needed any more.

First remove the numeric facet for numStudents and create a new numeric facet for endowment (remember you might have to transform the cells to numeric first). Select only the non-numeric values, as was done for the number of students, and add a text facet alongside.

Already we see issues like "US$1.3 billion" and "US $186 million"

Assuming that everything is in $ (a somewhat bogus assumption), we can clean up the data similarly to how we did it before. Click on the endowment column -> **Edit cells -> Transform**

value.replace("US $","").replace("US$", "")

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_ConvertToLowerCase.png)Both "million" and "Million" are in the values, so it's useful to convert all the values to lowercase instead of cleaning this up twice.

Click on the endowment column again, and create a custom text facet to locate all the rows with the word "million" in them: **Facet -> Custom text facet**

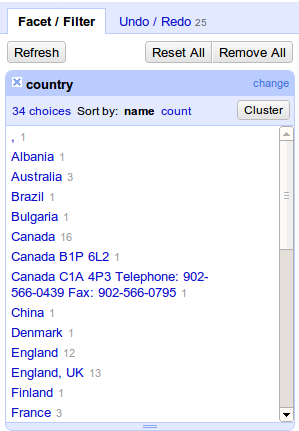
value.contains("million")

Then **Edit cells -> Transform**. It's not advisable to just replace "million" by "000000" since you have some values like "$13.8 million", which would be converted to "$13.8000000". It's better to first remove "million" from the text, convert the remaining text to a number, and then multiply this by 1000000:

toNumber(value.replace(" million", "").replace("A$", "").replace("$", ""))\*100000

The term "billion" is in the values as well, so remove previous facet for endowment, and create a new one for billion, and repeat process described above.

After most of this has been cleaned up, select the non-numeric values, and delete them, just as was done for the numStudents.

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_TextFacet.png)**Step 7: Looking for *more* issues in other columns**

OpenRefine has plenty of features that can help clean up the other columns as well. For example, if you do a text facet on the column with country names, you will find issues such as entries for both "England" and "England, UK", along with entries for Canada that contains parts of the university address.

How can you clean these using the methods above?

**Step 8: Cleaning up dates**

The dates are a mess as well, but there are a few techniques that can be used to help clean them up.

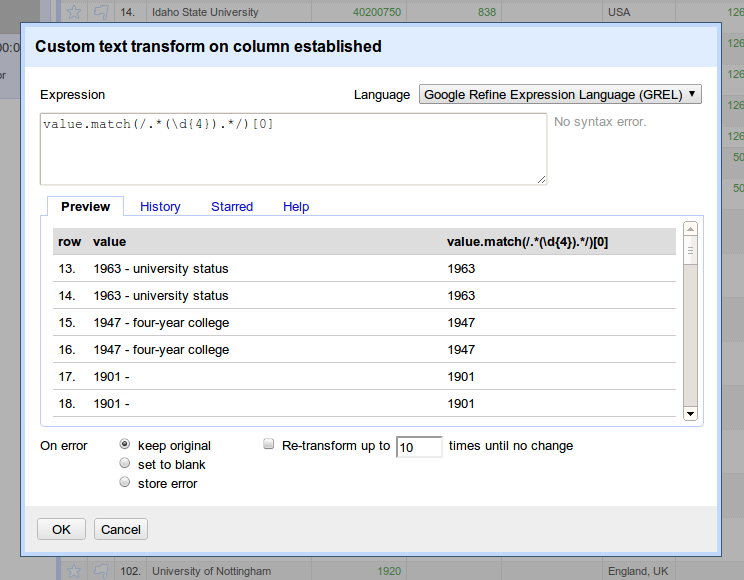
First we want to convert everything to text - **Edit cells -> Common transformations -> To text**, and then you need to **Edit cells -> Common transformations -> To date**. If you did not convert all the values to text first, then you may find that some of the years are represented as numbers, and have not been converted.

If only a year is listed, then the date created will use January 1st as the month and day. We will clean this up later to use only the year.

To further clean up the dates, we need to use **Facet -> Timeline facet** and select only "Non-Time" values. This shows us that we have a bit of a tricky situation as years are mixed in with text such as "Established 1985". We need some way to recognize a sequence of four numbers in a section of text and extract only the numbers. To do this, we need to use [regular expressions](http://code.google.com/p/google-refine/wiki/UnderstandingRegularExpressions). This is a very powerful technique that allows you to specify very complex patterns that you wish to match. For this tutorial, you don't need to know how to write regular expressions, but you should at least know that they exist, and that they can be used to help you with seemingly impossible tasks.

We now want to do **Edit cells -> Transform**, and use the code below. The ".\*" means a sequence of zero or more characters (letters, numbers, symbols, etc). The "\d" indicates that we're looking for a digit. The "{4}" shows that we want to match exactly 4 digits. The value.match function returns an array of results, so we use "[0]" to retrieve only the first match.

value.match(/.\*(\d{4}).\*/)[0]

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_CleanUpDates.png)

We can now convert these extracted values to dates - **Edit cells -> Common transformations -> To date**. At this point, we've done almost everything we can to track down usable dates, and we now want to just extract the years. To do this, we want to **Edit cells -> Transform** with the code below:

value.toString('yyyy')

What's happening here is that we're using a string ('yyyy' in this case) to specify what parts of the date we want, and how it should be displayed. The [documentation here](http://docs.oracle.com/javase/1.4.2/docs/api/java/text/SimpleDateFormat.html) describes this in much more detail. As illustrated in the table below, you can experiment with different commands to get different formats of dates.

|  |  |
| --- | --- |
| **Command** | **Result** |
| value.toString('M') | 1 |
| value.toString('MM') | 01 |
| value.toString('MMM') | Jan |
| value.toString('MMMM') | January |

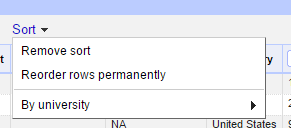
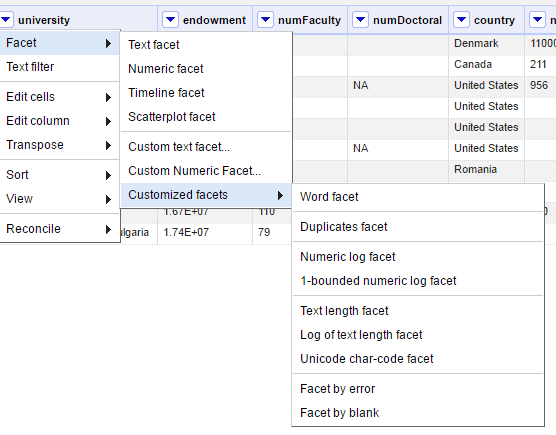
As described [here](http://code.google.com/p/google-refine/wiki/GRELDateFunctions), you can use code such as that below to reformat multiple date formats into a single format.

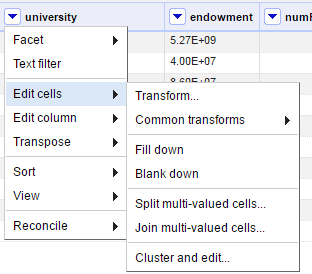
value.toDate('MM/yy','MMM-yy').toString('yyyy-MM')

**Challenge: Which college is the most recently (uhum) established one?**

**Step 8: Deduplicate entries ([near] duplicate rows)**

There are a lot of (nearly) duplicate rows in the data, this can happen for various reasons including human error and because multiple historic values are stored. We want to keep just one copy.

To do this (based on documentation [here](http://googlerefine.blogspot.nl/2011/08/remove-duplicate.html)), click on the column with the university names, and then click on "Sort". Once you do this, you will notice that there is a new "Sort" menu at the top. Click on this and select "Reorder rows permanently". This may take a while as it renumbers the rows in which the entries appear.

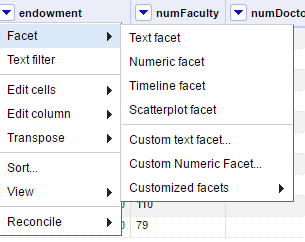


Then on the column with university names, **Edit cells -> Blank down**

Then on the same column, **Facet -> Customized facets -> Facet by blank**

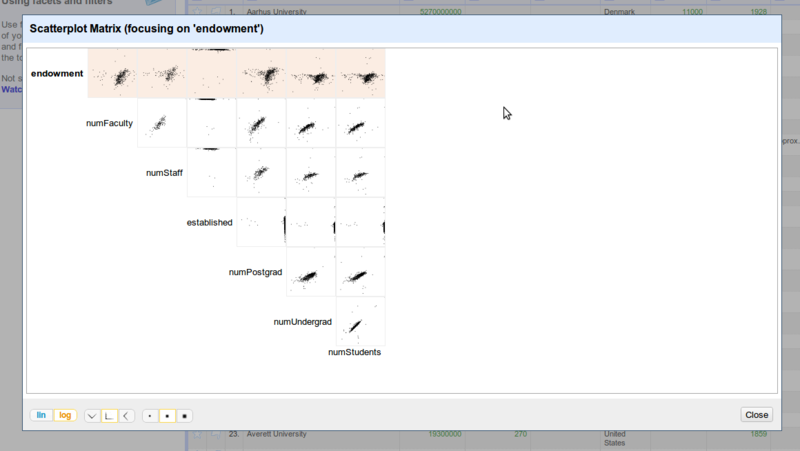
Now we want to remove all the blank rows, so select true, then on the "All" column on the left, Edit rows -> Remove all matching rows, like you have done when working with the numStudents and endowment columns.

Once you remove all the facets, and you now have a (mostly) cleaned data set.

**Step 9: Exploring the data with scatter plots**

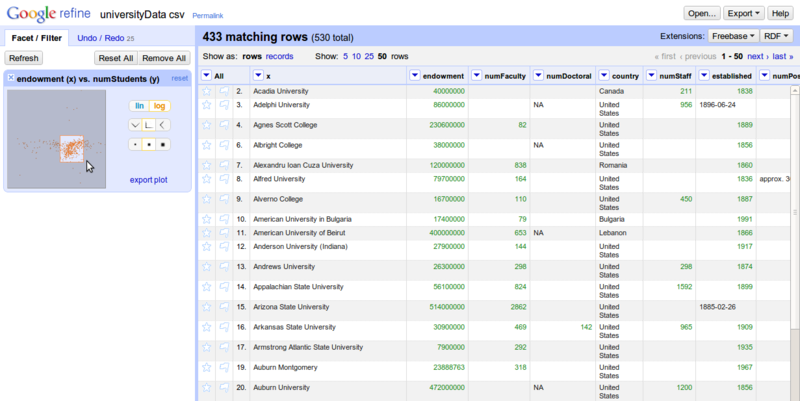
Click on the "endowment" column, **Facet -> Scatterplot facet**.

This shows the relationships between all of the numeric values in each of the columns. Click on "log" to get a better view.

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_ScatterplotFacet.png)

Click on the plot for endowment vs. numStudents. You can now drag select a portion of the plot, and then see the rows corresponding to that selection.

**Challenge: Where has a very high endowment with a very low number of faculty?**

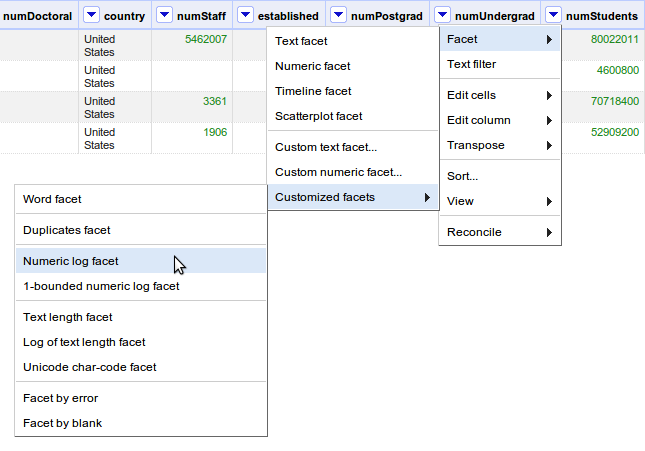
[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_SelectFromScatterplotFacet.png)

**Step 10: Geocoding names and addresses**

This next part shows (based on documentation [here](http://code.google.com/p/google-refine/source/browse/wiki/Geocoding.wiki?r=1342)) how to go from a description of a place (i.e. the name of a university) to values for its (likely) geographic coordinates. Behind the scenes, this uses Google Maps to figure out what is the most likely location you are asking for.

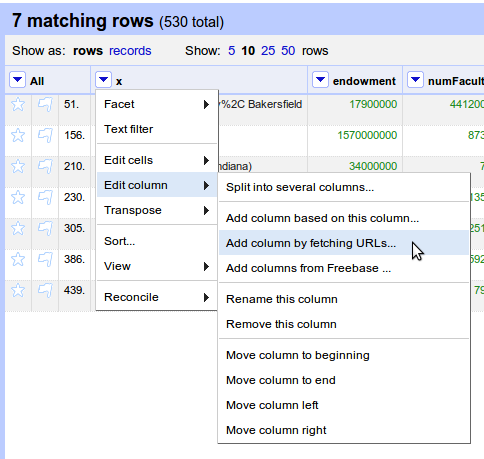
To learn how to do this, you don't need to do process the whole data set. This can take a while, and Google limits you to 2000 requests per day. It's better to just select around 10 rows and verify that it works.

An easy way to get a limited set of rows is by using a numeric log facet of the number of students, so use **Facet -> Customized facets -> Numeric log facet**

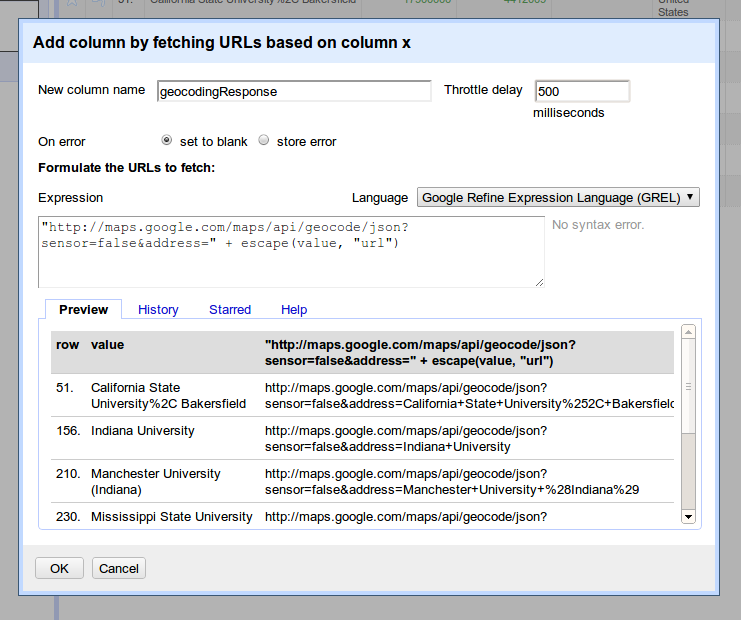
[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_NumericLogFacet.png)

Use this facet to make a selection of around ten rows, and then check the **matching rows** number to verify that you have a reasonable selection size:

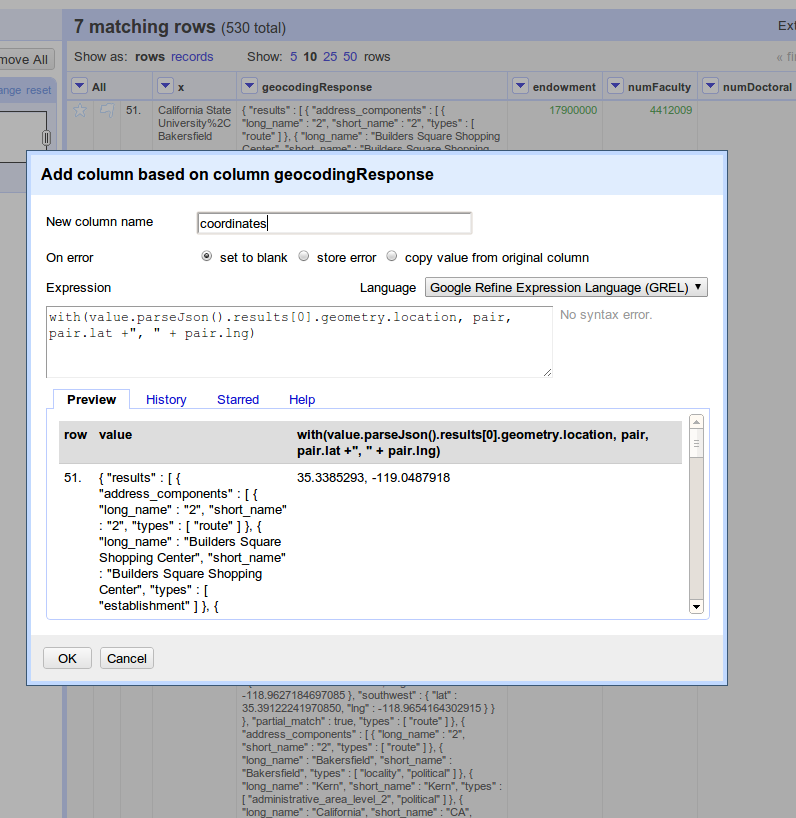
[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_NumberOfMatchingRows.png)

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_AddColumnByFetchingURLs.png)

Now the fun begins and we want to do **Edit column -> Add column by fetching URLs**. In other words, the values of the cells in the new column are based on data that is retrieved from the Internet.

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_FetchURLsExpression.png)Enter in the expression below, and you should see a list of URLs with the names of the universities at the end of the URLs. Specify a new column name such as "geocodingResponse", and set the throttle delay to around 500 milliseconds.

"http://maps.google.com/maps/api/geocode/json?sensor=false&address=" + escape(value, "url")

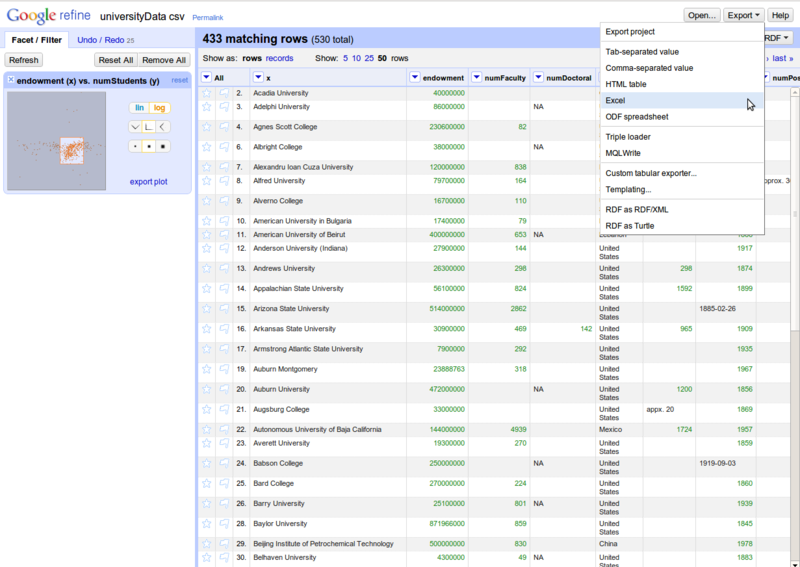
[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_CleanUpGeocodingResponse.png)You should get a bunch of data back. To convert this into a more readable format, you need to click on the geocodingResponse column, and then on **Edit column -> Add column based on this column**. Enter in the expression below

with(value.parseJson().results[0].geometry.location, pair, pair.lat +", " + pair.lng)

Now you have a single column with coordinates. You can split this into columns for latitude and longitude by selecting **Edit Column -> Split into several columns** and specifying a separator of ",". These columns can then be renamed using **Edit Column -> Rename this column**.

**Step 11: Export Data**

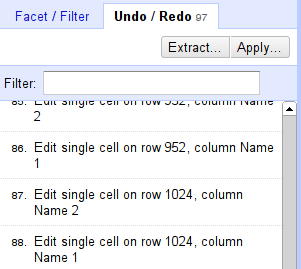
The data can be exported to formats such as Excel. If you read this into tools such as SPSS and notice that the last column is missing, then open the file up in Excel, re-save it, and try to open it up again in SPSS.

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_ExportToExcel.png)

## Conclusions

**Congratulations – You successfully cleaned up a dataset using Refine!**

However there is even more you can do with Refine: For example did you notice how there is always a number next to the categories in the facet – telling you how many rows are in that category? By combining two facets, can you find out how many Universities with under 1000 students there are in the USA?

Once you made your transforms you might wonder: What if I made a mistake? Also if you work with data you generally want to keep track of what you did to the data. Since Refine was build with data processing in mind it keeps track of what you’re doing with your data and allows you to go back and forth in time. To see your history of changes click on the “Undo/Redo” tab on the left.

You see all the changes you made – by simply clicking on one of the steps you’ll be undoing all the changes after the step (don’t worry you can redo pretty much the same way). Play with this system until you are comfortable.

## Lessons learned

1. Creating projects in OpenRefine, sorting and faceting to explore data
2. Cleaning up inconsistent spelling of terms (i.e. "USA", "U.S.A", "U.S.", etc).
3. Converting values that are text descriptions of numeric values (i.e. $123 million) to actual numeric values (i.e. 123000000) which are usable for analysis.
4. Identifying which rows of a specific column contain a search term
5. Extracting and cleaning values for dates
6. Removing duplicate rows
7. Using a scatterplot to visualize relationships between values in different columns
8. Finding geographic coordinates for a list of place names (i.e. the names of universities, etc.)
9. Exporting cleaned data to Excel

## Acknowledgement

This resource is adapted from:

1. [**http://enipedia.tudelft.nl/wiki/OpenRefine\_Tutorial#Clean\_up\_country\_names**](http://enipedia.tudelft.nl/wiki/OpenRefine_Tutorial#Clean_up_country_names) **(under a CC-By-SA 3.0 license)** C.B.Davis, A. Chmieliauskas, G.P.J. Dijkema, I. Nikolic (2015), Enipedia, <http://enipedia.tudelft.nl>, Energy & Industry group, Faculty of Technology, Policy and Management, TU Delft, Delft, The Netherlands.
2. And <http://schoolofdata.org/handbook/recipes/cleaning-data-with-refine/>

Both under a CC-By-SA license. To cite, Knight, S., (2016). Cleaning Data with Refine,

**Cleaning Data with Refine (v2.6)**

**What you’ll need:**

1. Refine – Download it from [openrefine.org](http://openrefine.org)
2. The sample Dataset – Download it from <https://goo.gl/EJxa20> (or, use this URL and the ‘web address’ option to import directly into Refine)

**Step 1: Creating a new Project**

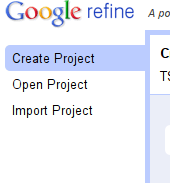
Open Refine (previously Google Refine) is a data cleaning software that uses your web browser as an interface. This means it will look like it runs on the internet but all your data remains on your machine and you do not need internet connection to work with it.

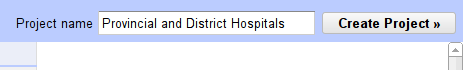
The main aim of Refine is to help you exploring and cleaning your data before you use it further. It is built for large datasets – so don’t worry as long as your spreadsheets can keep the information: Refine can as well.

To work with your data in Refine you need to start a new project:

**Walkthrough:** Creating a Refine project

1. Start Refine – this will open a browser window pointing to <http://127.0.0.1:3333> if this doesn’t happen open the link with your browser directly
2. Create a new project: On the left tab select the “Create Project” tab:



1. Click on “Choose Files” to choose your downloaded file and click on “next” – you can also use the URL to the CSV directly if your data is hosted on the web.
2. You will get a preview on how refine will interpret your data – if you have selected a well formatted CSV or other file: this should be pretty automatic.
3. Review the preview carefully to make sure the data looks right. Double check character encoding. Much, but not all data uses UTF-8 these days, but make sure you don’t see any funny characters in preview.
4. **You may want to turn off “guess data types”, particularly if you have data that contains leading zeros in numbers or identifiers which are significant.**
5. Name your project in the box on the top right side and click on “Create Project”
6. The project will open in the project view, this is the basic interface you are going to work with: by default refine shows only 10 rows of data, you can change this on the bar above the data rows. Also you can use the navigation on the right to see the next or previous rows.

You now have successfully created your first Refine project. **Remember:** although it runs in a web-browser, the Refine server is still on your machine – all the data is there (so no worries if you handle sensitive information)

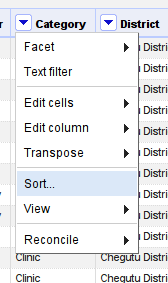
**Step 2: Sorting and Facetting**

Once we created our project, let’s go and explore the data and the Refine interface a bit. Using Refine might be intimidating at first, since it seems so different from spreadsheets, once you get used to it you will notice how easily you can do things with it.

One of the commonly used functions in spreadsheets is sorting and filtering data – to figure out minima, maxima or things about certain categories. Refine can do the same thing.

**Walkthrough:** Sorting rows

1. Refine handles data similar to a spreadsheet: you have rows, columns and cells – a cell is a field defined by a row and a column.
2. To sort your rows based on a specific column click on the small downward triangle next to the column

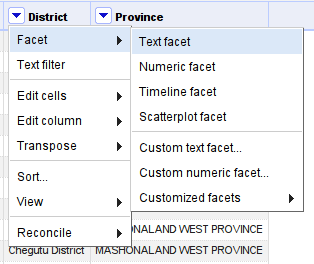


1. Select “Sort…” to open the sorting dialog
2. You can select what to sort the values as and then what order to sort in. (We’ll sort in text, since for now we only have text columns)
3. Click “OK” and your rows will be sorted based on the column
4. To undo the sort, click on the column options again, select “sort” then “remove sort”

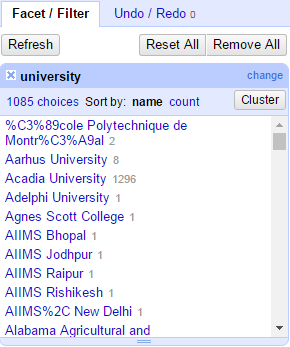
The other frequently used function in Spreadsheets is filtering – in Refine this is called facetting. Facetting in Refine is really powerful – you will see in most of the rest of the Recipe we’ll use facets.

### Walkthrough: Facetting rows based on a column

1. Select the column options for the column you want to facet with
2. Select “Facet”



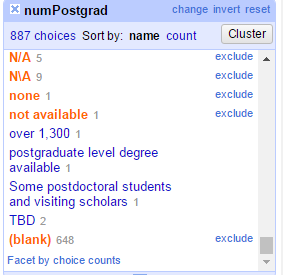
1. You can facet differently for text, numbers or dates – let’s facet as text – click on “Text facet”
2. This will open a facet in the left bar



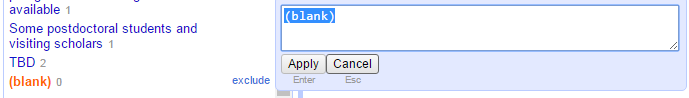
1. Now select one or more of the choices and you’ll see how your data rows are limited to just those selected.
2. Of course you can add more than one facet and thus filter more than once.

**Step 3 Dealing with Blank Cells and merging**

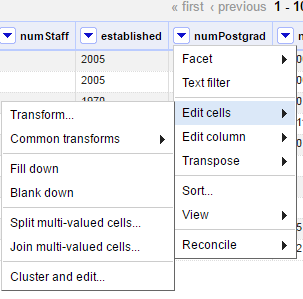
If you look closely at some facets, you’ll notice that on the bottom you have a selector saying “(blank)”. You’ll also see some duplicate options (e.g. N/A, N\A, none, etc.)

**Walkthrough:** Filling in the (blank)s

1. Choose the “(blank)” facet in your “numPostgrad” column
2. Hover over ‘edit’ and replace (blank) with NA (or whatever other value)



1. You can also edit individual values within the table on the right http://i0.wp.com/farm9.staticflickr.com/8506/8570737157_b168621858_d.jpg?w=591
2. Use the menu to navigate to ‘cluster and edit’ (note the other options here are also very useful)



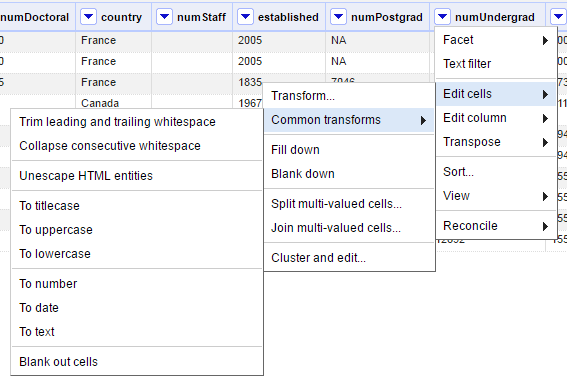
1. This allows you to merge multiple the multiple NA values using string matching – note that ‘none’ (i.e., 0) is not the same as NA (not available)
2. Play with the different options on the ‘country’ column and merge the values

**Step 4: Fighting the Invisible Man**

As illustrated in [The Invisible Man is in your Spreadsheets](http://schoolofdata.org/handbook/courses/data-cleaning-invisible-man-in-spreadsheets/) having spaces or newlines in your datafields is a problem. Since this is a very common problem, Refine has specific functions to remove whitespaces that shouldn’t be there.

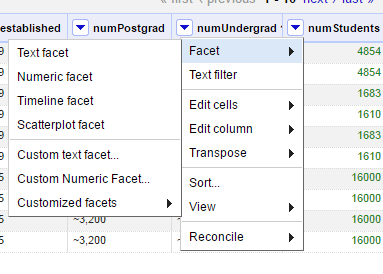
**Walkthrough:** Removing hidden whitespaces

1. An issue you may encounter in your data is that values *look* identical to you, while looking different to the computer
2. A common example of this is in the presence of ‘white space’: in a leading or trailing position (i.e., before or after another string); or with variable space lengths (e.g., a double space “ “ versus a single “ “)
3. Refine has a way to remove these
4. Refine can help you clean this up in an instant – open the column options for the “numUndergrad” column
5. Select “Edit Cells” – “Common Transforms” – “Collapse consecutive whitespace”

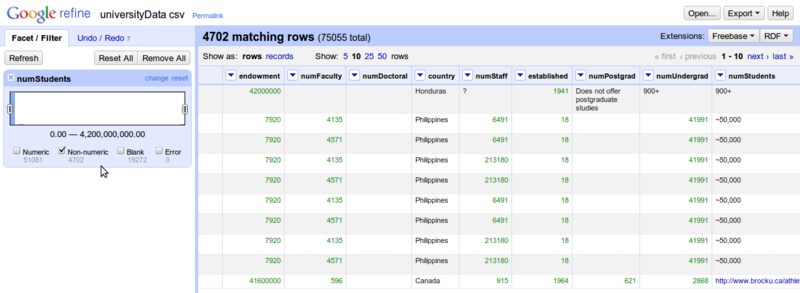


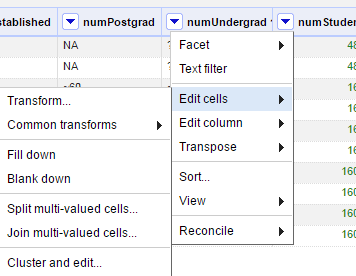
**Step 5: Advanced Clean up values for the number of students and removing bad data**

We need to clean the data for the number of students. Not all of the values are numeric, and many of them contain bits of text in addition to the actual number of the students.

To figure out which entries need to be fixed, we need to use a Numeric facet (you might need to do “Edit Cells -> Common Transformations -> To numeric” first):

This shows us a histogram of the values, and also lists the number of entries per type (numeric, non-numeric, blank, error, etc). Make sure that only the non-numeric rows are selected (see below)

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_SelectNonNumericValuesFromNumericFacet.png)



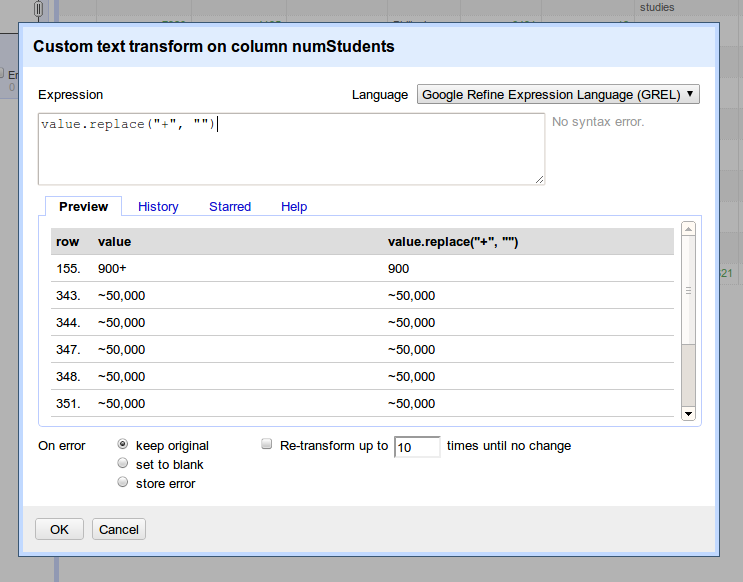
We can see some problems already, as some cells have "+" and "~" in them. To fix this, we need to do **Edit cells -> Transform**

This allows us to now type in commands that can replace sequences of characters:

value.replace("+", "")

Also, if you see entries with strange symbols like "Lumi%C3%A8re University Lyon 2" in the "x" column (should be "Lumière University Lyon 2"), you can fix this via **Edit cells -> Transform** with this command:

value.unescape('url')

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_TextTransform.png)

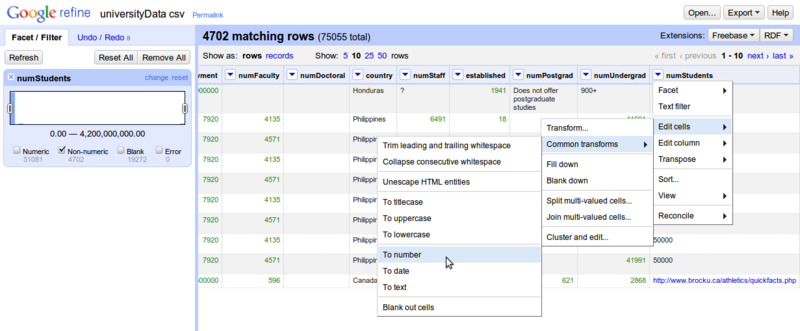
In doing this, you're actually using bits of a programming language. A lot of advanced features are available (not covered in this tutorial), and if you want to understand this further, you can refer to the [Google Refine Expression Language (GREL) reference](http://code.google.com/p/google-refine/wiki/DocumentationForUsers#Reference)

If you find multiple things that need to be replaced, you don't have to keep clicking **Edit cells -> Transform** for every single issue. You can chain these commands together to fix several issues at once:

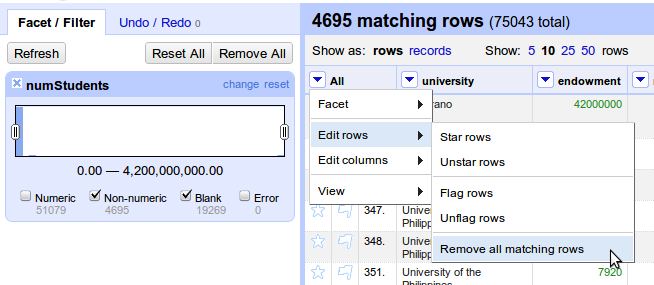
value.replace("~", "").replace(",","")

More issues can be cleaned up via:

value.replace(" total", "").replace(" -", "")

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_ConvertToNumber.png)In order to update the selection of non-numeric values, it's sometimes necessary to convert the values of the columns to numbers - **Edit cells -> Common transforms -> To number**. Once you do this, you should see that there are fewer non-numeric values.

You can continue cleaning up the data, but for this exercise we will move on and remove all the rows that do not have numeric values for the number of students. To do this, use a numeric facet again on numStudents to select only the non-numeric and blank values. Then do **All -> Edit rows -> Remove all matching rows**

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_RemoveAllMatchingRows.png)

**Step 6: Clean up values for the endowment**

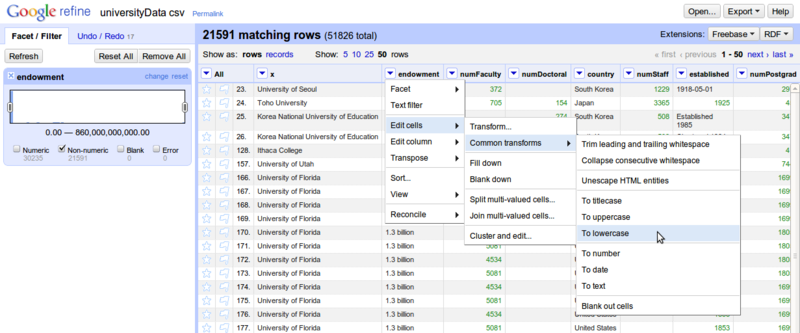
It's possible to have multiple facets in use at once. When you do this, each additional facet makes a sub-selection of the data selected by the previous facet. If you find that the number of rows you have selected and are working with is smaller than expected, then check to see if you still have facets in use which are not needed any more.

First remove the numeric facet for numStudents and create a new numeric facet for endowment. Select only the non-numeric values, as was done for the number of students.

Already we see issues like "US$1.3 billion" and "US $186 million"

Assuming that everything is in $ (a somewhat bogus assumption), we can clean up the data similarly to how we did it before. Click on the endowment column -> **Edit cells -> Transform**

value.replace("US $","").replace("US$", "")

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_ConvertToLowerCase.png)Both "million" and "Million" are in the values, so it's useful to convert all the values to lowercase instead of cleaning this up twice.

Click on the endowment column again, and create a custom text facet to locate all the rows with the word "million" in them: **Facet -> Custom text facet**

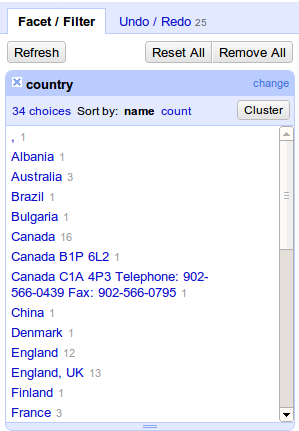
value.contains("million")

Then **Edit cells -> Transform**. It's not advisable to just replace "million" by "000000" since you have some values like "$13.8 million", which would be converted to "$13.8000000". It's better to first remove "million" from the text, convert the remaining text to a number, and then multiply this by 1000000:

toNumber(value.replace(" million", ""))\*1000000

The term "billion" is in the values as well, so remove previous facet for endowment, and create a new one for billion, and repeat process described above.

After most of this has been cleaned up, select the non-numeric values, and delete them, just as was done for the numStudents.

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_TextFacet.png)**Step 7: Looking for *more* issues in other columns**

OpenRefine has plenty of features that can help clean up the other columns as well. For example, if you do a text facet on the column with country names, you will find issues such as entries for both "England" and "England, UK", along with entries for Canada that contains parts of the university address.

How can you clean these using the methods above?

**Step 8: Cleaning up dates**

The dates are a mess as well, but there are a few techniques that can be used to help clean them up.

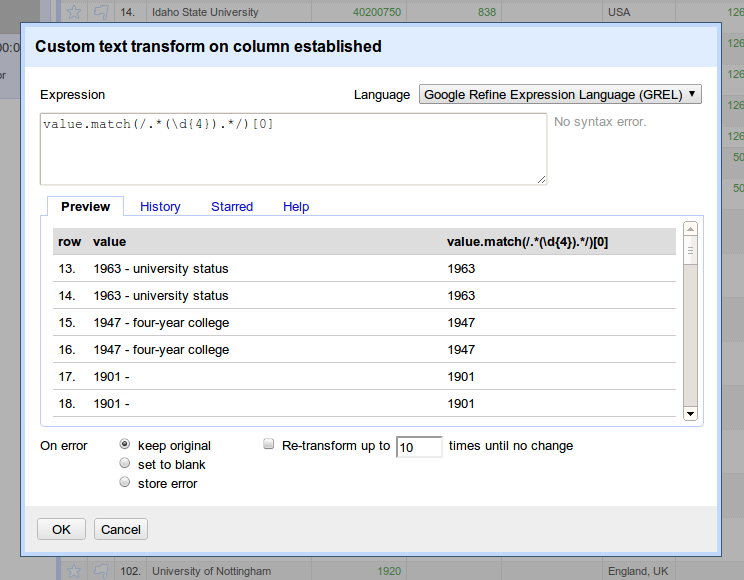
First we want to convert everything to text - **Edit cells -> Common transformations -> To text**, and then you need to **Edit cells -> Common transformations -> To date**. If you did not convert all the values to text first, then you may find that some of the years are represented as numbers, and have not been converted.

If only a year is listed, then the date created will use January 1st as the month and day. We will clean this up later to use only the year.

To further clean up the dates, we need to use **Facet -> Timeline facet** and select only "Non-Time" values. This shows us that we have a bit of a tricky situation as years are mixed in with text such as "Established 1985". We need some way to recognize a sequence of four numbers in a section of text and extract only the numbers. To do this, we need to use [regular expressions](http://code.google.com/p/google-refine/wiki/UnderstandingRegularExpressions). This is a very powerful technique that allows you to specify very complex patterns that you wish to match. For this tutorial, you don't need to know how to write regular expressions, but you should at least know that they exist, and that they can be used to help you with seemingly impossible tasks.

We now want to do **Edit cells -> Transform**, and use the code below. The ".\*" means a sequence of zero or more characters (letters, numbers, symbols, etc). The "\d" indicates that we're looking for a digit. The "{4}" shows that we want to match exactly 4 digits. The value.match function returns an array of results, so we use "[0]" to retrieve only the first match.

value.match(/.\*(\d{4}).\*/)[0]

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_CleanUpDates.png)

We can now convert these extracted values to dates - **Edit cells -> Common transformations -> To date**. At this point, we've done almost everything we can to track down usable dates, and we now want to just extract the years. To do this, we want to **Edit cells -> Transform** with the code below:

value.toString('yyyy')

What's happening here is that we're using a string ('yyyy' in this case) to specify what parts of the date we want, and how it should be displayed. The [documentation here](http://docs.oracle.com/javase/1.4.2/docs/api/java/text/SimpleDateFormat.html) describes this in much more detail. As illustrated in the table below, you can experiment with different commands to get different formats of dates.

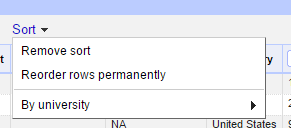
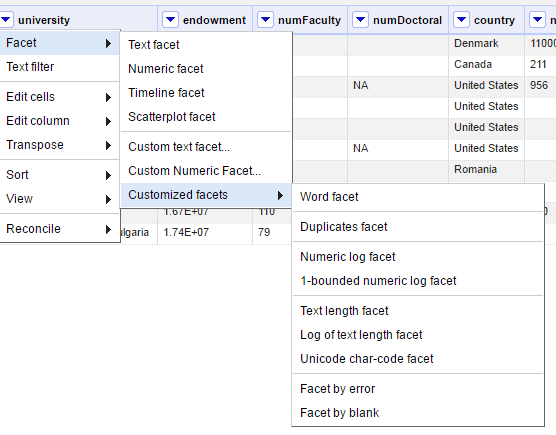
|  |  |
| --- | --- |
| **Command** | **Result** |
| value.toString('M') | 1 |
| value.toString('MM') | 01 |
| value.toString('MMM') | Jan |
| value.toString('MMMM') | January |

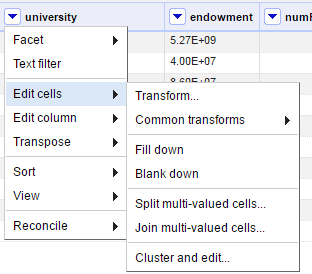
As described [here](http://code.google.com/p/google-refine/wiki/GRELDateFunctions), you can use code such as that below to reformat multiple date formats into a single format.

value.toDate('MM/yy','MMM-yy').toString('yyyy-MM')

**Step 8: Deduplicate entries ([near] duplicate rows)**

There are a lot of (nearly) duplicate rows in the data, this can happen for various reasons including human error and because multiple historic values are stored. We want to keep just one copy.

To do this (based on documentation [here](http://googlerefine.blogspot.nl/2011/08/remove-duplicate.html)), click on the column with the university names, and then click on "Sort". Once you do this, you will notice that there is a new "Sort" menu at the top. Click on this and select "Reorder rows permanently". This may take a while as it renumbers the rows in which the entries appear.

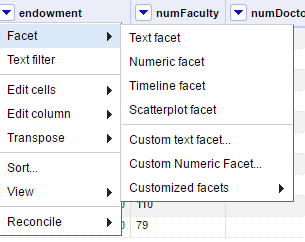


Then on the column with university names, **Edit cells -> Blank down**

Then on the same column, **Facet -> Customized facets -> Facet by blank**

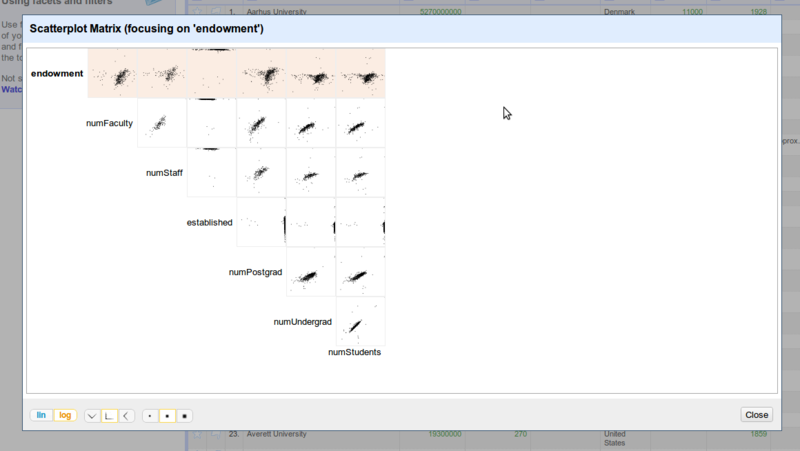
Now we want to remove all the blank rows, so select true, then on the "All" column on the left, Edit rows -> Remove all matching rows, like you have done when working with the numStudents and endowment columns.

Once you remove all the facets, and you now have a (mostly) cleaned data set.

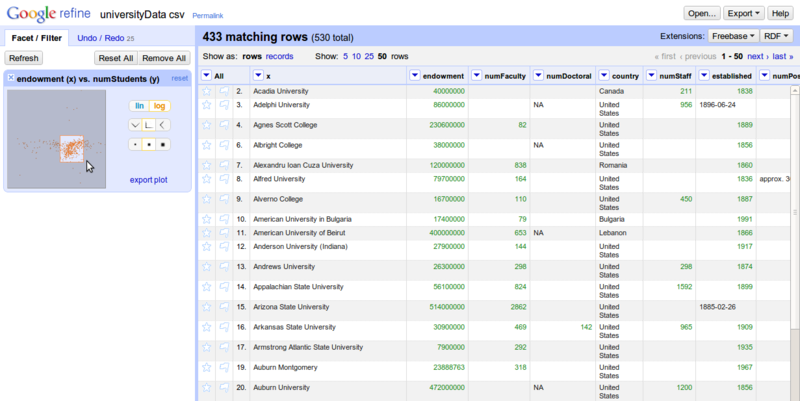
**Step 9: Exploring the data with scatter plots**

Click on the "endowment" column, **Facet -> Scatterplot facet**.

This shows the relationships between all of the numeric values in each of the columns. Click on "log" to get a better view.

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_ScatterplotFacet.png)

Click on the plot for endowment vs. numStudents. You can now drag select a portion of the plot, and then see the rows corresponding to that selection.

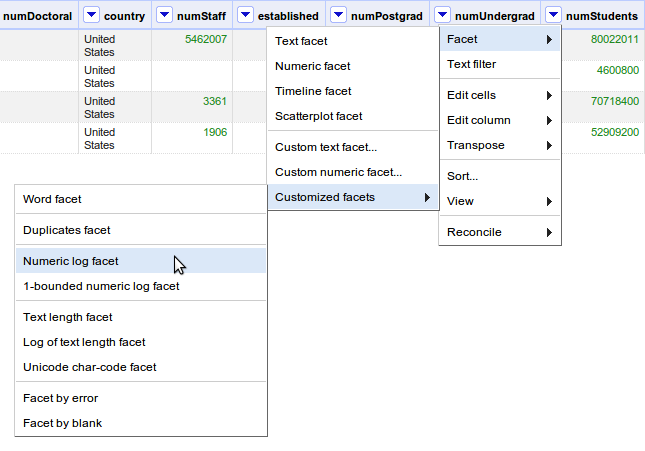
[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_SelectFromScatterplotFacet.png)

**Step 10: Geocoding names and addresses**

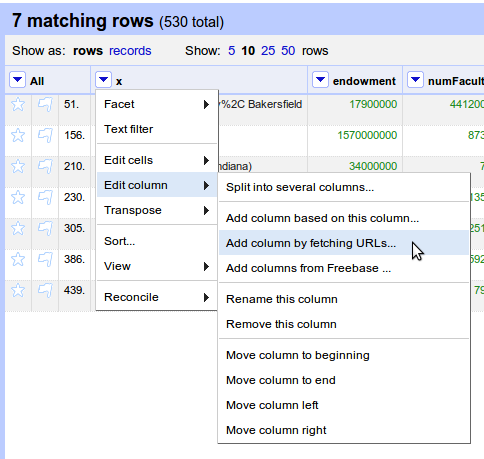
This next part shows (based on documentation [here](http://code.google.com/p/google-refine/source/browse/wiki/Geocoding.wiki?r=1342)) how to go from a description of a place (i.e. the name of a university) to values for its (likely) geographic coordinates. Behind the scenes, this uses Google Maps to figure out what is the most likely location you are asking for.

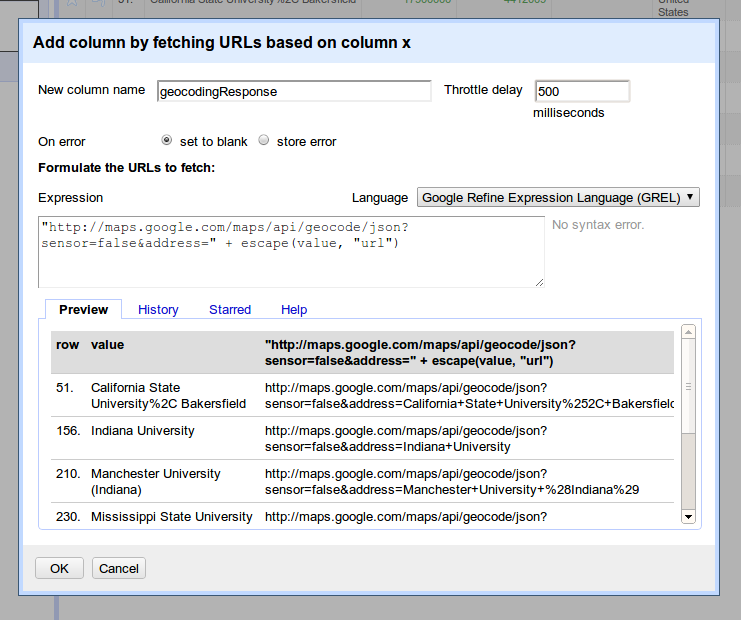
To learn how to do this, you don't need to do process the whole data set. This can take a while, and Google limits you to 2000 requests per day. It's better to just select around 10 rows and verify that it works.

An easy way to get a limited set of rows is by using a numeric log facet of the number of students, so use **Facet -> Customized facets -> Numeric log facet**

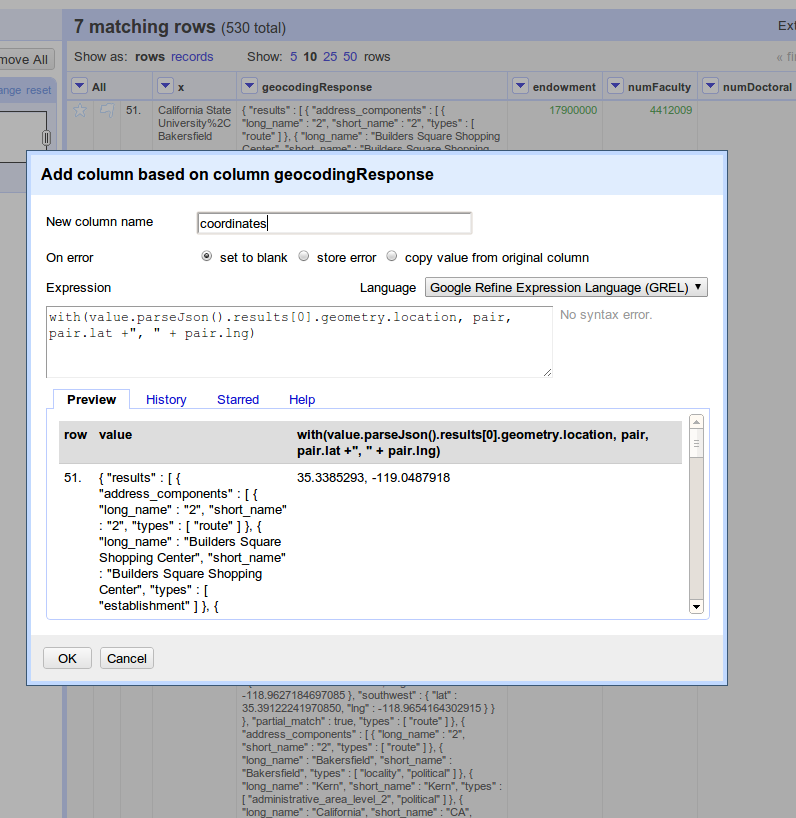
[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_NumericLogFacet.png)

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_NumberOfMatchingRows.png)Use this facet to make a selection of around ten rows, and then check the **matching rows** number to verify that you have a reasonable selection size:

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_AddColumnByFetchingURLs.png)Now the fun begins and we want to do **Edit column -> Add column by fetching URLs**. In other words, the values of the cells in the new column are based on data that is retrieved from the Internet.

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_FetchURLsExpression.png)Enter in the expression below, and you should see a list of URLs with the names of the universities at the end of the URLs. Specify a new column name such as "geocodingResponse", and set the throttle delay to around 500 milliseconds.

"http://maps.google.com/maps/api/geocode/json?sensor=false&address=" + escape(value, "url")

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_CleanUpGeocodingResponse.png)

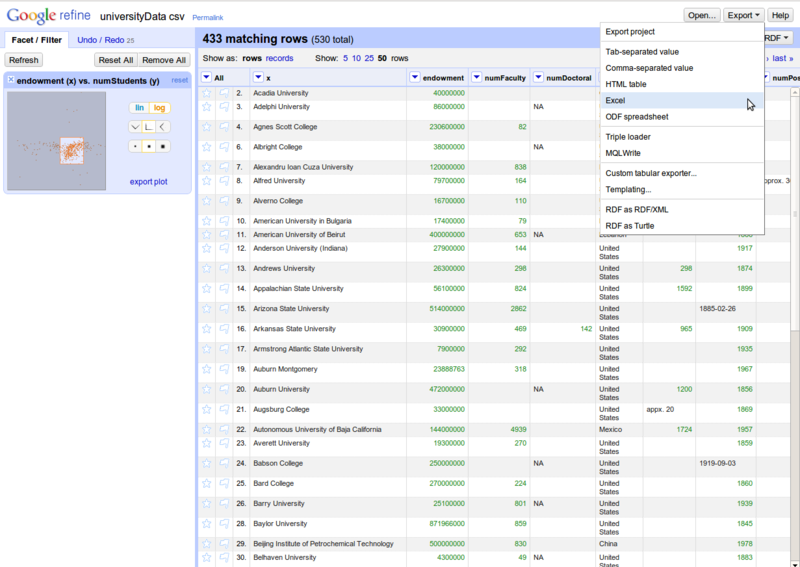
You should get a bunch of data back. To convert this into a more readable format, you need to click on the geocodingResponse column, and then on **Edit column -> Add column based on this column**. Enter in the expression below

with(value.parseJson().results[0].geometry.location, pair, pair.lat +", " + pair.lng)

Now you have a single column with coordinates. You can split this into columns for latitude and longitude by selecting **Edit Column -> Split into several columns** and specifying a separator of ",". These columns can then be renamed using **Edit Column -> Rename this column**.

**Step 11: Export Data**

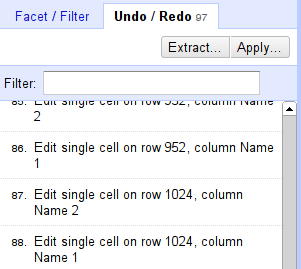
The data can be exported to formats such as Excel. If you read this into tools such as SPSS and notice that the last column is missing, then open the file up in Excel, re-save it, and try to open it up again in SPSS.

[](http://enipedia.tudelft.nl/wiki/File:GoogleRefine_ExportToExcel.png)

## Conclusions

**Congratulations – You successfully cleaned up a dataset using Refine!**

However there is even more you can do with Refine: For example did you notice how there is always a number next to the categories in the facet – telling you how many rows are in that category? By combining two facets, can you find out how many Universities with under 1000 students there are in the USA?

Once you made your transforms you might wonder: What if I made a mistake? Also if you work with data you generally want to keep track of what you did to the data. Since Refine was build with data processing in mind it keeps track of what you’re doing with your data and allows you to go back and forth in time. To see your history of changes click on the “Undo/Redo” tab on the left.

You see all the changes you made – by simply clicking on one of the steps you’ll be undoing all the changes after the step (don’t worry you can redo pretty much the same way). Play with this system until you are comfortable.

## Lessons learned

1. Creating projects in OpenRefine, sorting and faceting to explore data
2. Cleaning up inconsistent spelling of terms (i.e. "USA", "U.S.A", "U.S.", etc).
3. Converting values that are text descriptions of numeric values (i.e. $123 million) to actual numeric values (i.e. 123000000) which are usable for analysis.
4. Identifying which rows of a specific column contain a search term
5. Extracting and cleaning values for dates
6. Removing duplicate rows
7. Using a scatterplot to visualize relationships between values in different columns
8. Finding geographic coordinates for a list of place names (i.e. the names of universities, etc.)
9. Exporting cleaned data to Excel

## Acknowledgement

This resource is adapted from:

1. [**http://enipedia.tudelft.nl/wiki/OpenRefine\_Tutorial#Clean\_up\_country\_names**](http://enipedia.tudelft.nl/wiki/OpenRefine_Tutorial#Clean_up_country_names) **(under a CC-By-SA 3.0 license)** C.B.Davis, A. Chmieliauskas, G.P.J. Dijkema, I. Nikolic (2015), Enipedia, <http://enipedia.tudelft.nl>, Energy & Industry group, Faculty of Technology, Policy and Management, TU Delft, Delft, The Netherlands.
2. And <http://schoolofdata.org/handbook/recipes/cleaning-data-with-refine/>

Both under a CC-By-SA license. To cite, Knight, S., (2016). Cleaning Data with Refine,