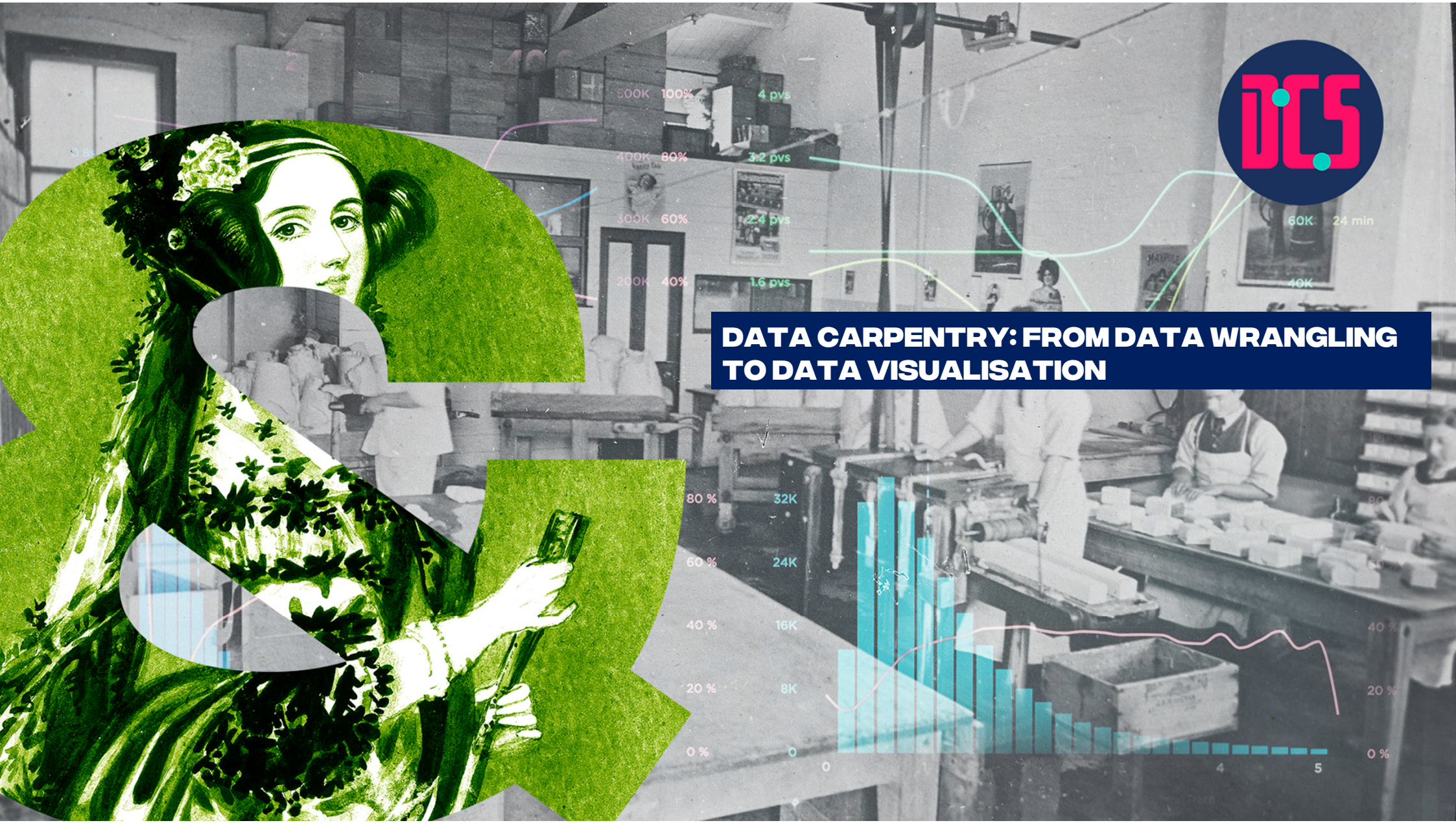

DATA CARPENTRY: FROM DATA WRANGLING TO DATA VISUALISATION

Plan of Today

1. Introduction
2. Working with OpenRefine
3. Filtering and Sorting with OpenRefine
4. Examining Numbers in OpenRefine
5. Scripts from OpenRefine
6. Exporting and Saving Data from OpenRefine

What is it?

- OpenRefine is a **standalone open source** desktop application for **data clean up** and **transformation** to other formats, the activity known as **data wrangling**.
- It is a **Java** program hence you operate it **through your browser** but you don't need to be online.
- Works with large-ish datasets (100,000 rows). Does not scale to many millions. (yet).
- More information on the software in here.

Why Using Open Refine?

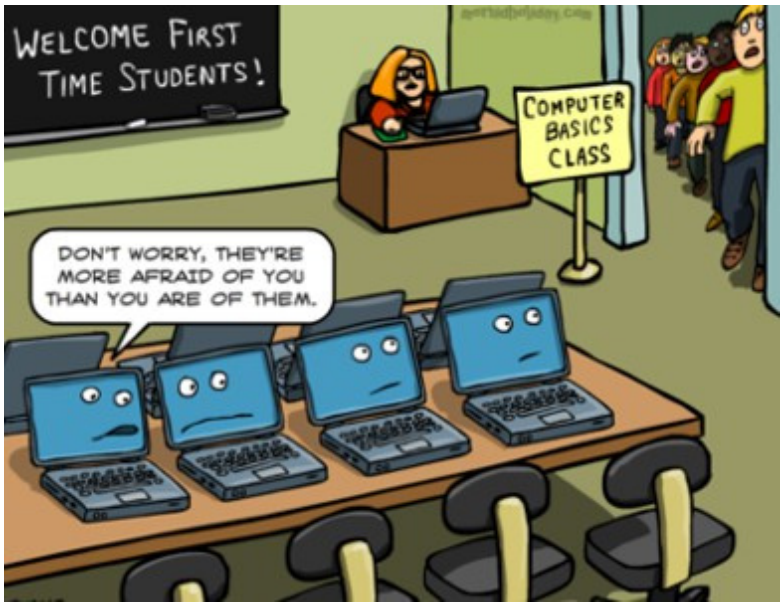
- Helps you **have a good overview** of your data
- All actions are **tracked and easily reversible**
- You ***must* save your work to a new file**
- Incredibly useful for **cleaning messy data.**
- **Large and helpful community online**. If you need help http://openrefine.org
- It is important to **know what you did to your data**. With OpenRefine, you can **capture all actions** applied to your raw data and share them with your publication as supplemental material if needed.

Sticker time!!!

Do you all have Open Refine download and working and the .csv file to work on?

Objectives

- Create a new OpenRefine project from a CSV file.
- Understand potential problems with file headers.
- Use facets to summarize data from a column.
- Use clustering to detect possible typing errors.
- Understand that there are different clustering algorithms
- Employ drop-downs to remove white spaces from cells.
- Manipulate data using previous steps with undo/redo.



Creating a Project
(Start looking at your data)

More information
in here

Main supported files:

TSV
CSV
*SV
Excel (XLS, XLSX)
JSON
XML
RDF as XML

Data Faceting

(Exploring data by applying multiple filters)

- **Seeing the big picture** of your data

- **Filtering down** to just the subset of rows that you want to change in bulk.

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.

Exercise: 1

1. Using faceting, find out **how many different interview_date values** there are in the survey results.

2. Is the column formatted as Number, Date, or Text? How does changing the format change the faceting display?

3. Use **faceting to produce a timeline display** for interview_date. You will need to use Edit cells > Common transforms > To date to **convert this column to dates**

4. During what period were **most of the interviews collected**?

More on Faceting

OpenRefine Wiki: Faceting

As well as 'Text facets' Refine 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

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.

Clustering

More on clustering

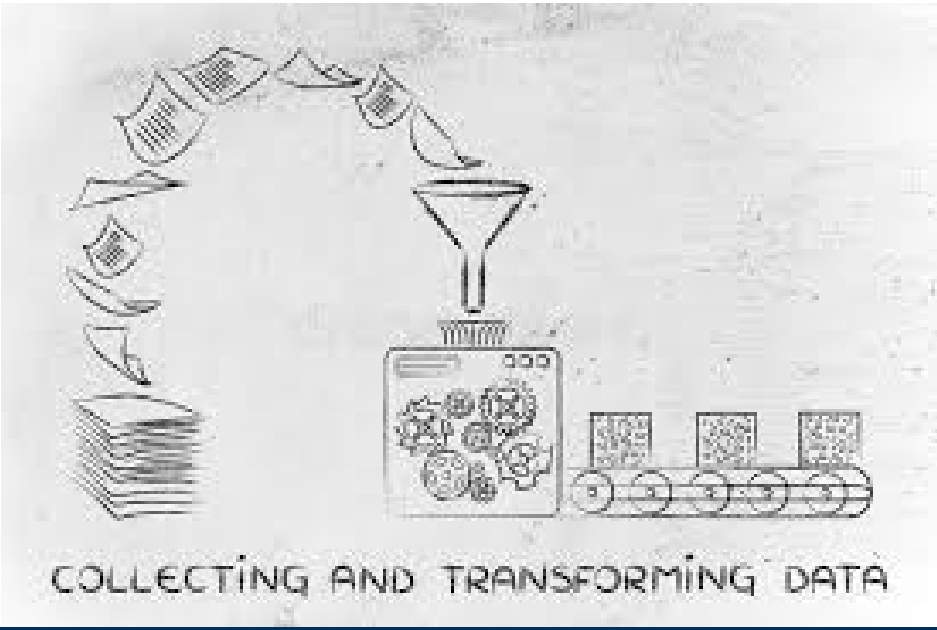
(Finding groups of different values that might be the same thing)

- Edit and control typos and different style of representing the same concept (e.g. Ruaca, Ruca, Ruca-Nhamuenda…)

Transforming data

he data in the **items_owned** column is a set of items in a **list**. The list is in **square brackets** and each item is in single quotes. Before we split the list into individual items in the next section, we first want to remove the brackets and the quotes.

1.First we will remove all of the left square brackets ([). In the Expression box type value.replace("[", "") and click OK.
2.What the expression means is this: Take the value in each cell in the selected column and replace all of the “[” with “” (i.e. nothing - delete).
3.Click OK. You should see in the items_owned column that there are no longer any left square brackets.



- Use this same strategy to remove the single quote marks ('), the right square brackets (]), and spaces from the items_owned column.

1.value.replace("'", "")
2.value.replace("]", "")
3.value.replace(" ", "")
You should now have a list of items separated by semi-colons (;).

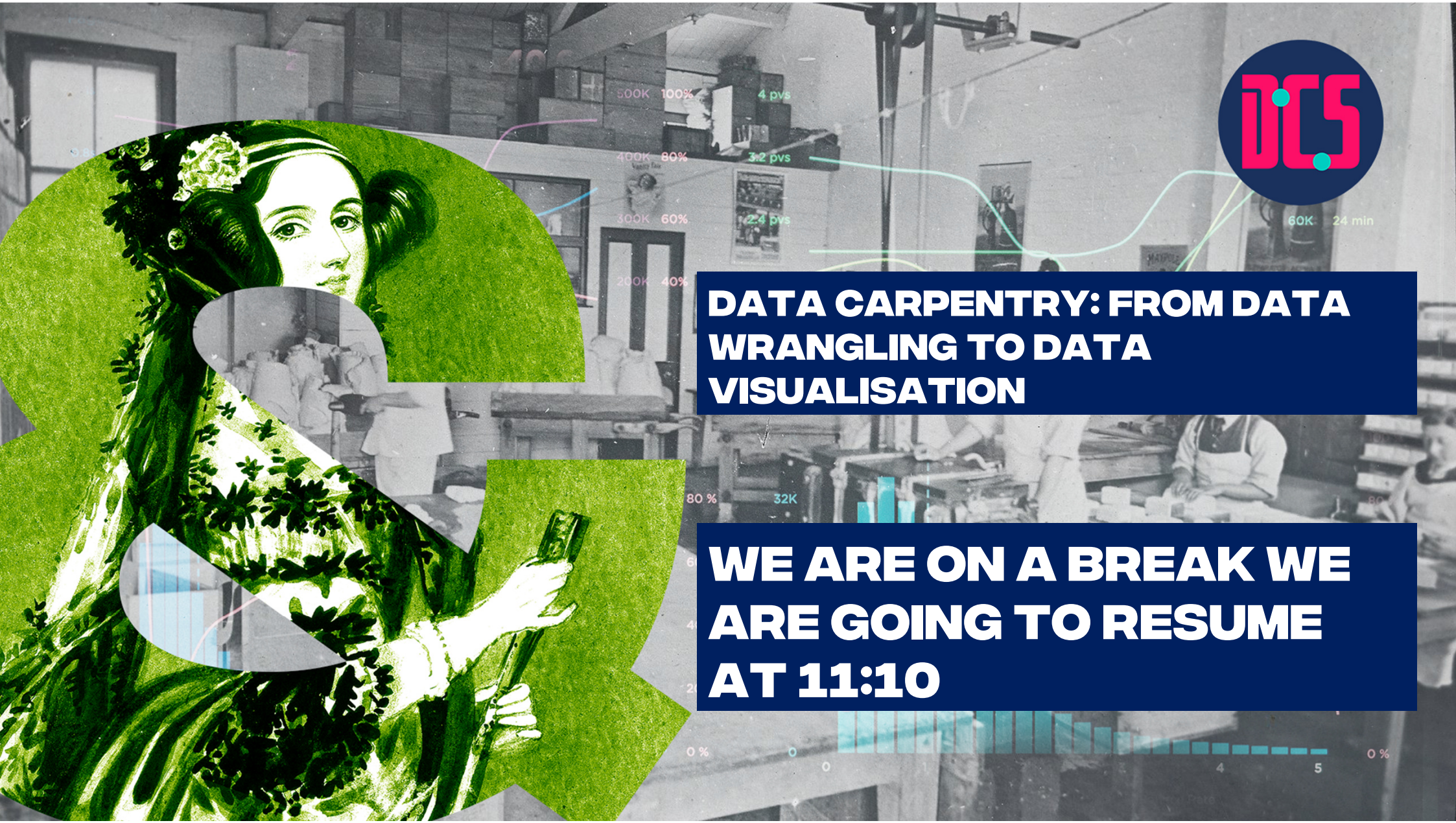
Now that we have cleaned out extraneous characters from our items_owned column, we can use a text facet to see which items were commonly owned or rarely owned by the interview respondents.
1.Click the down arrow at the top of the items_owned column. Choose Facet > Custom text facet...
2.In the Expression box, type value.split(";").
3.Click OK.

Exercise: 2

1. Perform the same clean up steps and customized text faceting for the **months_lack_food** column. Which month(s) were farmers more likely to lack food? **Hint:** To reuse a GREL command, click the History tab and then click Reuse next to the command you would like to apply to that column.

Undo / Redo
(Control your steps)

Trim Leading and Trailing Whitespace
(Remove blank characters from the beginning and end)


DATA CARPENTRY: FROM DATA WRANGLING TO DATA VISUALISATION
WE ARE ON A BREAK WE ARE GOING TO RESUME AT 11:10

Objectives

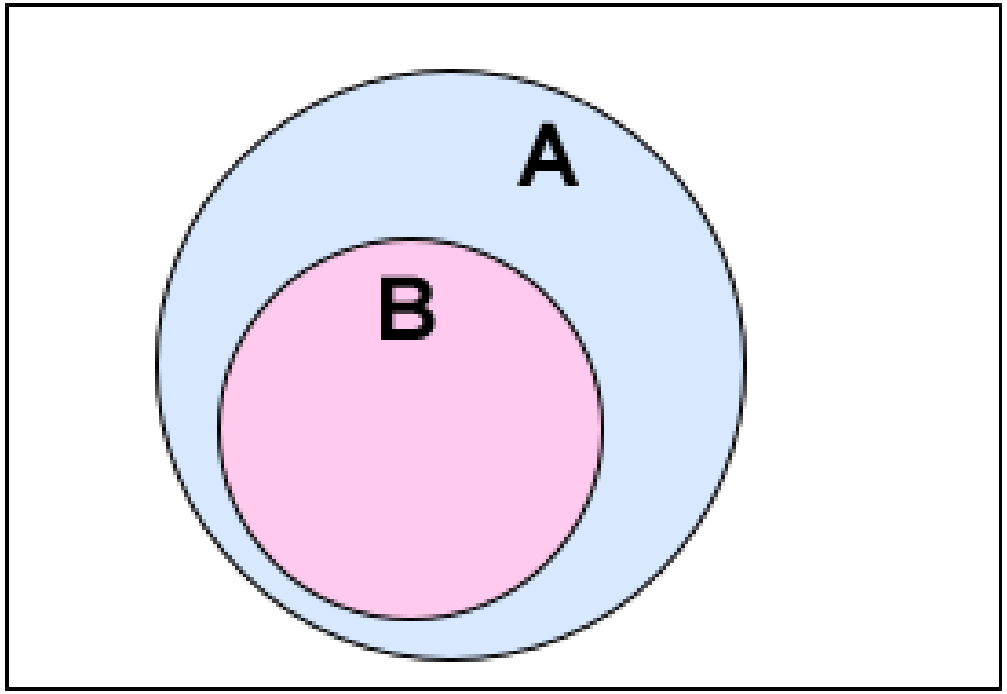
- Filter to a subset of rows by text filter or include/exclude.
- Sort table by a column.
- Sort by multiple columns.

Tidying Up your Data!

Filtering

(Working on a subset)

Using Filter and include/exclude



Sorting Data

(Single column and multiple column)

Exercise 3

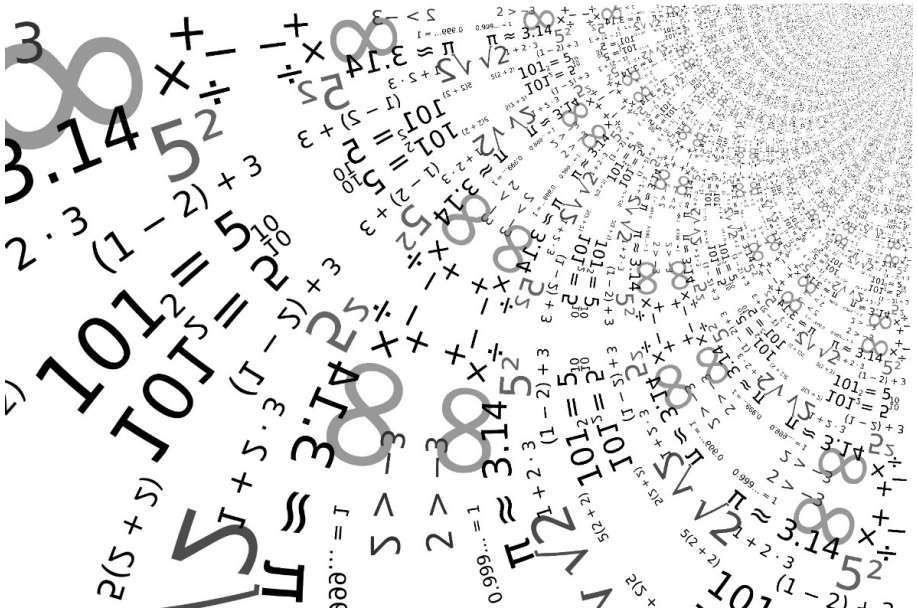
- Sort the data by gps_Altitude. Do you think the first few entries may have incorrect altitudes?

The first few values are all 0. The altitudes are more likely 'missing' than incorrect. The survey is delivered by Smartphone with the gps information added automatically by the app. The lack of an altitude value suggests that the smartphone was unable to provide it and it defaulted to 0.

- 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.

Objectives

- Transform a text column into a number column.
- Identify and modify non-numeric values in a column using facets.



Exercise 4a:

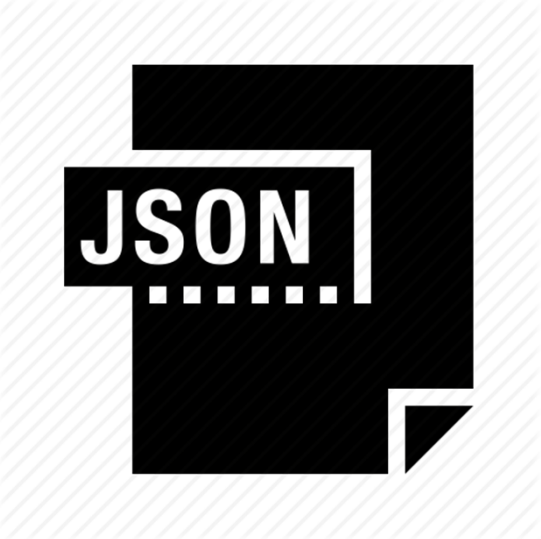
Transform 3 columns (**no_members, yrs_liv, and buildings_in_compound**) from **text data** to **number data**? What happen to the columns that are not integers?

Exercise 4b:

- 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).
- Use the pulldown menu to apply a numeric facet to the column you edited. The facet will appear in the left panel.

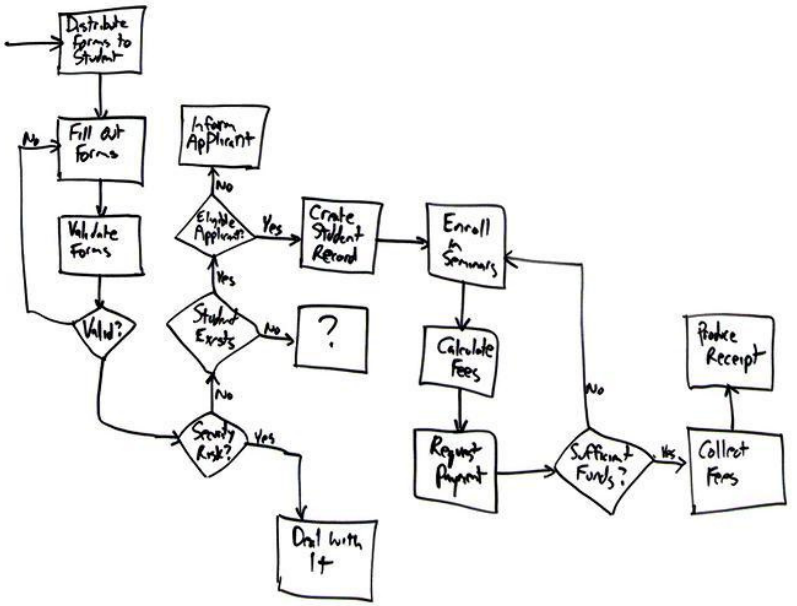
Objectives

- Describe how OpenRefine generates JSON code.
- Demonstrate ability to export JSON code from OpenRefine.
- Demonstrate ability to import a JSON code file to apply the analysis to another dataset.



Saving the cleaning flowchart

As you conduct your **data cleaning and preliminary analysis**, Open Refine saves every change you make to the dataset. T**hese 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.



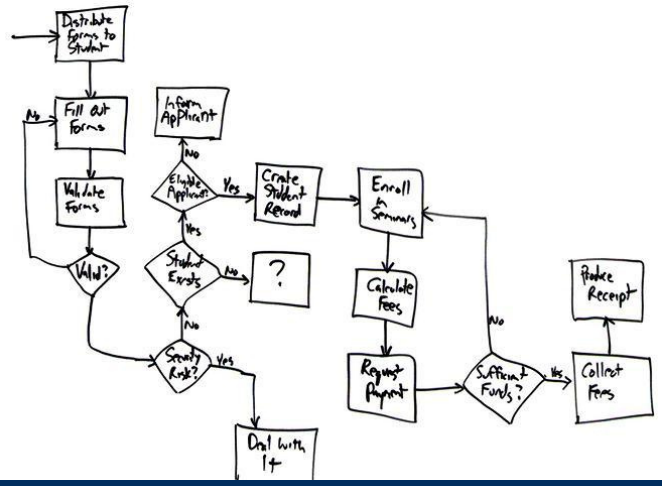
Let’s save our steps

Importing a script to use against another dataset

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.Start a new project in OpenRefine using the messy dataset you downloaded before. Give the project a new name.
2.Click the Undo / Redo tab > Apply and paste in the contents of .txt file with the JSON code.
3.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). The data in this file was generated from an eSurvey system with the actual survey being delivered centrally to a smartphone, so the column headings are pretty much guaranteed to be the same.

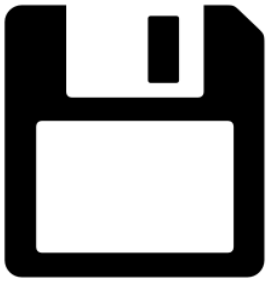


Objectives

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

Saving

By default OpenRefine is saving your project continuously. 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.

Any Question?