2025-11-04-strathclyde OpenRefine lesson instructor notes

Summary and Setup

[SLIDE HERE: Links/QR codes]

- Lesson site: https://swcarpentry.github.io/shell-novice/

Data cleaning

- Before you can analyse your data, you normally need to *clean it**
- What is "data cleaning"?
  - Identification (and maybe correction) of errors in the data
  - Formatting data for consistency
- **If your data is not clean, your analysis results may be false and/or not reproducible**
- OpenRefine is a powerful, free, and open source tool for working with messy data
  - It can clean data and transform it from one format to another
  - It can also generate reproducible data cleaning workflows or pipelines that track the changes you make
  - The scripts can also be applied to **undo** changes that were made to the raw data
- Data cleaning steps often need to be applied to more than one file
  - The pipelines OpenRefine makes allow you to reapply the processes to new datasets

Introduction

- OpenRefine is a Java program that runs on your own machine
  - It is a desktop application that uses your web browser as an interface
  - No internet connection is required and no data or commands are sent to any server: **it can be used on confidential data**
- OpenRefine **does not modify your original dataset** - this is best practice
  - You must save the cleaned data in a different file
- All OpenRefine actions can be reversed
- OpenRefine saves as you go - you can leave or pick up a data cleaning proejct at any point
- OpenRefine can import your data in many formats, including
  - Comma-separated (CSV) or tab-separated (TSV)
  - text files

  - fixed-width columns
  - JSON
  - XML
  - OpenDocument Spreadsheet (ODS)
  - Excel spreadsheet (XLS/XLSX)
  - RDF data

Importing data

[OPENREFINE DEMO]

- Let's get started with OpenRefine, by creating a project
- Start OpenRefine
  - This will open in a web browser at http://127.0.0.1:3333
  - In the left margin you'll see options: `Create Project`, `Open Project` and so on
- Click `Create Project`
  - The window "Create a project by importing data" should be visible
- Select `This Computer` as you want to use a local file
- Click `Browse`
  - Navigate to where you downloaded `Portal_rodents_19772002_simplified.csv`
  - Select that file and click `Open`
  - The filename should be visible
- Click `Next` to upload the data to OpenRefine
  - Once it's loaded, OpenRefine will present a preview of your data
- **Any obvious errors (like file formatting problems) should be visible at this point**
- In the middle of the page there should be a set of options including `Character Encoding`
  - Ensure that the `Trim leading and trailing whitespace from strings` option is **not** checked
- Everything should look OK
- Click `Create project`
- OpenRefine should present you with a table of 790 rows
  - All data is imported as text - **even numerical data**

Exploring data

Exploring data with facets

- **Facets** are a useful concept in OpenRefine

  - **Faceting** is the process of exploring data using filters, to explore its composition
  - Faceting allows us to identify a subset of data to change in bulk

- **Faceting groups all like values that appear in a column**
  - It allows filtering by those values
- This is easier to understand through practice
- Find the `scientificName` column in the dataset
- Click the down arrow/triangle next to the header
- Choose `Facet -> Text facet`
- In the left panel you will see a box representing each unique value in the `scientificName` column
  - The number of times each entry appears is also shown
- Sort the facet by `name` and `count`
  - What problems do you see with the data?
- Hover your mouse over a name in the facet
  - You should see an `edit` and an `include` link
  - If you click on `edit` you will see options to repair errors immediately - **don't do this yet**

[SHOW SLIDES: Challenge (faceting)]

[OPENREFINE DEMO]

- By default, all imported columns are treated as text
- We can convert data we know to be a different data type (e.g. number or date) using `Edit cells -> Common transforms`
  - **Sometimes the data has errors that prevent us treating it as numeric** so we explore it with a facet first
- Click on the down arrow/triangle for `yr`
- Select `Facet -> Numeric facet`
  - This will generate a facet that says there is no numeric data present (the data is in text format)
- Click on the down arrow/triangle for `yr`
- Select `Edit cells -> Common transforms... -> To number`
  - The data shifts to right-justified and turns green
  - The facet shows a histogram
- Transform `period`, `plot`, and `recordID` from text to numbers

- Remove the facets by clicking the `X`

Using scatterplots

- We can see how numeric columns relate to each other using the **scatterplot facet**
- Click on the down arrow/triangle for `recordID`
  - Select `Facet -> Scatterplot facet`
  - A new window called `Scatterplot matrix` appears
  - This plots each numeric column against the others
- Select the plot of `recordID` against `yr`
  - A facet of the comparison appears
- Click in the scatterplot and drag to highlight a rectangle
  - **What happens to the data**? It is selected/subsetted in the table
  - **We can use this approach to identify outliers in a dataset**
- Click anywhere in the scatterplot to deselect

Transforming data

Data splitting

- We can split data from one column into multiple columns, if we can identify a common separator (like a comma or space)

- Let's split the `scientificName` into two columns: one for genus, one for species

- Click on the down arrow/triangle for `scientificName`

  - Select `Edit Column -> Split into several columns...`
  - In the `Separator box` replace the comma with a space
  - **Uncheck the `Remove this column` box**
  - Click `OK`

- You should now see several columns: `scientificName1`, `scientificName2`, etc.

  - We can see that some of the data was not well-formatted, in the sense it wasn't simple *Genus species* data
  - We're going to have to fix this

Undoing/redoing actions

- But first let's undo what we just did with the split

  - This is quite a common requirement when cleaning data

- Click `Undo/Redo` in the left margin

  - You should see all the changes you made to the data in that panel
  - The current stage is highlighted
  - We can click on any stage in the processing to jump to that version of the data

- Click on the appropriate step just before splitting `scientificName`

- We could click on the last stage and redo the change, but let's not do that.

Trimming leading/trailing whitespace

- It's quite common for whitespace to be added at the start/end of text in datasets
    - But it's difficult to spot this by eye
- OpenRefine can remove these characters for us automatically
- Click on the down arrow/triangle for `scientificName`
    - Select `Edit cells -> Common transforms -> Trim leading and trailing whitespace`
    - OpenRefine will warn us about deleting project history (taking a different path), but click `Apply anyway`
- Let's split the `scientificName` into columns again: one for genus, one for species
- Click on the down arrow/triangle for `scientificName`
    - Select `Edit Column -> Split into several columns...`
    - In the `Separator box` replace the comma with a space
    - **Uncheck the `Remove this column` box**
    - Click `OK`
- You should now see two columns: `scientificName1`, `scientificName2`

Renaming columns

- We have Genus and species separated into two columns, but they have generic names
    - We want to call the columns `genus` and `species`
- Click on the down arrow/triangle for `scientificName1`
    - Select `Edit Column -> Rename column...`
    - Enter `genus` into the field
- Click on the down arrow/triangle for `scientificName2`
    - Select `Edit Column -> Rename column...`
    - Enter `species` into the field
    - **You will not be allowed to rename the column**
    - **Change the name of the `species` column to `species_abbreviation`**
    - Rename the `scientificName2` column to `species`

Combine columns

- The date for each row is split in three columns, for month, day, and year
    - **We want to create a single column which contains the date**

- Click on the menu for the `yr` column
  - Select `Edit column -> Join columns...`
  - Check the boxes next to `mo` and `dy`
  - Enter `-` as a separator
  - Select the option `Write result in new column named` and write `date` in the field
  - Click `OK`
- Click on the menu for the new `date` column
  - Select `Edit cells -> Common transforms -> To date`
- Now we can use a **Timeline facet** to inspect the data
- Click on the menu for the new `date` column
  - Select `Facet -> Timeline facet`
  - A histogram view of sampled dates should appear in the left margin

Data clustering

- Clustering allows us to find groups of entries that are not identical, but that are similar
- These may be alternative representations of the same thing - possibly even typos
- Make a text facet for `scientificName`
- In `scientificName` there are several near-identical entries
  - For example a misspelling of *Ammospermophilus harrisii*
  - Click the `Cluster` button
    - Try some options for `Method` and `Keying function` and click the `Cluster` button
- Choose `Key collision` and `Metaphone3`
  - This identifies a cluster of options for *Ammospermophilus harrisii*
  - It also suggests a new cell value to replace the likely misspellings
- Tick the `Merge` checkbox
- Click `Merge selected and close`
- Note that the facet has updated so that there are no more misspellings of *Ammospermophilus harrisii*

Filtering and sorting

- Sometimes you only want to view and work with a subset of your data, or to apply an operation only to that subset
- In OpenRefine, we do this with **filters**

Include/Exclude in facets

- The `include` and `exclude` buttons in facets allow us to control our subset for working with

- In the text face for `scientificName`, hover over *Baiomys taylori*
  - The `edit` and `include` options will appear
  - Click on `include`
- The *Baiomys taylori* text changes to orange to show it is included, and the `exclude` option appears
  - The number of matching rows has also changed (to 33 from 790)
- Hover over *Chaetodipus baileyi* and click `include`
  - This text also turns orange, and the number of matching rows increases to 35
- Click on `reset` at the top of the facet
  - This returns the selection to the original state
  - All the entries in the facet return to blue text

Text filters

- We can also create text filters on columns
- Close all the facets that are currently open
- Click on the down arrow/triangle for `scientificName`
  - Click on `Text filter`
  - A new text filter opens in the left margin
- Type `bai` into the filter field
  - The top of the page shows the 35 rows where the `scientificName` column contains the string `bai`
- Change the number of rows to be shown to `50`
  - Scroll down to see the matching rows
- Click on the down arrow/triangle for `scientificName`
  - Click on `Facet -> Text facet`
  - We can see that two species have been selected

Sorting data

- Sorting data is a good way to see the range spanned by your data, and to spot outliers at the extremes
- The `Sort...` tool is available in the drop-down menu for each column
- Click on the down arrow/triangle for `mo` (month)
  - Click on `Sort...`
  - Select `numbers` and make sure `smallest first` is checked so that months are sorted in order
  - Click `OK`
- Click on the down arrow/triangle for `plot`

  - Click on `Sort...`
  - Select `numbers` and make sure `smallest first` is checked
  - Check `sort by this column alone` to see which samples came from plot 1 and clear the sorting on `mo`

- We can also sort by multiple columns

- Sorting takes place in the order you apply `sort` to each column

- Click on the down arrow/triangle for `yr` (year)

  - Click on `Sort...`
  - Select `numbers` and make sure `smallest first` is checked so that years are sorted in order
  -
    - Check `sort by this column alone` to see which samples came from plot 1 and clear the sorting on `plot`
  - Click `OK`

- Click on the down arrow/triangle for `mo` (month)

  - Click on `Sort...`
  - Select `numbers` and make sure `smallest first` is checked so that months are sorted in order
  - Click `OK`

- Click on the down arrow/triangle for `day` (day)

  - Click on `Sort...`
  - Select `numbers` and make sure `smallest first` is checked so that days are sorted in order
  - Click `OK`

- The data should now be in chronological order

- Click on `reset` in the `Facet / Filter` panel

Exporting, importing, and applying the data cleaning pipeline/workflow

- OpenRefine saves every change you make during your data cleaning and preliminary analysis

  - The data is kept in a file format called JSON - an open standard - that is shareable with other applications and which can be applied to other data files
  - If you find yourself having to clean multiple data files of the same kind, you can use this JSON script to clean them quickly in OpenRefine

- Click on the `Undo / Redo` tab

  - Click on the `Extract` button
  - This shows the entire history of operations you have performed on your data, in order
  - On the left you can choose to include some, none, or all of these steps in the exported file
  - The operations that will be exported are shown on the right hand side of the window (this is JSON format)

- Click on the `Export` button

    - Save the file as `history.json` in a convenient location
- Click the `Close` button
- We can use this JSON file to clean a new dataset
    - Let's practice in a new project
- Click on the `Open` button at the top of the window
    - This opens a new project in the browser
- Click `Browse`
    - Navigate to where you downloaded `Portal_rodents_19772002_simplified.csv`
    - Select that file and click `Open`
    - The filename should be visible
- Click `Next` to upload the data to OpenRefine
    - Once it's loaded, OpenRefine will present a preview of your data
- In the middle of the page there should be a set of options including `Character Encoding`
    - Ensure that the `Trim leading and trailing whitespace from strings` option is **not** checked
- Everything should look OK
- Click `Create project`
    - You should recognise the uncleaned data from earlier
- Click the `Undo / Redo` tab
    - Click the `Apply` button
    - Click `Browse` and select the `history.json` file you just saved - the JSON script will appear in the window
    - Click the `Run Operations` button
- You should see a newly-cleaned dataset
- Using standardised data cleaning approaches is an essential component of reproducible science
    - These kinds of data-cleaning scripts should be provided alongside your published work and data (e.g. as Supplementary Information)

Exporting and saving cleaned data

- When you have cleaned your data, you will want to export it to a new file
- **It is important to keep raw and cleaned data in separate files**
- Click the `Export` button at the top right of the window

    - Click on `Tab-separated value`
    - In the file dialogue box, change the filename to something that indicates the data have been cleaned (e.g. `Portal-rodents-19772002-simplified-csv-cleaned.tsv`)
    - Click on `Save`

- The file can then be imported into downstream analysis tools

Saving/Exporting a project

- In addition to saving the data cleaning pipeline and the cleaned data separately, OpenRefine can save the entire project

- OpenRefine saves your projects by default

    - When you start OpenRefine you will be presented with a list of former projects
    - You can click on items in this list to open them

- You can also export the project, so it can be shared with others

- Click the `Export` button

    - Click `OpenRefine project archive to file`
    - In the file dialogue box, navigate to a convenient location, and click `Save`

- This saves a `.tar.gz` (compressed) file that can be reused by OpenRefine

Importing a project

- To import an OpenRefine project, click on the `Open` button
    - Click on `Import project` in the left menu
    - Click on `Browse` and navigate to the location of the project file
    - Select it and click `Open`
    - Click `Import project`
- You will be presented with a view onto the project, with the history of actions intact

Summary

- That's all I have to say about OpenRefine

    - You can find more resources listed at https://datacarpentry.github.io/OpenRefine-ecology-lesson/10-resources.html

- In this session you've learned how to use OpenRefine to reproducibly clean and inspect your data

    - You've created, imported, and exported projects
    - You've imported raw data and exported cleaned data
    - You've created, imported, and exported custom data-cleaning workflows
    - You've used facets to edit and clean data
    - You've used facets and filters to subset your data for analysis
    - You've used OpenRefine to transform your data into more useful forms (like combining or splitting columns, and clustering)

- This is a very useful skillset to have for professional, reproducible science and data analysis