MARC21 : https://www.loc.gov/marc/bibliographic/

Tasks:

* Facet/clusters to standardise information in the columns
* Regular expressions to edit text column (column icon 🡪 edit cells 🡪 transform)
* Switch between record mode and row mode
* Undo/Redo function to go back to previous stages
* Reconcile/link Wikidata entries (column icon 🡪 reconcile 🡪 start reconciling) – can only be done on clean text and column needs to be split (no special characters indicating metadata structure)
* Standardise and normalise the fields with clustering (isolate the content or work on original columns – decide what is more efficient [and then I assume put this in the report]). **Never delete original version – work on a separate column**
* Identify fields on which it can be interesting to work and isolate the content from the tags to apply Wikidata reconciliation (for instance people, or places, or titles, or institutions, where available). If the software cannot search for the proper text in Wikidata, there won’t be any match.
* Be careful not to lose information. If, for instance, you decide to isolate one specific subfield from one column, make sure that the code of the subfield is preserved somewhere (see below for a possible strategy)

A screenshot of a computer screen

Description automatically generated

Workflow: we need to document changes performed in dataset using GitHub (instructions on Toledo)

**Submission:**

1. **Report (4 pages max):**
   * **team member names, s number, email**
   * **brief description of dataset**
   * **description of steps taken and choices made**
   * **description of final dataset (differences from original)**
   * **description of workflow**
   * **evaluation of the tool**
   * **examples of GREL used**
2. **Final Dataset as .csv file**

Tutorial steps: <https://programminghistorian.org/en/lessons/cleaning-data-with-openrefine>

1. remove blank rows (via facet)
2. remove duplicates (*permanently* sort unique value, blank down, facet) – needs double checking!
3. to fix problems in categories/shared rows – need to keep an eye on number of records/rows for inconsistencies:
   1. split rows (‘Edit cells’, ‘Split multi-valued cells’)
   2. view issues in metadata (customised facet, ‘facet by blank’)
   3. overview of data (text facet, sort by xyz // numerical facet, sort by xyz)
   4. cluster: solve issues regarding case inconsistencies, incoherent use of either the singular or plural form, and simple spelling mistakes. Lots of different methods and functions to play with
   5. re-join rows (edit cells, join multi-valued cells, use ‘|’ as a separator)
4. use “edit cells” 🡪 “transform” to fix problems (uses GREL – OpenRefine’s language) – use different regular expressions to mass modify values e.g.
   1. type inconsistencies
   2. repeated info
5. Export as CSV file

<https://handsondataviz.org/open-refine.html>

* Can change text to numbers (Edit cells > Common transforms > To number)
* Can remove/replace things (Edit cells > Transform > value.replace(‘this’,’that’)

GREL functions:

<https://openrefine.org/docs/manual/grelfunctions>

Regular expressions:

<https://openrefine.org/docs/manual/expressions#regular-expressions>

<https://docs.oracle.com/javase/tutorial/essential/regex/>

<https://gist.github.com/pmgreen/6e133c5dcde65762d29c>

<http://www.rexegg.com/regex-quickstart.html>

Regular Expressions:

* **start and end with /**
* /a/: = match all “a” in text
* to build efficient regular expressions, some characters are “special characters” (meta-characters) used to identify patterns

A screenshot of a computer

Description automatically generated

*can we run through slides on Toledo?* regex\_open\_refine.pptx

Instructions and Description of MARCXML derived files:

* replace:
  + replace(value, /a{2}/, “ab” = sequence of 2 ‘a’s replaced with ‘ab’
  + replace(value, /(\d\d)\d?, “$1”) = sequence of 3 digits replaced with first two
* conditional: if(value==”x”, “y”, value) = if value is x, change it to y; if not, leave it
* value between others: replace(value,/x(.y)z/, “$1”) = will leave y
* Field number = column header
* 1st indicator, 2nd indicator, subfield = in cell (separated by $c – does this mean we can replace($c, “ “)???)
* New field: ^ before $