Data Cleaning with OpenRefine for Ecologists (./)

Introduction

❓ Overview

Teaching: 10 min
Exercises: 0 min
Questions

- What is OpenRefine useful for?

Objectives

- Describe OpenRefine’s uses and applications.
- Differentiate data cleaning from data organization.
- Experiment with OpenRefine’s user interface.
- Locate helpful resources to learn more about OpenRefine.

Lesson

Motivations for the OpenRefine Lesson

- It is important to know what you did to your data. Additionally, journals, granting agencies, and other institutions are requiring documentation of the steps you took when working with your data. With OpenRefine, you can capture all actions applied to your raw data and share them with your publication as supplemental material.
- You *must* export your modified dataset to a new file: OpenRefine does not write back into your original sources. If you don’t save it, your OpenRefine work will be lost.
- Any operation that changes the data in OpenRefine can be easily reversed or undone.
- Data is often very messy, and this tool saves a lot of time on cleaning headaches.
- Data cleaning steps often need repeating with multiple files. OpenRefine is perfect for speeding up repetitive tasks by replaying previous actions on multiple datasets.
- Some concepts such as clustering algorithms are quite complex, but OpenRefine makes it easy to introduce them, use them, and show their power.

Before we get started

The following setup is necessary before we can get started (instructions here (../setup.html).

Do you need help with any of the following?

- Mozilla Firefox or Google Chrome is set as your default browser.
- Download and install software from http://openrefine.org/download.html
- Download this data file (https://ndownloader.figshare.com/files/7823341) and save to your desktop

- If after installation and running OpenRefine, it does not automatically open for you, point your browser at http://127.0.0.1:3333/ (http://127.0.0.1:3333/) or http://localhost:3333/ (http://localhost:3333/) to launch the program.

What is OpenRefine?

- OpenRefine is a Java program that runs on your machine (not in the cloud): it is a desktop application that uses your web browser as a graphical interface. No internet connection is needed, and none of the data or commands you enter in OpenRefine are sent to a remote server.
- OpenRefine does not modify your original dataset. All actions are easily reversed in OpenRefine and you can capture all the actions applied to your data and share this documentation with your publication as supplemental material.OpenRefine saves as you go. You can return to the project at any time to pick up where you left off, or export your data to a new file.
- OpenRefine can be used to standardise and clean data across your file.

It can help you:

- Get an overview of a data set
- Resolve inconsistencies in a data set
- Help you split data up into more granular parts
- Match local data up to other data sets
- Enhance a data set with data from other sources
- Save a set of data cleaning steps to replay on multiple files

OpenRefine is a powerful, free and open source tool with a large growing community of practice. More help can be found at http://openrefine.org (http://openrefine.org).

Basics of OpenRefine

You can find out a lot more about OpenRefine at http://openrefine.org (http://openrefine.org) and check out some great introductory videos (https://www.youtube.com/channel/UCqwSVsJ8CWD9pQUZDbJC1ew). There is a Google Group (https://groups.google.com/forum/?hl=en#!forum/openrefine) that can answer a lot of beginner questions and problems. There is also an OpenRefine Google Plus community (https://plus.google.com/communities/117280693504889048168) where you can find a lot of help, especially from community members from the life sciences. OpenRefine recipes (https://github.com/OpenRefine/OpenRefine/wiki/Recipes), scripts, projects, and extensions are available too, where you can find and copy them into your OpenRefine instance to run on your dataset.

The OpenRefine GitHub wiki page has a reference (https://github.com/OpenRefine/OpenRefine/wiki/GREL-Functions) of the General Refine Expression Language (GREL).

Features

- Open source (source on GitHub (https://github.com/OpenRefine/OpenRefine)).
- A large growing community, from novice to expert, ready to help.
- Works with large-ish datasets (100,000 rows). Does not scale to many millions. (yet).

Key Points

- OpenRefine is a powerful, free and open source tool that can be used for data cleaning.
- OpenRefine will automatically track any steps you take in working with your data.

Working with OpenRefine

Overview

Teaching: 15 min
Exercises: 20 min
Questions

- How can we bring our data into OpenRefine?
- How can we sort and summarize our data?
- How can we find and correct errors in our raw data?

Objectives

- Create a new OpenRefine project from a CSV file.
- Recall what facets are and how they are used to sort and summarize data.
- Recall what clustering is and how it is applied to group and edit typos.
- Manipulate data using previous steps with undo/redo.
- Employ drop-downs to split values from one column into multiple columns.
- Employ drop-downs to remove white spaces from cells.

Lesson

Creating a Project

Start the program. (Double-click on the openrefine.exe file (or google-refine.exe if using an older version). Java services will start on your machine, and OpenRefine will open in your browser).

Launch OpenRefine (see Getting Started with OpenRefine (http://www.datacarpentry.org/OpenRefine-ecology-lesson/00-getting-started/)).

OpenRefine can import a variety of file types, including tab separated (`tsv`), comma separated (`csv`), Excel (`xls` , `xlsx`), JSON, XML, RDF as XML, Google Spreadsheets. See the OpenRefine Importers page (https://github.com/OpenRefine/OpenRefine/wiki/Importers) for more information.

In this first step, we'll browse our computer to the sample data file for this lesson. In this case, we modified the `Portal_rodents` CSV file, adding several columns: `scientificName` , `locality` , `county` , `state` , `country` and generating several more columns in the lesson itself (`JSON` , `decimalLatitude` , `decimalLongitude`). Data in `locality` , `county` , `country` , `JSON` , `decimalLatitude` and `decimalLongitude` are contrived and are in no way related to the original dataset.

If you haven't already, download the data from:
https://ndownloader.figshare.com/files/7823341 (https://ndownloader.figshare.com/files/7823341)

Once OpenRefine is launched in your browser, the left margin has options to `Create Project` , `Open Project` , or `Import Project` . Here we will create a new project:

1. Click `Create Project` and select `Get data from  This Computer` .
2. Click `Choose Files` and select the file `Portal_rodents_19772002_scinameUUIDs.csv` . Click `Open` or double-click on the filename.
3. Click `Next>>` under the browse button to upload the data into OpenRefine.

4. OpenRefine gives you a preview - a chance to show you it understood the file. If, for example, your file was really tab-delimited, the preview might look strange, you would choose the correct separator in the box shown and click `Update Preview` (bottom right). If this is the wrong file, click `<<Start Over` (upper left).
5. If all looks well, click `Create Project>>` (upper right).

Note that at step 1, you could upload data in a standard form from a web address by selecting `Get data from Web Addresses (URLs)`. However, this won't work for all URLs.

Faceting

Exploring data by applying multiple filters

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

- seeing a big picture of your data, and
- filtering down to just the subset of rows that you want to change in bulk.

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

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

Here we will use faceting to look for potential errors in data entry in the `scientificName` column.

1. Scroll over to the `scientificName` column.
2. Click the down arrow and choose `Facet > Text facet`.
3. In the left panel, you'll now see a box containing every unique value in the `scientificName` column along with a number representing how many times that value occurs in the column.
4. Try sorting this facet by name and by count. Do you notice any problems with the data? What are they?
5. Hover the mouse over one of the names in the `Facet` list. You should see that you have an `edit` function available.
6. You could use this to fix an error immediately, and OpenRefine will ask whether you want to make the same correction to every value it finds like that one. But OpenRefine offers even better ways to find and fix these errors, which we'll use instead. We'll learn about these when we talk about clustering.

Solution

There will be several near-identical entries in `scientificName`. For example, there is one entry for `Ammospermophilis harrisi` and one entry for `Ammospermophilus harrisii`. These are both misspellings of `Ammospermophilus harrisi`. We will see how to correct these misspelled and mistyped entries in a later exercise.

📌 More on Facets

OpenRefine Wiki: Faceting (https://github.com/OpenRefine/OpenRefine/wiki/Faceting)

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

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

Numeric and Scatterplot facets display graphs instead of lists of values. The numeric facet graph includes ‘drag and drop’ controls you can use to set a start and end range to filter the data displayed. These facets are explored further in Examining Numbers in OpenRefine (http://www.datacarpentry.org/OpenRefine-ecology-lesson/03-numbers/)

Custom facets are a range of different types of facets. Some of the default custom facets are:

- Word facet - this breaks down text into words and counts the number of records each word appears in
- 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
- Text length facet - creates a numeric facet based on the length (number of characters) of the text in each row for the selected column. This can be useful for spotting incorrect or unusual data in a field where specific lengths are expected (e.g. if the values are expected to be years, any row with a text length more than 4 for that column is likely to be incorrect)
- Facet by blank - a binary facet of ‘true’ or ‘false’. Rows appear in the ‘true’ facet if they have no data present in that column. This is useful when looking for rows missing key data.

Facets are intended to group together common values and OpenRefine limits the number of values allowed in a single facet to ensure the software does not perform slowly or run out of memory. If you create a facet where there are many unique values (for example, a facet on a ‘book title’ column in a data set that has one row per book) the facet created will be very large and may either slow down the application, or OpenRefine will not create the facet.

Exercise

1. Using faceting, find out how many years are represented in the census.
2. Is the column formatted as Number, Date, or Text? How does changing the format change the faceting display?
3. Which years have the most and least observations?

Solution

1. For the column `yr` do `Facet > Text facet` . A box will appear in the left panel showing that there are 26 unique entries in this column.
2. By default, the column `yr` is formatted as Text. You can change the format by doing `Edit cells > Common transforms > To number` . Doing `Facet > Numeric facet` creates a box in the left panel that shows a histogram of the number of entries per year. Notice that the data is shown as a number, not a date. If you instead transform the column to a date, the program will assume all entries are on January 1st of the year.
3. After creating a facet, click `Sort by count` in the facet box. The year with the most observations is 1997. The least is 1977.

Clustering

In OpenRefine, clustering means “finding groups of different 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. Likewise, `Gödel` and `Godel` probably refer to the same person. Clustering is a very powerful tool for cleaning datasets which contain misspelled or mistyped entries. OpenRefine has several clustering algorithms built in. Experiment with them, and learn more about these algorithms and how they work.

1. In the `scientificName` Text Facet we created in the step above, click the `Cluster` button.
2. In the resulting pop-up window, you can change the `Method` and the `Keying Function` . Try different combinations to see what different mergers of values are suggested.
3. Select the `key collision` method and `metaphone3` keying function. It should identify three clusters.
4. Click the `Merge?` box beside each, then click `Merge Selected and Recluster` to apply the corrections to the dataset.
5. Try selecting different `Methods` and `Keying Functions` again, to see what new merges are suggested. You may find there are still improvements that can be made, but don’t `Merge` again; just `Close` when you’re done. We’ll now see other operations that will help us detect and correct the remaining problems, and that have other, more general uses.

Important: If you `Merge` using a different method or keying function, or more times than described in the instructions above, your solutions for later exercises will not be the same as shown in those exercise solutions.

More on clustering (https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth)

Split

If data in a column needs to be split into multiple columns, and the parts are separated by a common separator (say a comma, or a space), you can use that separator to divide up the pieces into their own columns.

1. Let us suppose we want to split the `scientificName` column into separate columns for genus and for species.
2. Click the down arrow at the top of the `scientificName` column. Choose `Edit Column > Split into several columns...`
3. In the pop-up, in the `Separator` box, replace the comma with a space.
4. Uncheck the box that says `Remove this column`.
5. Click `OK`. You'll get some new columns called `scientificName 1`, `scientificName 2`, and so on.
6. Notice that in some cases `scientificName 1` and `scientificName 2` are empty. Why is this? What do you think we can do to fix this?

Solution

The entries that have data in `scientificName 3` and `scientificName 4` but not the first two `scientificName` columns had an extra space at the beginning of the entry. Leading white spaces are very difficult to notice when cleaning data manually. This is another advantage of using OpenRefine to clean your data. We'll look at how to fix leading and trailing white spaces in a later exercise.

Exercise

Try to change the name of the second new column to "species". How can you correct the problem you encounter?

Solution

On the `scientificName 2` column, click the down arrow and then `Edit column > Rename this column`. Type "species" into the box that appears. A pop-up will appear that says `Another column already named species`. This is because there is another column where we've recorded the species abbreviation. You can choose another name like `speciesName` for this column or change the other `species` column name to `speciesAbbreviation`.

Undo / Redo

It's common while exploring and cleaning a dataset to discover after you've made a change that you really should have done something else first. OpenRefine provides `Undo` and `Redo` operations to make this easy.

1. Click where it says `Undo / Redo` on the left side of the screen. All the changes you have made so far are listed here.
2. Click on the step that you want to go back to, in this case the previous step. The added columns will disappear.
3. Notice that you can still click on the last step and make the columns reappear, and toggle back and forth between these states.
4. Leave the dataset in the state in which the `scientificNames` were clustered, but not yet split.

Important: If you skip this step, your solutions for later exercises will not be the same as shown in those exercise solutions.

Trim Leading and Trailing Whitespace

Words with spaces at the beginning or end are particularly hard for we humans to tell from strings without, but the blank characters will make a difference to the computer. We usually want to remove these. OpenRefine provides a tool to remove blank characters from the beginning and end of any entries that have them.

1. In the header for the column `scientificName` , choose `Edit cells > Common transforms > Trim leading and trailing whitespace` .
2. Notice that the `Split` step has now disappeared from the `Undo / Redo` pane on the left and is replaced with a `Text transform on 3 cells`
3. Perform the same `Split` operation on `scientificName` that you undid earlier. This time you should only get two new columns. Why?

Solution

Removing the leading white spaces means that each entry in this column has exactly one space (between the genus and species names). Therefore, when you split with space as the separator, you will get only two columns.

Important: `Undo` the splitting step before moving on to the next lesson. If you skip this step, your solutions for later exercises will not be the same as shown in those exercise solutions.

Key Points

- Faceting and clustering approaches can identify errors or outliers in data.

Filtering and Sorting with OpenRefine

Overview

Teaching: 10 min
Exercises: 10 min
Questions

- How can we select only a subset of our data to work with?
- How can we sort our data?

Objectives

- Employ *text filter* or *include/exclude* to filter to a subset of rows.
- Sort tables by a column.
- Sort tables by multiple columns.

Lesson

Filtering

There are many entries in our data table. We can filter it to work on a subset of the data in the list for the next set of operations. Please ensure you perform this step to save time during the class.

1. Click the down arrow next to `scientificName > Text filter` . A `scientificName` facet will appear on the left margin.

2. Type in `bai` and press return. There are 48 matching rows of the original 35549 rows (and these rows are selected for the subsequent steps).
3. At the top, change the view to `Show 50 rows`. This way you will see all the matching rows.

Exercise

1. What scientific names (genus and species) are selected by this procedure?
2. How would you restrict this to one of the species selected?

Solution

1. Do `Facet > Text facet` on the `scientificName` column after filtering. This will show that two names match your filter criteria. They are `Baiomys taylori` and `Chaetodipus baileyi`.
2. To restrict to only one of these two species, you could make the search case sensitive or you could split the `scientificName` column into species and genus before filtering or you could include more letters in your filter.

Excluding entries

In addition to the solutions included above, another way to narrow our filter is to `include` and/or `exclude` entries in a facet. If you still have your facet for `scientificName`, you can use it, or use drop-down menu > `Facet > Text facet` to create a new facet. Only the entries with names that agree with your `Text filter` will be included in this facet.

Faceting and filtering look very similar. A good distinction is that faceting gives you an overview description of all of the data that is currently selected, while filtering allows you to select a subset of your data for analysis.

Exercise

Use `include / exclude` to select only entries from one of these two species.

Solution

1. In the facet (left margin), click on one of the names, such as `Baiomys taylori`. Notice that when you click on the name, or hover over it, there are entries to the right for `edit` and `include`.
2. Click `include`. This will explicitly include this species, and exclude others that are not expicitly included. Notice that the option now changes to `exclude`.
3. Click `include` and `exclude` on the other species (`Chaetodipus baileyi`) and notice how the two entries appear and disappear from the table.

Important: Select both species for your filtered dataset before continuing with the rest of the exercises.

Sort

You can sort the data in a column by using the drop-down menu available in that column. There you can sort by `text`, `numbers`, `dates` or `booleans` (`TRUE` or `FALSE` values). You can also specify what order to put `Blanks` and `Errors` in the sorted results.

If this is your first time sorting this table, then the drop-down menu for the selected column shows `Sort...`. Select what you would like to sort by (such as `numbers`). Additional options will then appear for you to fine-tune your sorting.

✎ Exercise

Sort by month. How can you ensure that months are in order?

If you try to re-sort a column that you have already used, the drop-down menu changes slightly, to > `Sort` without the `...`, to remind you that you have already used this column. It will give you additional options:

- `Sort` > `Sort...` - This option enables you to modify your original sort.
- `Sort` > `Reverse` - This option allows you to reverse the order of the sort.
- `Sort` > `Remove sort` - This option allows you to undo your sort.

✎ Exercise

Sort the data by `plot`. What year(s) were observations recorded for plot 1 in this filtered dataset.

👁 Solution

In the `plot` column, select `Sort...` > `numbers` and select `smallest first`. The years represented are 1990 and 1995.

Sorting by multiple columns.

You can sort by multiple columns by performing sort on additional columns. The sort will depend on the order in which you select columns to sort. To restart the sorting process with a particular column, check the `sort by this column alone` box in the `Sort` pop-up menu.

Exercise

You might like to look for trends in your data by month of collection across years.

1. How do you sort your data by month?
2. How would you do this differently if you were instead trying to see all of your entries in chronological order?

Solution

1. For the `mo` column, click on `Sort...` and then `numbers`. This will group all entries made in, for example, January, together, regardless of the year that entry was collected.
2. For the `yr` column, click on `Sort > Sort... > numbers` and select `sort by this column alone`. This will undo the sorting by month step. Once you've sorted by `yr` you can then apply another sorting step to sort by month within year. To do this for the `mo` column, click on `Sort > numbers` but do not select `sort by this column alone`. To ensure that all entries are shown chronologically, you will need to add a third sorting step by day within month.

If you go back to one of the already sorted columns and select > `Sort > Remove sort`, that column is removed from your multiple sort. If it is the only column sorted, then data reverts to its original order.

Exercise

Sort by `year`, `month` and `day` in some order. Be creative: try sorting as `numbers` or `text`, and in reverse order (`largest to smallest` or `z to a`).

Use > `Sort > Remove sort` to remove the sort on the second of three columns. Notice how that changes the order.

Key Points

- OpenRefine provides a way to sort and filter data without affecting the raw data.

Examining Numbers in OpenRefine

Overview

Teaching: 10 min
Exercises: 10 min
Questions

- How can we convert a column from one data type to another?
- How can we visualize relationships among columns?

Objectives

- Transform a text column into a number column.
- Identify and modify non-numeric values in a column using facets.
- Use scatterplot facet to examine relationships among columns.

Lesson

Numbers

When a table is imported into OpenRefine, all columns are treated as having text values. We saw earlier how we can sort column values as numbers, but this does not change the cells in a column from text to numbers. Rather, this interprets the values as numbers for the purposes of sorting but keeps the underlying data type as is. We can, however, transform columns to other data types (e.g. number or date) using the `Edit cells > Common transforms` feature. Here we will experiment changing columns to numbers and see what additional capabilities that grants us.

Be sure to remove any `Text filter` facets you have enabled from the left panel so that we can examine our whole dataset. You can remove an existing facet by clicking the `x` in the upper left of that facet window.

To transform cells in the `recordID` column to numbers, click the down arrow for that column, then `Edit cells > Common transforms… > To number`. You will notice the `recordID` values change from left-justified to right-justified, and black to green color.

Exercise

Transform three more columns, including `period`, from text to numbers. Can all columns be transformed to numbers?

Solution

Only observations that include only numerals (0-9) can be transformed to numbers. If you apply a number transformation to a column that doesn't meet this criteria, and then click the `Undo / Redo` tab, you will see a step that starts with `Text transform on 0 cells`. This means that the data in that column was not transformed.

Numeric facet

Sometimes there are non-number values or blanks in a column which may represent errors in data entry and we want to find them. We can do that with a `Numeric facet`.

Exercise

1. For a column you transformed to numbers, edit one or two cells, replacing the numbers with text (such as `abc`) or blank (no number or text).
2. Use the pulldown menu to apply a numeric facet to the column you edited. The facet will appear in the left panel.
3. Notice that there are several checkboxes in this facet: `Numeric`, `Non-numeric`, `Blank`, and `Error`. Below these are counts of the number of cells in each category. You should see checks for `Non-numeric` and `Blank` if you changed some values.
4. Experiment with checking or unchecking these boxes to select subsets of your data.

When done examining the numeric data, remove this facet by clicking the `x` in the upper left corner of its panel. Note that this does not undo the edits you made to the cells in this column. If you want to reverse these edits, use the `Undo / Redo` function.

Scatterplot facet

Now that we have multiple columns representing numbers, we can see how they relate to one another using the scatterplot facet. Select a numeric column, for example `recordID` , and use the pulldown menu to > `Facet > Scatterplot facet` . A new window called `Scatterplot Matrix` will appear. There are squares for each pair of numeric columns organized in an upper right triangle. Each square has little dots for the cell values from each row.

✎ Exercise

1. Examine the scatterplots overall. Do the patterns make sense?
2. Why does the scatterplot for `recordID` vs `period` have the pattern it does?

Examine pair of columns in detail

We can examine one pair of columns by clicking on its square in the `Scatterplot Matrix` A new facet with only that pair will appear in the left margin.

✎ Exercise

Click in the scatterplot facet in the left margin and drag to highlight a rectangle. This will subset the data to those entries.

✎ Exercise

- Click on the `Scatterplot Matrix` square for `recordID` and `period` to get that as a facet in the left margin.
- Redo the `Text filter` on `scientificName` to show only entries including the letters `bai` . Notice the change in the scatterplot. It might be easier to see if you click `export plot` to put it on a new browser tab.

❗ Key Points

- OpenRefine also provides ways to get overviews of numerical data.

Scripts from OpenRefine

Overview

Teaching: 10 min
Exercises: 5 min
Questions

- How can we document the data-cleaning steps we've applied to our data?
- How can we apply these steps to additional data sets?

Objectives

- Describe how OpenRefine generates JSON code.
- Demonstrate ability to export JSON code from OpenRefine.
- Save JSON code from an analysis.
- Apply saved JSON code to an analysis.

Lesson

Scripts

As you conduct your data cleaning and preliminary analysis, OpenRefine saves every change you make to the dataset. These changes are saved in a format known as JSON (JavaScript Object Notation). You can export this JSON script and apply it to other data files. If you had 20 files to clean, and they all had the same type of errors (e.g. species name misspellings, leading white spaces), and all files had the same column names, you could save the JSON script, open a new file to clean in OpenRefine, paste in the script and run it. This gives you a quick way to clean all of your related data.

1. In the `Undo / Redo` section, click `Extract...`, and select the steps that you want to apply to other datasets by clicking the check boxes.
2. Copy the code from the right hand panel and paste it into a text editor (like NotePad on Windows or TextEdit on Mac). Make sure it saves as a plain text file. In TextEdit, do this by selecting `Format > Make plain text` and save the file as a `txt` file.

Let's practice running these steps on a new dataset. We'll test this on an uncleaned version of the dataset we've been working with.

1. Download an uncleaned version of the dataset: https://ndownloader.figshare.com/files/7823341 (https://ndownloader.figshare.com/files/7823341) or use the version of the raw dataset you saved to your computer.
2. Start a new project in OpenRefine with this file and name it something different from your existing project.
3. Click the `Undo / Redo` tab > `Apply` and paste in the contents of `txt` file with the JSON code.
4. Click `Perform operations`. The dataset should now be the same as your other cleaned dataset.

For convenience, we used the same dataset. In reality you could use this process to clean related datasets. For example, data that you had collected over different fieldwork periods or data that was collected by different researchers (provided everyone uses the same column headings).

Reproducible science

Now, that you know how scripts work, you may wonder how to use them in your own scientific research. For inspiration, you can read more about the succesful application of the reproducible science principles in archaeology or marine ecology:

1. Marwick et al. (2017) Computational Reproducibility in Archaeological Research: Basic Principles and a Case Study of Their Implementation (https://link.springer.com/article/10.1007/s10816-015-9272-9)
2. Stewart Lowndes et al. (2017) Our path to better science in less time using open data science tools (https://www.nature.com/articles/s41559-017-0160)

❗ Key Points

- All changes are being tracked in OpenRefine, and this information can be used for scripts for future analyses or reproducing an analysis.
- Scripts can (and should) be published together with the dataset as part of the digital appendix of the research output.

Exporting and Saving Data from OpenRefine

❓ Overview

Teaching: 10 min
Exercises: 5 min
Questions

- How can we save and export our cleaned data from OpenRefine?

Objectives

- Save an OpenRefine project.
- Export cleaned data from an OpenRefine project.

Lesson

Saving and Exporting a Project

In OpenRefine you can save or export the project. This means you're saving the data and all the information about the cleaning and data transformation steps you've done. Once you've saved a project, you can open it up again and be just where you stopped before.

Saving

By default OpenRefine is saving your project. If you close OpenRefine and open it up again, you'll see a list of your projects. You can click on any one of them to open it up again.

Exporting

You can also export a project. This is helpful, for instance, if you wanted to send your raw data and cleaning steps to a collaborator, or share this information as a supplement to a publication.

1. Click the `Export` button in the top right and select `Export project` .

2. A `tar.gz` file will download to your default `Download` directory. The `tar.gz` extension tells you that this is a compressed file. Which means that this file contains multiple files. You can double-click on the `tar.gz` file and it will expand into a directory. A folder icon will now appear.
    - On Windows, opening `tar.gz` files requires additional software such as 7-zip (http://www.7-zip.org/) or WinZip (http://www.winzip.com/). Download and run the installer of your choice.
    - Double-click the exported `tar.gz` file. If Windows asks how you want to open the file, check the ‘Always use this app to open `.gz` files’ box, then select “More apps”.
    - If your chosen application is not listed, select ‘Look for another app on this PC’.
    - In the file browser, navigate to `C:\Program Files`, find the application you installed, and double-click on its executable (`7zFM`, for example).
3. Look at the files that appear in this folder. What files are here? What information do you think these files contain?

Solution

You should see:

- a `history` folder which contains three `zip` files. Each of these files itself contains a `change.txt` file. These `change.txt` files are the records of each individual transformation that you did to your data.
- a `data.zip` file. When expanded, this `zip` file includes a file called `data.txt` which is a copy of your raw data. You may also see other files.

You can import an existing project into OpenRefine by clicking `Open...` in the upper right > `Import Project` and selecting the `tar.gz` project file. This project will include all of the raw data and cleaning steps that were part of the original project.

Exporting Cleaned Data

You can also export just your cleaned data, rather than the entire project.

1. Click `Export` in the top right and select the file type you want to export the data in. `Tab-separated values` (`tsv`) or `Comma-separated values` (`csv`) would be good choices.
2. That file will be exported to your default `Download` directory. That file can then be opened in a spreadsheet program or imported into programs like R or Python, which we’ll be discussing later in our workshop.

Remember from our lesson on Spreadsheets that using widely-supported, non-proprietary file formats like `tsv` or `csv` improves the ability of yourself and others to use your data.

Key Points

- Cleaned data or entire projects can be exported from OpenRefine.
- Projects can be shared with collaborators, enabling them to see, reproduce and check all data cleaning steps you performed.

Other Resources in OpenRefine

Overview

Teaching: 5 min
Exercises: 5 min
Questions

- What other resources are available for working with OpenRefine?

Objectives

- Explore the online resources available for more information on OpenRefine.
- Identify other resources about OpenRefine.

Lesson

Identify other Resources about OpenRefine.

OpenRefine is more than a simple data cleaning tool. People are using it for all sorts of activities. Here are some other resources that might prove useful.

OpenRefine has its own web site with documentation and a book:

- OpenRefine web site (http://openrefine.org/)
- OpenRefine Documentation for Users (Wiki site) (https://github.com/OpenRefine/OpenRefine/wiki/Documentation-For-Users)
- Using OpenRefine (http://www.worldcat.org/title/using-openrefine-the-essential-openrefine-guide-that-takes-you-from-data-analysis-and-error-fixing-to-linking-your-dataset-to-the-web/oclc/889271264) book by Ruben Verborgh, Max De Wilde and Aniket Sawant
- OpenRefine history from Wikipedia (https://en.wikipedia.org/wiki/OpenRefine)

In addition, see these other useful resources:

- Grateful Data (https://github.com/scottythered/gratefuldata/wiki) is a fun site with many resources devoted to OpenRefine, including a nice tutorial.
- Margaret Heller (http://www.gloriousgeneralist.com/) shows how she uses OpenRefine for Measuring and Counting Impact in Repositories (http://www.gloriousgeneralist.com/2014/12/notes-on-measuring-and-calculating-impact-in-institutional-repositories/).
- Intersect Course Resources (https://github.com/IntersectAustralia/TrainingMaterials/tree/master/or) has Jared Berghold’s **Cleaning & Exploring your data with Open Refine**.

There are more advanced uses of OpenRefine, such as bringing in column or cell data using web locators (URLs or APIs). The links above can give you a start on your journey.

Exercise

Visit one of these sites and share what you find with another person.

Key Points

- Other examples and resources online are good for learning more about OpenRefine

Edit on GitHub (https://github.com/datacarpentry/OpenRefine-ecology-lesson/edit/gh-pages/aio.md) / Contributing (https://github.com/datacarpentry/OpenRefine-ecology-lesson/blob/gh-pages/CONTRIBUTING.md) / Source (https://github.com/datacarpentry/OpenRefine-ecology-lesson/) / Cite (https://github.com/datacarpentry/OpenRefine-ecology-lesson/blob/gh-pages/CITATION) / Contact (mailto:team@carpentries.org)

Using The Carpentries style (https://github.com/carpentries/styles/) version 9.5.2 (https://github.com/carpentries/styles/releases/tag/v9.5.2).