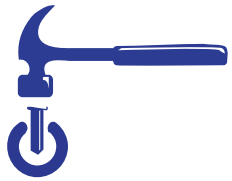
Software skills for librarians:

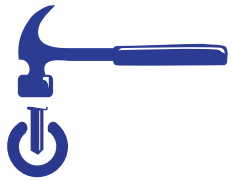
Library carpentry

Module 2: Open Refine



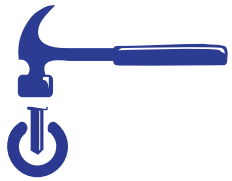
Introduction to Open Refine

- A tool for working with tabular data
- Examine your data
- Resolve inconsistencies and perform global edits
- Split data into smaller chunks
- Match local data with remote sources
- Enhance data from other sets



Common uses

- How many times does a particular value appear?
- Change dates to a common format
- Correct variants in spelling, capitalisation or punctuation
- Split addresses into component parts
- Add data from another file



Creating a project

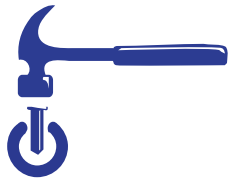
- Formats accepted include TSV, CSV, JSON, XML
- Run Open Refine in your browser

  Use http://127.0.0.1:3333/ if necessary
- Select your file and click next
- Select format options:

  Character encoding

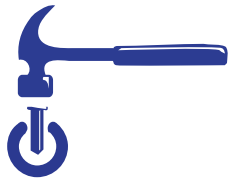
  Headings from first row

  Don’t automatically recognise numbers and dates
- Click create project



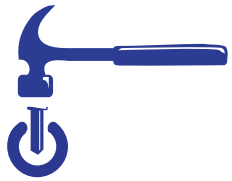
Getting started

- Open Refine presents a spreadsheet like display
  - A preview of part of your data
  - Options to change number of rows displayed
  - And move through your data using ‘previous’ and ‘next’
- Infinite ‘undo’ and ‘redo’
  - Save history of operations, and apply them to other projects
- Star and flag in left hand column
  - Select two distinct groups of rows
  - Limit display to those with star, flag, or both



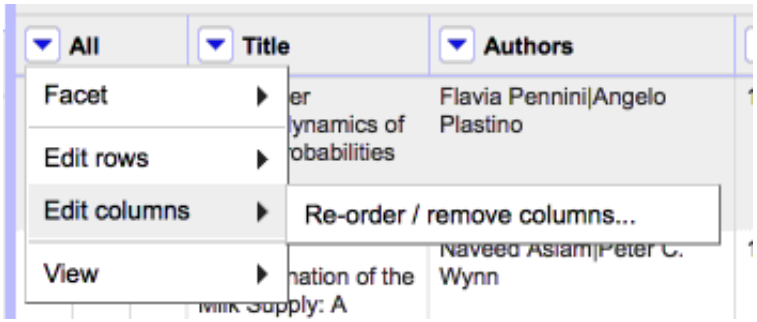
Rows and records

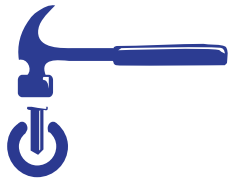
- The original data is row oriented:
  - One row per record
  - One field per column, fields are not repeatable
- Records describe a unique object
  - May have several instances of a field
  - Such as repeated name or subject heading fields
  - Grouped by identifier in first column
- Multi-valued cells can be split
- Or records merged into single rows



Simple operations

- Reordering, renaming or removing columns
- Menu at top of first column



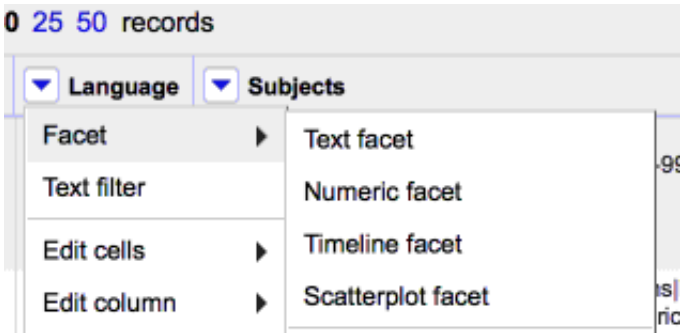


Simple operations

- Sorting data
- Menu at top of column
- Sorting is temporary

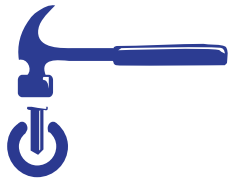
Simple operations

- Faceting
- Groups the values in a column
- Menu at top of column
- Types of facet: Text, numeric, dates, custom



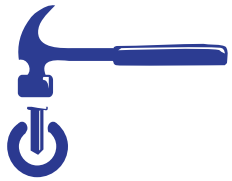
Simple operations

- Filtering
- Menu at top of column
- Records containing a word or regular expression
- Changes apply only to filtered data



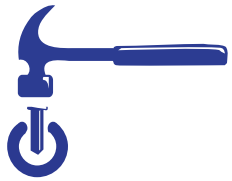
Clustering

- Groups similar values together
  - Useful with things like names which exhibit small variations
- Created algorithmically, many different methods to try
- Merges values together:
  - Using most common value
  - By clicking on one
  - Or by entering one
- Use with caution
- Accessed from menu at top of column



Transformations

- Make changes to your data including:
  - Splitting a single column into multiple columns
  - Merging columns
  - Standardising the format of data
  - Extracting data from a longer string
- Written in GREL (Google Refine Expression Language)
- Similar to formulae in a spreadsheet, but focussing on text



Common transformations

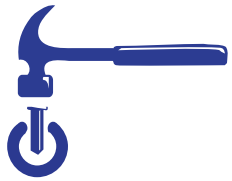
- These may be accessed from menu options:

  To uppercase: `value.toUppercase()`

  To lowercase: `value.toLowercase()`

  To title case: `value.toTitlecase()`

  Remove leading and trailing whitespace: `value.trim()`

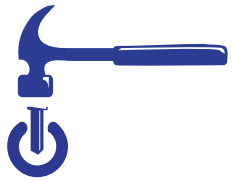


Writing transformations

- Select a column and choose edit cells → transform
- Type a GREL expression into the box
- `value` – gives the value in the current column:

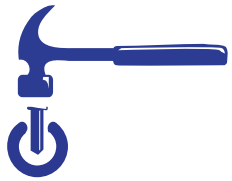
  also `cell` and `row`
- So: `value.function(param)`

  Or: `function(value,param)` apply function to value
- Preview the effect on ten rows of your data



Data types

- String, Number, Date, Boolean, Array
- Some operations only work specific data types
  - Such as formatting dates
  - Or mathematical functions
- Booleans or Arrays are not encountered directly
  - But may be the result of functions like contains and split



Arrays

- Cannot be stored in a cell

  Literals: `["Mon", "Tue", "Wed", "Thu", "Fri"]`

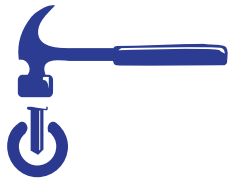
  Subscripts: `array[0]`

  Split: `value.split(",")`

  Sort: `array.sort()`

  Join: `array.join(" ")`

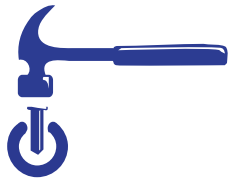
  Reverse: `value.split(",").reverse()`



Example

- Custom Facet: `value.contains(",")`
- Select 'true'
- Apply transformation: `value.match(/(.*),(.*)/)`
- Reverse array and join the two elements

```
value.match(/(.*),(.*)/).reverse().join(" ")
```



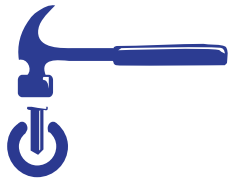
Looking up external data

- Look up additional information from a remote service

  eg. find titles given ISBN numbers

  Dates of birth given author names

- Edit column → add column by fetching URLs
- Build a query from values in your data
- Fetch results for each line
- Parse resulting data



Looking up external data

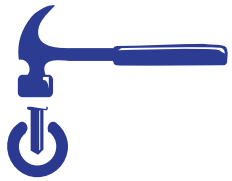
- Fetch data from another OpenRefine project
- Uses a ‘key’ to match additional data
- Create a new column to hold result of look-up:

  Edit column → add column based on this column

- Uses cross function

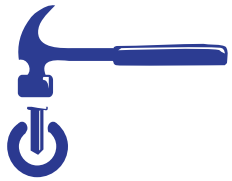
```
cell.cross(project,column)
```

- Returns entire rows: can extract relevant column with subscripts



Export data

- Changes are retained with the project
- The modifed data set can be exported
- Supported formats include CSV, TSV, HTML and Excel
- Custom export
- Export button in top right



Further reading

- Cleaning data with OpenRefine:

  http://www.programminghistorian.org/lessons/

  cleaning-data-with-openrefine

- OpenRefine documentation wiki:

  https://github.com/OpenRefine/OpenRefine/wiki

- Using OpenRefine / Ruben Verborgh and Max de Wilde.

  Packt, 2013 — ISBN 9781783289080