OBJECTIVES

- Introduction to *OpenRefine*
- Hands-on
  - OpenRefine interface
  - Column manipulation
  - Sorting
  - Filtering / Facets / Clustering
  - History (Undo/Redo)
  - Transformations / GREL
  - Data types
  - Regular Expressions
  - Export
- Helpful hints and Resources
WHAT IS OpenRefine?

▶ A tool for working with messy data [1]
  ▶ Clean
  ▶ Transform from one format into another
  ▶ Extend with web services and external data
▶ Formerly Freebase Gridworks and then Google Refine and formerly supported by Google until 10/02/2012
▶ Rebranded as OpenRefine and supported by volunteers on a GitHub open source community
▶ More history on ‘http://openrefine.org/2013/10/12/openrefine-history.html’
WHAT CAN IT HELP YOU DO?

With a simple tabular format, you can

▶ Get an overview of a data set
▶ Resolve inconsistencies in a data set formats
▶ Resolve inconsistencies in where data appears
▶ Resolve inconsistencies in terminology used in the data
▶ Split data up into more granular parts
▶ Match local data up to other data sets
▶ Enhance a data set with data from other sources
COMMON SCENARIOS

- Separate data combined together in a single

<table>
<thead>
<tr>
<th>Data you have</th>
<th>Desired data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Institution</td>
<td>Library name</td>
</tr>
<tr>
<td>University of Wales, Llyfrgell Thomas Parry Library, Llanbadarn Fawr, ABERYSTWYTH, Ceredigion, SY23 3AS, United Kingdom</td>
<td>University of Wales</td>
</tr>
<tr>
<td>University of Aberdeen, Queen Mother Library, Meston Walk, ABERDEEN, AB24 3UE, United Kingdom</td>
<td>University of Aberdeen</td>
</tr>
<tr>
<td>University of Birmingham, Barnes Library, Medical School, Edgbaston, BIRMINGHAM, West Midlands, B15 2TT, United Kingdom</td>
<td>University of Birmingham</td>
</tr>
<tr>
<td></td>
<td>Library name</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Address 1</td>
<td>Address 2</td>
</tr>
<tr>
<td>Llanbadarn Fawr</td>
<td>Aberystwyth</td>
</tr>
<tr>
<td>Meston Walk</td>
<td>Aberdeen</td>
</tr>
<tr>
<td>Medical School</td>
<td>Edgbaston</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Town/City</th>
<th>Region</th>
<th>Country</th>
<th>Postcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aberystwyth</td>
<td>Ceredigion</td>
<td>United Kingdom</td>
<td>SY23 3AS</td>
</tr>
<tr>
<td>Aberdeen</td>
<td></td>
<td>United Kingdom</td>
<td>AB24 3UE</td>
</tr>
<tr>
<td>Birmingham</td>
<td>West Midlands</td>
<td>United Kingdom</td>
<td>B15 2TT</td>
</tr>
</tbody>
</table>
COMMON SCENARIOS

▶ Correct inconsistencies in a data format

<table>
<thead>
<tr>
<th>Data you have</th>
<th>Desired data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st January 2014</td>
<td>2014-01-01</td>
</tr>
<tr>
<td>01/01/2014</td>
<td>2014-01-01</td>
</tr>
<tr>
<td>2014-01-01</td>
<td>2014-01-01</td>
</tr>
<tr>
<td>Jan 1 2014</td>
<td>2014-01-01</td>
</tr>
</tbody>
</table>

▶ Add data from an external data source

<table>
<thead>
<tr>
<th>Data you have</th>
<th>Date of Birth from VIAF (Virtual International Authority File)</th>
<th>Date of Death from VIAF (Virtual International Authority File)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Braddon, M. E. (Mary Elizabeth)</td>
<td>1835</td>
<td>1915</td>
</tr>
<tr>
<td>Rossetti, William Michael</td>
<td>1829</td>
<td>1919</td>
</tr>
<tr>
<td>Prest, Thomas Peckett</td>
<td>1810</td>
<td>1879</td>
</tr>
</tbody>
</table>
DOWNLOAD AND INSTALLATION

- You can download OpenRefine from ‘http://openrefine.org/download.html’

Windows/Linux
- Download zip file.
- Unzip the downloaded file in any directory of choice. OpenRefine should run wherever you put the unzipped folder.

Mac
- Download ‘dmg’ (disk image) file and open it.
- Drag the OpenRefine application to an appropriate folder.

- OpenRefine is a java application, and you need to have a ‘java runtime environment’ (JRE) installed

- To download and install JRE go to ‘http://java.com’ and click ‘Free Java Download’
### RUNNING OpenRefine

<table>
<thead>
<tr>
<th>Windows</th>
<th>Linux</th>
<th>Mac</th>
</tr>
</thead>
</table>
| ▶ Navigate to folder where OpenRefine is unzipped  
   ▶ Double-click ‘openrefine.exe’ | ▶ Open a terminal window  
   ▶ Navigate to folder where OpenRefine is unzipped  
   ▶ Type ‘./refine’ | ▶ Navigate to location with OpenRefine  
   ▶ Click the OpenRefine icon |
RUNNING OpenRefine

- OpenRefine interface is accessed via a web browser
- A window in your default web browser pointing at the address ‘http://127.0.0.1:3333’
CREATING OpenRefine PROJECTS

File formats supported include:
- TSV (tab-separated values)
- CSV (comma-separated values)
- Excel
- JSON (javascript object notation)
- XML
- Google Spreadsheet
SELECTING YOUR IMPORT OPTIONS

Options to ensure that the data gets imported into OpenRefine correctly depending on file type

1. Set the ‘Character encoding’ to ‘UTF-8’

2. First line as header

Options for:
• Comma delimited files
• Tab delimited files
• Custom delimiters e.g. ‘;’
Your saved project

- Created project are automatically opened
- Projects are saved as they are worked on. No need to save copies as you go along
- Data displays in tabular format.
- A Row represent a ‘record’ in the data, a column represents information type

<table>
<thead>
<tr>
<th>8287 rows</th>
<th>Show as rows records</th>
<th>Show 5 10 25 50 rows</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>Identifier</td>
</tr>
<tr>
<td>2</td>
<td>00000201600</td>
<td>London, Virtus &amp; Ivonon</td>
</tr>
<tr>
<td>3</td>
<td>00000203000</td>
<td>London</td>
</tr>
<tr>
<td>4</td>
<td>00000404720</td>
<td>London</td>
</tr>
<tr>
<td>7</td>
<td>00000201901</td>
<td>London</td>
</tr>
</tbody>
</table>
THE INTERFACE

Open a new Instance of OpenRefine:
• See all saved projects
• Create new project
• Import projects

Your saved projects

More Options
REORDER, RENAME AND REMOVE COLUMN

Many operations are accessed from the drop down menus at the top of each column or the top of the first column (labelled ‘All’)

**Reorder**

**Option 1**
1. Go to the ‘All’ column and accessed from the drop down menus
2. ‘Edit columns’ -->
   ‘Re-order / remove columns…’
3. Click and drag the column to its desired position
4. Click ‘OK’

**Option 2**
1. Go to the respective column and accessed from the drop down menus
2. ‘Edit column’ -->
   ‘Move column [to] [beginning, end, left, right]’

**Rename**

1. Go to the respective column and accessed from the drop down menus
2. ‘Edit column’ -->
   ‘Rename this column’

**Remove**

**Option 1**
1. Go to the ‘All’ column and accessed from the drop down menus
2. ‘Edit columns’ -->
   ‘Re-order / remove columns…’
3. Click and drag the column to the right section labelled ‘Drop column here to remove’
4. Click ‘OK’

**Option 2**
1. Go to the respective column and accessed from the drop down menus
2. ‘Edit column’ -->
   ‘Remove this column’
SORTING DATA

Sort Action

1. Go to the respective column to be sorted by and accessed from the drop down menus
2. Click on ‘Sort…’
3. Choose options and sort positions of blanks and errors
4. Click ‘OK’

A new ‘Sort’ drop down

‘Sort’ are temporary. The ‘Sort’ drop down menu allows for
- Undoing to sort to it’s original ‘unordered’ state.
- Remove existing sorts
- Making the sort Permanent.
- Amend the existing sort (e.g. reverse the sort order)

More column ‘Sort’ options

More options for amending are added to ‘Sort’ under the sorted column.
1. Go to the respective sorted column and accessed from the drop down menus
2. ‘Sort’ -->
3. Choose appropriate option
Exercise 1

Find the ‘Date of Publication’ column and sort the information by date of publication in descending order.

Move the title column to be the second column (after the “Identifier” column)
FACETS

- Helps get an overview of data
- Helps bring more consistency to data
- With Facets we can:
  - Group all common values that appear in a column
  - Filter the data by these values
  - Edit values across many records at the same time
- Facet information appears in the left hand panel
TEXT FACETS

1. Go to the respective column of interest and accessed from the drop down menu

2. ‘Facet’ --> ‘Text facet’

3. The facet will then appear in the left hand panel with all possible values in the column

4. Try this on the ‘Issuance Type’ column

5. You can see this contains two values ‘continuing’ and ‘monographic’
TEXT FACETS MANIPULATIONS

Filter
Option 1
1. Click on a value in the Facet
2. Click on the again to undo filter

Option 2
1. Pass the cursor over a value in the Facet
2. Click on the ‘Include’

Include/Exclude
1. Pass the cursor over a value in the Facet
2. ‘include’
   Appears if value is not already filtered or not included in the filter.
   Click ‘include’ to filter value or add the value to existing filter.
3. ‘exclude’
   Appears if value is already included in the filter.
   Click ‘exclude’ to remove value from the filter

Edit
▶ Used to change the text value of the facet
▶ Values are edited across same valued records at the same time
1. Pass the cursor over a value in the Facet
2. Click on ‘edit’
3. Change the value in the text box
4. Click ‘Apply’

Reset/Invert
▶ Shows under the ‘Issuance Type’ tab of the facet if there is at least one filtered value
▶ ‘reset’
   Undo all changes (edits/filters) under the facet
▶ ‘invert’
   Negates/reverses/flips the data selection of the filter.
   Excludes filtered values from displayed data
TEXT FACETS MANIPULATIONS

Sort by

Facets can be sorted by
- 'name' (value)
- 'count' (number of record)

Cluster

- Helps find groups of different cell values that might be alternative representations of the same thing, e.g. 'New York' and 'new york', 'Gödel' and 'Godel'
- The various value can be reconciled and merged into one value
- Click on 'Cluster' button on the Facet panel
- Can be accessed also by 'Column dropdown' --> 'Edit cells' --> 'Cluster and edit …'
- Clustering algorithms can be varied under the 'Method' and the 'Keying Function' dropdown
- Experimentation may be required to choose best clustering algorithm a particular data set
- Using different algorithms highlights different clusters
Exercise 2

- Create a text facet for the Edition column
- Sort the facet by ‘count’ to see the most common values
- Pick a facet value that refer to a ‘second edition’ and edit them using a consistent wording
- Attempt to do this for all facet values that refer to ‘second edition’
  [Hint: use the ‘Cluster’ feature]
MORE ON FACETS

- **OpenRefine** also supports other types of facet
  - Numeric facets
  - Timeline facets (for dates)
  - Scatterplot facets
  - Custom facets
NUMERICAL AND TIMELINE FACETS

- Displays graphs instead of lists of values
- Includes ‘drag and drop’ controls used to set a start and end range for filtering displayed data
- Included check buttons to select more options.
CUSTOM FACETS

- Allows you write your own custom facets

- Some of the default custom facets include:
  
  - **Word facet**
    Breaks down text into words and counts the number of records each word appears in
  
  - **Duplicates facet**
    - Generates a binary ‘true’ or ‘false’ valued facet
    - Rows appear under ‘true’ if the value in the selected column is an **exact match** for a value in the same column in another row (duplicated values)
  
  - **Text length facet**
    - Generates a numeric facet based on the length of the text for the selected column in each row
    - This is useful for spotting incorrect or unusual data in a field where specific lengths are expected e.g. for Date columns of format ‘mm/dd//yyyy’ any row with a text length either than 10 for that column is incorrect or incorrectly formatted
  
  - **Facet by blank**
    - Generates a binary ‘true’ or ‘false’ valued facet
    - Rows appear under ‘true’ if they have no data in that column
SCATTERPLOT FACETS

▶ Less commonly used
▶ For further tutorial of Scatterplot Facets visit
  ‘http://enipedia.tudelft.nl/wiki/OpenRefine_Tutorial#Exploring_the_data_with_scatter_plots’

A tad bit more

▶ **OpenRefine** limits the number of values allowed in a single facet
▶ This is to ensure the application does not perform slowly or run out of memory
▶ Facet on columns with many unique values will be very large
▶ Application may slow down or **OpenRefine** will not create the facet
Exercise 3

Find all publications without a date of publication
Do you notice anything about these records?
[Hint: look at the “Place of Publication” column]
Aside Facets, ‘Text filter’ filter the data filter data

VERY IMPORTANT NOTE: Any OpenRefine operation only apply to rows that match the filter; i.e. the data currently being displayed

1. Go to the respective column of interest to be filtered and accessed from the drop down menu
2. ‘Text Filter’
3. Type your filter string in text box
4. You can choose the ‘case sensitive’ check box or ‘regular expression’ check box
**CHANGE HISTORY; UNDO AND REDO**

- **OpenRefine** lets you **undo** and **redo** any number of steps in cleaning the data.
- **History** of all actions are shown under the ‘**Undo/Redo**’ tab on the left hand panel.
- To **undo**, click on the last step you want to preserve in the history.
- This will automatically undo all changes after that step.
- Remaining steps below will be **greyed out**.
- Greyed out steps can be reapplied by simply clicking on them.
- If you **undo** a set of steps and perform new transformations, the greyed out steps will disappear.
SAVING HISTORICAL STEPS

- A step to be re-applied later to current work or to a different project
- Click on the ‘Extract…’ button
- Select the steps you want to save
- Copy the transformations included in the selected steps the ‘JSON’ format
APPLYING SAVED ‘JSON’ FORMAT TRANSFORMATIONS

▶ To apply a saved in this ‘JSON’ format transformation;
▶ Go to the ‘Undo/Redo’ tab
▶ Click on the ‘Apply…’ button
▶ Paste the paste in the ‘JSON’ script
▶ Click ‘Perform Operation’
More complex changes can be achieved by **Transformations**; They include

- Splitting data in a single column into multiple columns e.g. splitting date into Month, day and year
- Standardizing a column’s data format without changing the values e.g. removing punctuation or standardizing a date format
- Extracting a particular type of data from a longer text string e.g. finding ISBNs in a bibliographic citation

‘**GREL**’ (General Refine Expression Language) is the language used for writing Transformations

For more on ‘GREL’ visit

PERFORMING TRANSFORMATIONS

1. Go to the respective column of interest and accessed from the drop down menu

2. ‘Edit Cells’ --> ‘Transform….’

3. Type the ‘GREL’ expression in the ‘Expression’ textbox

4. The simplest expression is ‘value’; which means the current value in the column i.e. ‘make no change’

- The tabs include;
  - Preview - shows effect of transformation on the first 10 rows of the column data
  - History - List of previous transformations They can be reused immediately or ‘starred’ for easy access later
  - Starred - List of ‘starred’ transformations
  - Help - List of GREL functions and brief information on how to use them
WRITING GREL FUNCTIONS

- ‘GREL’ s are written by giving a value to a GREL function
- GREL functions may take more than one parameter which may be optional
- Two ‘GREL’ syntaxes are supported. Either is valid and the choice of syntax down to personal preference
  - value.function([options])
  - function(value, [options])
## SOME COMMON GREL FUNCTIONS

<table>
<thead>
<tr>
<th>GREL expression</th>
<th>Examples</th>
<th>Action carried out</th>
</tr>
</thead>
<tbody>
<tr>
<td>toUppercase(string)</td>
<td>toUppercase(value)</td>
<td>Converts the current value to uppercase</td>
</tr>
<tr>
<td></td>
<td>value.toUppercase()</td>
<td></td>
</tr>
<tr>
<td>toLowercase(string)</td>
<td>toLowercase(value)</td>
<td>Converts the current value to lowercase</td>
</tr>
<tr>
<td></td>
<td>value.toLowercase()</td>
<td></td>
</tr>
<tr>
<td>toTitlecase(string)</td>
<td>toTitlecase(value)</td>
<td>Converts the current value to titlecase (i.e. each word starts with an uppercase character and all other characters are converted to lowercase)</td>
</tr>
<tr>
<td></td>
<td>value.toTitlecase()</td>
<td></td>
</tr>
<tr>
<td>trim(string)</td>
<td>trim(value)</td>
<td>Removes any ‘whitespace’ characters (e.g. spaces, tabs) from the start or end of the current value</td>
</tr>
<tr>
<td></td>
<td>value.trim()</td>
<td></td>
</tr>
<tr>
<td>substring(string, number from, optional number to)</td>
<td>substring(value, 0, 4)</td>
<td>Finds the first four characters of the current value</td>
</tr>
<tr>
<td></td>
<td>value.substring(0,4)</td>
<td></td>
</tr>
<tr>
<td>replace(string, string to find, replacement string)</td>
<td>replace(value,&quot;a&quot;, &quot;b&quot;)</td>
<td>Find the letter ‘a’ in the current value and replace it with the letter ‘b’</td>
</tr>
<tr>
<td></td>
<td>value.substring(&quot;a&quot;, &quot;b&quot;)</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td>“Prefix: ” + value</td>
<td>Adds (concatenates) the word “Prefix” to the front of the current value</td>
</tr>
</tbody>
</table>

[2] OpenRefine Workshop Handout Developed by Owen Stephens on behalf of the British Library
Exercise 4

Create a facet based on the Date of Publication column

Sort the facet by ‘name’

What are common issues with the values in this list?

Use the ‘replace’ GREL expression to remove the characters [ , ] and ? from the Date of Publication column
## OPENREFINE DATA TYPES

- Every piece of data in OpenRefine has a ‘type’ referred to as ‘data types’
- Transformations let you convert data from one data type to another appropriate data type
- The data types supported are:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String</strong></td>
<td>Text values</td>
</tr>
<tr>
<td><strong>Number</strong></td>
<td>Numerical values (integers, decimals and exponentials)</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>Date and time values</td>
</tr>
</tbody>
</table>

### Boolean
- Binary ‘true’ or ‘false’ data values
- Can be used directly in OpenRefine cell
- But are more often used in transformations as part of a GREL expression
  - E.g. `value.contains("test")` generates a ‘true’ or ‘false’ Boolean value

### Array
- List of values
- Represented by the use of square brackets `[]` containing a list of values surrounded by inverted commas “,” and separated by commas ‘,’
  - E.g. List of days [“Monday”,“Tuesday”,“Wednesday”]
- Usually the result of a transformation using GREL
  - E.g. `value.split(“,”)` can split “Monday, Tuesday, Wednesday” into [“Monday”,“Tuesday”,“Wednesday”]
REGULAR EXPRESSIONS (REGEX)

- Representing patterns in text strings
- Can be used to search for matching texts with pattern represented by the `Regex`
- Typically surrounded by ‘/‘ characters
- To write a `Regex` you need to know the special syntax used to represent different types of characters that can occur in a text string
**REXEG SYNTAXES** [3]

E.g. `/organi.e/` - would match both ‘organise’ and ‘organize’

`/organi[sz]e/` - more specific

<table>
<thead>
<tr>
<th>Type of character</th>
<th>Regular expression syntax</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any character</td>
<td></td>
<td>A ‘period’ can represent any character at all - essentially a ‘wildcard’</td>
</tr>
<tr>
<td>A list or range of characters</td>
<td><code>&lt;list/range to be matched&gt;</code></td>
<td>You can put a list of characters or a range of characters inside square brackets to match any of the characters in the list. e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[ABC] matches A or B or C (note this is case sensitive).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[A-Z] matches any uppercase letter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[A-Za-z0-9] matches any upper or lower case letter or any digit</td>
</tr>
<tr>
<td>Any digit</td>
<td><code>\d</code></td>
<td>The syntax <code>\d</code> will match any single digit character (equivalent to [0-9])</td>
</tr>
<tr>
<td>Any 'word' characters</td>
<td><code>\w</code></td>
<td>The syntax <code>\w</code> will match any character that can be part of a 'word'. In practice this means any letter (upper or lower case), any digit or the underscore character. (equivalent to [A-Za-z0-9_] )</td>
</tr>
<tr>
<td>Any 'whitespace' characters</td>
<td><code>\s</code></td>
<td>The syntax <code>\s</code> will match any 'space' type character - such as a space, a tab, or a newline.</td>
</tr>
<tr>
<td>The start of a string</td>
<td><code>^</code></td>
<td>The syntax <code>^</code> will match the start of the string - useful if you want to find strings that start with a specific pattern</td>
</tr>
<tr>
<td>The end of a string</td>
<td><code>$</code></td>
<td>The syntax <code>$</code> will match the start of the string - useful if you want to find strings that end with a specific pattern</td>
</tr>
</tbody>
</table>
REGEX REPETITIVE OPERATORS

- Allow you to say how many times a character or pattern is repeated
- Apply to only the character or expression immediately preceding the operator
- Curly brackets containing one or two numbers separated by comma “,” specify exact numbers of repetitions, or a maximum/minimum number of repetitions respectively

E.g.
/a{2}/ - Matches the letter ‘a’ appearing **twice** (i.e. matches ‘aa’)
/a{2,4}/ - Matches the letter ‘a’ appearing a **minimum of two** times or a **maximum of four** times (i.e. matches any of ‘aa’, ‘aaa’, ‘aaaa’)

<table>
<thead>
<tr>
<th>Repetition character</th>
<th>Meaning</th>
<th>Explanation/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>The preceding character/ expression can be repeated any number of times (including zero)</td>
<td>The regular expression <code>/.*/</code> represents any text string at all (any character repeated any number of times)</td>
</tr>
<tr>
<td>+</td>
<td>The preceding character/ expression can be repeated one or more times</td>
<td>Unlike the <code>*</code> using a <code>+</code> means the character must appear at least once. The expression <code>/head\s+/rest/</code> would match ‘head rest’ (one space), ‘head  rest’ (two spaces) but not ‘headrest’.</td>
</tr>
<tr>
<td>?</td>
<td>The preceding character/ expression can be repeated 1 or zero times</td>
<td>Essentially makes a character optional in a regular expression. The regular expression <code>/colou?r/</code> would match both ‘colour’ and ‘color’</td>
</tr>
</tbody>
</table>
ONLINE TUTORIALS & RESOURCES

▶ CEAS Library OpenRefine resources

http://guides.libraries.uc.edu/OpenRefine

▶ Online OpenRefine links & tutorials

▶ OpenRefine Resources

• 3 great introductory videos
  https://github.com/OpenRefine/OpenRefine/wiki/Screencasts

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

▶ British Library

Questions?
Thank you for attending the workshop!!

@ceaslibrary #ceaslibrary