# OpenRefine for Health Sciences: Part 1

## Learning Objectives

1. Summarize OpenRefine’s uses and applications
2. Import and export a data file successfully
3. Utilize data manipulation options available in OpenRefine’s dropdown menus, including facets, filters, and clustering
4. Apply Undo/Redo actions to data transformations

## Logistics/Housekeeping

Welcome to Session 1 of OpenRefine for Health Sciences. This is the first class in a two-part series focused on OpenRefine, today’s session is an Introduction to OpenRefine, and next week’s class will focus on Transformations. Today we have a couple instructors and helpers that are here to support your learning.

## Launching OpenRefine

*Based on* [*Library Carpentry: OpenRefine Lesson 2*](https://librarycarpentry.org/lc-open-refine/instructor/02-importing-data.html)

This class is based on material from The Carpentries which is known for live coding. If you were able to download and install OpenRefine on your device, we ask that you try to follow along if possible with the live coding.

Hopefully everyone was able to download and install OpenRefine. I am using version 3.9.3 on Windows. If you weren’t able to install OpenRefine, that’s okay! We encourage you to follow along as best you can and take notes of things to try on your own. The slide deck we shared has some key points we are covering about different features and tasks you can do in OpenRefine.

Now I’m going to launch OpenRefine. From the **blue diamond shortcut**.

This will open in a web browser, but remember it's not actually connected to the internet. It just uses the web browser as the interface. *NOTE: If OpenRefine does not open in a browser window, open your browser and type the address* [*127.0.0.1:3333*](https://127.0.0.1:3333/) *to take you to the OpenRefine interface.*

OpenRefine does not manipulate your data directly. Instead, the data you import and all the changes you make are stored in a project. You can stop working on a project and continue later if you like. When you want to ‘refine’ a new file, you start by creating a new project.

### Importing Data

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

* TSV (tab-separated values)
* CSV (comma-separated values)
* TXT
* Excel
* JSON (javascript object notation)
* XML (extensible markup language)
* Google Spreadsheet

Now to bring our dataset into OpenRefine, we select **Browse**, which opens the File Explorer in Windows. Select the dataset file from wherever you have it saved in your files. Click **Open**. Click **Next**.

The next screen gives you options to ensure the data is imported into OpenRefine correctly. We can see a preview of our dataset. The options vary depending on the type of data you are importing. We are working with an Excel file today, but if you are working with other file types in OpenRefine, you’ll want to check the options here in this preview pane.

Ensure the first row is used to create the column headings by checking the box **Parse next 1 line(s) as column headers**.

Always double check the preview to make sure your data is imported correctly before going to the next step. There is an "**Update Preview**" button on the lower right to refresh your view if you have changed a lot of options.

The **Project Name** box in the upper right corner will default to the title of your imported file. Click in the Project Name box to give your project a different name, if desired. I’m going to name this project: OpenRefine\_NNLM\_YYYYMMDD, with today’s date.

Once you have selected the appropriate options for your project, click the **Create project** button at the top right of the screen. This will create the project and open it for you. Projects are saved as you work on them, so there is no need to save copies as you go along.

## Layout and Display Options

*Based on* [*Library Carpentry: OpenRefine Lesson*](https://librarycarpentry.github.io/lc-open-refine/instructor/03-working-with-data.html) *3*

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. As with a spreadsheet, the individual bits of data live in ‘cells’ at the intersection of a row and a column.

OpenRefine only displays a limited number of rows of data at one time. You can adjust the number choosing between 5, 10 (the default), 25, 50, 100, 500, and 1000 at the top left of the table of data. Right now, it's set to only show 10 rows, but we can switch to look at more rows at one time or less.

*Click on Show 5. Click on Show 25.*

### **Rows vs. Records**

OpenRefine has two modes of viewing data: ‘Rows’ and ‘Records’. At the moment we are in Rows mode, where each row represents a single record in the dataset – in this case, an article. A row is a simple way to organize data: a series of cells, one cell per column. But sometimes there are multiple pieces of information in one cell, such as the **Author(s)** column where there are multiple authors listed.

If we toggle **records**, you will see from 3198 rows, it is now 3198 records. You can navigate through the records page by page by using the first/previous/next/last navigation options at the top right of the table of data. Also, in the menu at the top, there is an Undo/Redo option but we will get to that later in the lesson.

In Records mode, OpenRefine can link together multiple rows and show as belonging to the same Record. Rows will be assigned to Records based on the values in the first column.

See more [details of Rows and Records in the OpenRefine documentation](https://docs.openrefine.org/manual/exploring#rows-vs-records).

## Columns

*Based on* [*Library Carpentry: OpenRefine Lesson 6*](https://librarycarpentry.github.io/lc-open-refine/instructor/06-working-with-columns.html)

Most options to work with data in OpenRefine are accessed from drop down menus at the top of the data columns.

*Click the drop option on a random column to show options. Don’t click anything yet.*

When you select an option in a particular column (e.g. to make a change to the data), it will affect all the cells in that column. If you want to make changes across several columns, you will need to do this one column at a time.

### Reordering/Removing Columns

You can reorder the columns by clicking the drop-down menu at the top of the first column (labeled ‘All’), and choosing Edit columns 🡪 Re-order/remove columns…

You can then drag and drop column names to reorder the columns, or remove columns completely if they are not required. For example, I don’t need **Column 24** and **Column 25** – these were extra, empty columns.

For instance, I want to look at the **Date of Publication** column, but it’s all the way at the end of my dataset. I can drag this column between **Source** and **Title**. Click **OK**. And we see the **Date of Publication** column between **Source** and **Title**. I’m going to do the same for the **Date Accessed** column as we’ll be looking at both of our date columns more in Part 2.

### Renaming Columns

We can also rename a column by opening the drop-down menu at the top of the column that you would like to rename, and choosing Edit column 🡪 Rename column. You will then be prompted to enter the new column name.

Now I’m going to rename the **Data of Publication** column. On the **Date of Publication**, click the drop down. Click **Rename this column**. Type in **“Date Published”** and click **OK**. In next week’s session we’re going to learn how to transform a column of dates, to be recognized as dates by OpenRefine. For now, I’m going to just leave these column names.

### Sorting Data

If you’ve used Excel before, you may be familiar with being able to sort rows, based on a particular column. We can sort data in OpenRefine by clicking on the drop-down menu for the column you want to sort on, and choosing Sort. I’m going to **Sort** based on the **Title(s)** column.

I’m asked in the pop-up box, if I’m sorting cell values based on text, numbers, dates, or Booleans. Since this column is text data, I’m going to leave that checked, and you can see by default it will sort from A-Z. And I’m not going to worry about **Position blanks and errors**. Click **OK**.

Notice that this is only sorting based on the first character of each title, which for this dataset includes a single quotation (‘). My first row is now *'Needle-free' delivery of local anesthesia: a valuable option in pediatrics*. You should also see a “**Sort**” button at the top of the grid, with its own drop-down menu.

Unlike in Excel, “sorting” in OpenRefine is temporary – that is, if you remove the Sort, the data will go back to its original ‘unordered’ state. The **Sort** drop-down menu lets you amend the existing sort (e.g., reverse the sort order), remove existing sorts, and/or make sorts permanent. To make a sort permanent, choose **Reorder Rows Permanently** from the **Sort** drop-down menu.

#### Exercise #1: Sort Authors column

To make sure everyone is with me so far, let’s do a quick check. I’m now going to Sort based on the Author column. What is the article title on the third row from the top?

The article title is **[Monitoring airway pressure in pediatric anesthesia: an experimental model of intratracheal medication and pressure-volume loops]. [Spanish]**, and the first Author is **Abad Gurumeta A/Calvo Vecino JM…**

#### Practice Question #1: Sort Language and Publisher columns

1. Use Sort to on the **Language** and **Publisher** columns.
   * What is the first entry in the **Publisher** column?
   * What is the last entry in the **Language** column?
   * Hint: Did you **Remove sort** between checking each column?
2. Answers:
   * What is the first value in the **Publisher** column?
     + AANA Publishing Inc.
   * What is the last value in the **Language** column?
     + Ukrainian
3. Solution steps:
   * Scroll to the **Publisher** column, and use the drop-down menu
   * Click Sort…
   * Under “Sort cell values as”, select text and a - z
   * Click OK
     + The first value in the **Publisher** column is AANA Publishing Inc.
   * Repeat these steps on the **Language** column
   * Check the box for “sort by this column alone”
   * From the **Sort** drop-down menu (next to rows), scroll down to **By Language**, and select **Reverse**
     + The last value in the Language column is Ukrainian

### **Key Points**

* You can reorder, rename and remove columns in OpenRefine
* Sorting in OpenRefine always sorts all rows
* The original order of rows in OpenRefine is maintained during a sort until you use the option to Reorder Rows Permanently from the Sort drop-down menu

## Looking Ahead: Undo/Redo

*Based on* [*Library Carpentry: OpenRefine Lesson 9*](https://librarycarpentry.github.io/lc-open-refine/instructor/09-undo-and-redo.html)

In the last exercise, you could select **Reorder rows permanently** which would permanently sort based on the Language or Publisher column. But if I don’t want my rows sorted by a specific column, I can undo this action. OpenRefine lets you undo, and redo, any number of steps you have taken in cleaning the data. This means you can always try out things and ‘undo’ if you need to. The **Undo** and **Redo** options are accessed via the left-hand panel.

The way OpenRefine records the steps you have taken even allows you to take the steps you’ve carried out on one data set, and apply it to another dataset by a copy and paste operation. We’ll look at this at the end of today’s session.

The Undo/Redo panel lists all the steps you’ve taken so far. To undo steps, click on the last step you want to preserve in the list and this will automatically undo all the changes made following that step.

The remaining steps will continue to show in the list but grayed out, and you can reapply them by clicking on the last step you want to apply.

However, if you **‘undo’** a set of steps and then start doing new transformations, the grayed-out steps will disappear and you will no longer have the option to ‘redo’ these steps.

### **Key Points**

* You can use Undo and Redo to retrace ones’ steps
* You can save and apply a set of steps to a new set of data using the ‘Extract’ and ‘Apply’ features

## Facets

*Based on* [*Library Carpentry: OpenRefine Lesson 4*](https://librarycarpentry.github.io/lc-open-refine/instructor/04-faceting-and-filtering.html)

Facets are one of the most useful features of OpenRefine and can help in both getting an overview of the data and to improve the consistency of 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.

### **Text Facet**

One of the most commonly used facets is called a ‘Text facet’. This groups all the text values in a column and counts how many times each value appears. Sort of like a Pivot table in Excel.

We are going to do a **Facet** in the **Journal Name** column. To create a **Text Facet** for a column, click on the drop-down menu at the top of the **Journal Name** column and choose Facet 🡪 Text facet.

The facet will then appear in the left-hand panel. The facet consists of a list of values used in the data. Looking at the Facet, you can see there are **709** choices, or **709** unique **Journal Names**. You can opt to order them alphabetically, by **Name** or by **Count** of how often they appear.

*Order by ‘count’,* to see the most frequently occurring facets.

If you hover over one of the journal names, you can see the option to “**edit**” or “**include**”. You can include multiple values from the facet in a filter at one time by using the ‘**include’** option. You can also edit the values in the facet to change the value for several records at the same time. To do this, mouse-over the value you want to edit and click the **‘edit’** option that appears.

*Scroll down to* ***Anesthesia & Analgesia****. (anul jee ze uh)*

You can see that the journal **Anesthesia and Analgesia** is written two different ways. The one with “and” spelled out appears less often. Hover over the one with and written out and select **edit**.

Change the “and” to an ampersand, and click **Apply**. Now Anesthesia & Analgesia appears in the dataset **175** times. This approach is useful in datasets where you might have small variations through punctuation or typing errors etc.

Now we are going **Reorder** the facet by **count** to see the top three journals which are: 1) Paediatric Anaesthesia, 2) Anesthesia & Analgesia, and 3) Current Opinion in Anaesthesiology.

Now you are only looking at information about those three journals. There should be **787** rows. You can also ‘**invert’** the filter to show all records which do **not** match your selected values. This will then exclude the selected journal names.

*Hit reset.* This resets everything so all of the journal names are included again. We will do a practice exercise then you’ll do a facet on your own.

#### Exercise #2: Create a text facet

1. Click on the drop-down menu at the top of the **Publisher** column and choose Facet 🡪 Text facet. The facet will then appear in the left-hand panel
2. Include only publications from the American Academy of Pediatrics.
3. Also include American Epilepsy Society and American Roentgen Ray Society. To select multiple values, click the ‘include’ option on the appropriate line in the facet (which only appears when you mouse over the line).
4. You can ‘invert’ your selections to exclude those publishers.
5. Reset so everything is included once more.

#### Practice Question #2: Text Facet with Solution (Learner’s Turn)

Use a text facet on the **Country of Publication** column and answer these questions. How many countries are represented in this dataset? And how many articles were published in the United States?

1. Use a text facet on the **Country of Publication** column and answer these questions.
   * How many countries are represented in this dataset?
     + **64**
   * How many articles were published in the United States?
     + **858**
2. Solution steps.
   * Create a text facet on the **Country of Publication** column.
   * At the top, look how many choices there are.
   * Order the facet by name and find the United States.

### Other Types of Facets

As well as ‘Text facets’ OpenRefine also supports a range of other types of facets. These include:

* Numeric facets
* Timeline facets (for dates)
* Scatterplot facets
* Custom facets

Numeric and Timeline facets display graphs instead of lists of values. In the next session, we will look at a Timeline facet. Scatterplot facets are less commonly used. For further information on these see this [OpenRefine tutorial in the Wayback Machine](https://web.archive.org/web/20190105063215/http:/enipedia.tudelft.nl/wiki/OpenRefine_Tutorial#Exploring_the_data_with_scatter_plots).

**Customized facets** are a range of different types of facets. Some of the default customized facets are duplicate facets and faceting by blank or null.

#### **Duplicates Facet**

Say for example we want to see how many DOIs are duplicates, we would use this customized facet. Select for the **DOI** column, Facet 🡪 Customized facets 🡪 Duplicates facet

This results in a binary facet of ‘true’ or ‘false’. Rows appear in the “true” facet if the value in the selected column is an exact match for a value in the same column in another row. There should be **788** “true” or **788 duplicates**. This is good for us to know because we may want to remove duplicate articles before continuing analyzing our data.

#### Facet by Blank

There is also **Facet by null** – again this is a binary facet, so it’s either ‘true’ or ‘false’. Rows appear as “true” if they have no data present in that column. This is useful when looking for rows missing key data.

In this dataset, we probably want to make sure we have DOIs for all our articles, so let's see how many blanks we have.

Select for the **DOI** column, Facets 🡪 Customized facets 🡪 Facets by blank.

Now there are **631** “true” meaning **631 records do not have a DOI**. And if we select include, we can look through those records and see that no – they do not have a listed DOI.

These kinds of facets can help you quickly determine if you have missing or duplicated information. Particularly when working with larger datasets like this. So, we have another practice problem for you.

##### **Practice Question #3 Facet by Blank**

Use the Facet by blank function to find all publications in this data set without an Abstract. How many records do not have an abstract or are marked “true”?

1. **Solutions Steps**
   1. Use the Facet by blank function to find all publications in this data set without an abstract.
      * Facets 🡪 Customized facets 🡪 Facet by null
   2. How many are true? Or how many are blank?
      * True means that it is blank. There should be **787 “true” with no abstracts**.
   3. Select include on True in the facet to filter the list of publications to only those that don’t have an abstract.
   4. Make sure to reset so you are still looking at the entire dataset before moving on to the next step.

## Filters

*Based on* [*Library Carpentry: OpenRefine Lesson 4*](https://librarycarpentry.github.io/lc-open-refine/instructor/04-faceting-and-filtering.html)

### **Text Filter**

As well as using Facets to filter the data displayed in OpenRefine you can also apply Text filters which searches for a particular piece of text appearing in a column based on a unique text string, like a ‘find’ feature. Text filters are applied by clicking the drop-down menu at the top of the column you want to apply the filter to and choosing **Text filter**.

As with **Facets**, the **Filter** options appear in the left-hand panel. We’ll do a filter on **Publication Type**. In the drop down on the **Publication Type** column, select Text filter. Type “conference”. We should get 382 matching records, so 382 of these citations were from a conference based on their publication type.

As you can see, OpenRefine only displays rows that contain that text in the relevant column. You can also invert your selection, so if you click on invert, you can look at the 2807 records that were **not** from conferences.

Make sure you **reset** everything so you have 3189 records again.

### Working With Filtered Data

It is very important to note that when you have filtered the data displayed in OpenRefine, any operations you carry out will apply only to the rows that match the filter – that is the data currently being displayed. To confirm you are working with the data you intended to select, check the number of matching records displayed above the data table.

Reset everything so you have 3189 records again.

#### Exercise #3: Create a filter

1. Click on the drop-down menu at the top of the MeSH Subject Headings column and choose > Text Filter. The facet will then appear in the left-hand panel.
2. Type in “human”.
   1. This should give you 2708 matching rows.
   2. Select **invert**.
      * This should give you 481 matching rows.
   3. Remember to reset or close the filter to return to having 3189 rows before going on to the next steps.

### Key Points

* You can use facets and filters to explore your data
* You can use facets and filters work with a subset of data in OpenRefine
* You can correct common data issues from a Facet

## Clustering

*Based on* [*Library Carpentry: OpenRefine Lesson 5*](https://librarycarpentry.github.io/lc-open-refine/instructor/05-clustering.html)

The Cluster feature groups together similar, but inconsistent values in a given column and lets you merge them into a single, consistent value. This is very effective where you have data with minor variations in data values, e.g. names of people, organizations, places, classification terms. Think of journals that may or may not start with ‘the’, or all the different ways you can abbreviate United States.

To use the ‘cluster’ function, click on the **Edit cells** menu option in the relevant column and choose **Cluster and edit**. We’ll try clustering on the **Publisher** column today.

Select Publisher column 🡪 Edit cells 🡪 Cluster and edit.

**‘Clusters’** are created automatically according to an algorithm. OpenRefine supports a number of different clustering algorithms – some experimentation may be required to see which clustering algorithm works best with any particular set of data. You may find that using different algorithms highlights different clusters.

For more information on the methods used to create clusters, see [OpenRefine’s Technical Reference on Clustering in Depth](https://openrefine.org/docs/technical-reference/clustering-in-depth).

Select **Key collision** for method. Select **n-Gram fingerprint** for **Keying Function**. Click **Cluster**.

For each cluster, you have the option of ‘merging’ the values together – that is, replace the various inconsistent values with a single consistent value. By default, OpenRefine uses the most common value in the cluster as the new value, but you can select another by clicking the value itself, or you can type the desired value into the **New Cell Value** box.

Check **Merge** for **BioMed** and **Institute of Medico**. Click Merge selected & re-cluster.

Try the **Metaphone3** keying function. There are a lot of options for Lippincott Williams and Wilkins. You type in the value you want everything to be merged to. So ‘Lippincott Williams and Wilkins’. Check the **Merge?** box and click Merge selected & Close.

You always want to double check the clusters before merging them. OpenRefine is pretty good at clustering, but sometimes it will get it wrong, so you want to look at the clusters closely.

### Key Points

* Clustering is a way of finding variant forms of the same piece of data within a dataset (e.g. different spellings of a name)
* There are a number of different Clustering algorithms that work in different ways and will produce different results
* The best clustering algorithm to use will depend on the data
* Using clustering you can replace varying forms of the same data with a single consistent value

## Export (Final Steps for Part 1)

*Based on* [*Library Carpentry: OpenRefine Lesson 12*](https://librarycarpentry.github.io/lc-open-refine/instructor/12-export-transformation.html)

### Exporting History of Steps – Potentially skip for time

We looked at the **Undo/Redo** tab before, and how we can undo or go back to a previous version of the dataset. We can also save a record of all the steps we’ve done on the data so far. This could be useful if you are cleaning or formatting data the same way on several spreadsheets.

If you wish to save a set of steps to be re-applied later, you can save the steps and quickly **Reapply** all the modifications you’ve made using the **Extract** option.

*Click the* ***Extract*** *button.*

This gives you the option to select steps that you want to save, and extract the code for those steps in a format called **‘JSON’** – which stands for **JavaScript Object Notation**. You can copy the extracted JSON and save it as a plain text file (e.g. in Notepad).

To apply a set of steps you have copied or saved in this JSON format use the Apply button and paste in the JSON. In this way you can repeat the transformations you’ve done on one dataset to another dataset, and between projects, or with other people.

**Undo/Redo** data is stored with the Project and is saved automatically as you work, so next time you open the project, you can access your full history of steps you have carried out and undo/redo in exactly the same way.

### Exporting Data

Now we have finished working with our data set for today. And it’s time to save and export. 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.

*Click* ***Export*** *and select the file format for saving the dataset.*

#### Exporting Portion of Your Data

You can also export a portion of your data by using facets or filters to select a portion of your data. With only those rows selected, you can select the export format and your resulting file will only include the select rows.

***Note****:* It’s easy to export only a portion of your data by accident, so make sure you look at the top left to ensure all rows are being displayed when you want to do a full export.

### **Key Points**

* You can export your data in a variety of formats