Sept. 23, 2014

Notes for DST4L training- Free Your Metadata
More OpenRefine!

Everyone here today was at Philips yesterday for part 1
What do people want to use Refine for?
What kind of data do we work with?

- Collection management
- University's data archives for info on students and courses
- Easier interface for analyzing data
- Work for students - cleaning up addresses for mapping
- MARC/biblio cleanup
- Bibliographic data in Excel
  - Clean up data, reconciling names
  - Collapsing data back into single field (expanded it yesterday)
- Journal articles' metadata (National Lib of Medicine)
- Circulation data (exported to Excel)
- Numeric ID - Text ID reconciling
- Table joins/lookups
- Finding stories in big data – DDJ = Data Driven Journalism
- Data mgmt plan cleanup

What did people hear about yesterday that we wat to know more about?

- Entity extraction
- Multivalue data
- Linking to external data sources (LCSH)
  - Reconciling with different data sources
  - After reconciling, pulling in additional information

Some Review from yesterday some **Refine Basics**:

- Format conversion - read and write many formats
- Can start with TSV and generate RDF (but so many others are possible)
- Can unpack zip files in one step
  - Zipped tar files? It can do that
- Supports many kinds of files
  - Even JSON or XML
- Can create project command for an API and it will suck all that info into itself in JSON format and convert that to usable table
  - If the API has a max, you can cycle through page #s
- Can paste into clipboard
- Can take Google spreadsheet data
- Refine looks at the file and takes a guess at what the format is going to be.
- "Export project" exports as OpenRefine document with all the info that you have (like undo data?)
- "Export"
  - "Custom tabular exporter" - many tools for output design here
  - "Templating" allows export in different expression languages - can use customize to a different format than they provide
- Undo history records all the actions I have done and thus preserve provenance

- Project creation side, options made there are not recorded
  - Until this is updated, Tom recommends including this info in Project Name
- Tom's Dropbox has all the projects from last year - also tutorials
  - He'll make it available later
- Parse cell text into number, dates, … can cause problems - it's automatically checked, he recommends unchecking
- Can ignore the first ## rows if there is a header
- Can make column headers import or not
- Can limit the number of rows you want to import (cannot randomize which rows import)

Our sample dataset is from Tish library

- Analyzed the collection data for what resources they must maintain and where they can deaccession, weed, or move off site
- Uneven data due to inconsistent cataloging
  - Normalize cataloging inconsistencies?
- What might be candidates for scanning?
- Need micro policies for making broader decisions
- Could we make an OCLC link dynamic?
- Visualization tools for age of things?
- Editions as different records - want to see all editions together

OpenRefine

- Good at slicing and dicing in different ways
- I.E. how publication dates very by number of checkouts
- Can use Facets (especially with numeric) to see outliers
- To change the max number of rows displayed go to: 127.0.0.1:3333/preferences
- Change numeric text or vice-versa go to Edit cells ->Common transforms -> to number (or to date)
- Permalink button will give you a link to your dataset with all facets and such at that moment - essentially a way to save your view
  - e.g.
    http://127.0.0.1:3333/project?project=1566736575776&ui=%7B%22facets%22%3A%5B%7B%22c%22%3A%7B%22type%22%3A%22timerange%22%2C%22name%22%3A%22Last%20Charge%20Date%20-%20Item%22%2C%22expression%22%3A%22value%22%2C%22columnName%22%3A%22Last%20Charge%20Date%20-%20Item%22%2C%22selectTime%22%3Atrue%2C%22selectNonTime%22%3Atrue%2C%22selectBlank%22%3Atrue%2C%22selectError%22%3Atrue%2C%22from%22%3A564382800000%2C%22to%22%3A1399953600000%7D%7D%2C%7B%22c%22%3A%7B%22type%22%3A%22timerange%22%2C%22name%22%3A%22Last%20Add%20Date%20-%20Title%22%2C%22expression%22%3A%22value%22%2C%22columnName%22%3A%22Last%20Add%20Date%20-%20Title%22%2C%22selectTime%22%3Atrue%2C%22selectNonTime%22%3Atrue%2C%22selectBlank%22%3Atrue%2C%22selectError%22%3Atrue%2C%22from%22%3A539413200000%2C%22to%22%3A1340078400000%7D%7D%2C%7B%22c%22%3A%7B%22type%22%3A%22timerange%22%2C%22name%22%3A%22Publication%20Year%22%2C%22expression%22%3A%22value%22%2C%22columnName%22%3A%22Publication%20Year%22%2C%22selectTime%22%3Atrue%2C%22selectNonTime%22%3Atrue%

2C%22selectBlank%22%3Atrue%2C%22selectError%22%3Atrue%2C%22from%22%3A-725828400000%2C%22to%22%3A1325394000000%7D%7D%5D%7D

- Can add columns as needed
  - "Add column based on column title
    - Value.split('?')[0].split(', by')[0].split(':')
    - "prefix.value"
    - This can be used to pull out sub-titles or anything else that is smashed in with other things
    - History of each used phrase is saved
    - Starred (if you use it often might want to star it)
    - Use Help tab for syntax specific questions
    - Help page in Wiki - Reference > Expressions > Functions
      - Or search for recipes (commonly used functions that people have saved)
    - Search for "Tutorials" or "Understanding Expressions"
  - Google Refine Expression Language (GREL)
  - Jython (is actually Python language supported through Java or Javascript somehow)
- Already existent columns = "star" and "flag"
  - Good for manual review -
  - Make up meaning for each of these
  - Can filter by these
- Clustering
  - Key collision - looks through whole dataset for
  - Metaphone3(English) and cologne-phonetic(German specific) are both idea for people's names!
  - Key collision + metaphone3 = great for conference proceedings
  - Nearest neighbor
    - Levenshtein - string distance match (good for catching type-os)
    - PPM
- WARNING - if you have facets selected - only certain rows will be impacted if you are doing
- Splitting and Joining subsets within column
  - Choose column header Edit cells -> Split multi-valued cells…
  - Choose column header Edit cells-> Join multi-valued cells…
  - Make sure you don't choose a delimiter that is in your data.
    - Can check for this using "Text filter" and searching for the delimiter you want to use
- Refine snapshots your work every 5 mins.
- Be careful if you are low on memory or disk space
- Record display mode
- Flipside of blank down is fill down
- This will fill the info from the row above into the blank row below.
  - If you want to do analysis on all the rows with the category but keeping them with their original ID info or whatnot
  - Can sort by multiple columns at once and can switch which one is the primary sort vs. secondary sort
  - This is a way to group records by category for example and make them into "record" by category
  - If rows are indented it is seen as a record with the non-indented row above

Lunch break

Starting with list from beginning and general people's question

Use API to expand your data - learn more about your information

API - Application Programming interface

- Most are web based (long complicated URLs)
- PRESTO API examples are in email
- Browser plugin or extensions that will render results nicely for you
  - e.g. JSON View or JSON Pretty Print
  - Plugins are available for each browser
- Information will be returned in different formats
- Refine can take the URL from the API and render it (as mentioned before)
- Create project from web address - Paste API URL
- Select XML or JSON depending on API
- ResultSet is entire result set
  - All items are available under this
  - Select an item and Refine can flatten it out
  - B/c it's tree structured the column headers become the whole tree structure down to the place where our info is
  - Orange bars at top will show groupings of records
  - Depending on your data source (API you choose) your data will be more or less messy/clean
- JSON vs JSONP: JSONP won't work
  - Refine doesn't support information wrapped in ()
- Information about APIs?
  - Google what trying to do: hopefully someone has done it before
  - Look at documentation on API, look for description of URL formats
  - Template replacement is your best bet (see examples given)
- Use APIs to extend the data that we have
  - OCL numbers
  - Unique and well managed info that we can match to ours
- If we have DOIs and want Pubmed IDs…
  - Maybe they have a SPARQL Endpoint
- Can use GREL to add column based on a URL formed by concatenating the URL for the API and the contents of the column
  - Add column by fetching URLs based on column [COLUMN NAME]
    - Expression is "URL_for_API"+"value"
    - Remember sometimes API won't return results for what you're looking for
    - Will give back info however API likes to - in this case we got JSON
  - Select throttle delay (defaults to 5seconds, may want to shorten it)
  - See variables index to grab different values from column as needed
  - To get the doi out of the key/value pairs received from API
    - Cells['JSON PMID'].value.parseJson() ['doi']
    - See that preview looks correct
    - Click OK and you get the DOIs
    - Cells['JSON PMID'] means "grab the equivalent cell in the column titled['column name']"

        - This works when info is received in JSON not XML
    - Use this for Strong Identifiers only (DOIs, ISBN, ISSN, ext.)
    - To get the XML back is more challenging
        - Selectors always return an array even if there is only one term
        - Query for XML is more complex and returned info is ugly
- Freebase has many different bases that it draws from and can be queried against
    - Can add columns from Freebase based on what interested in:
        - e.g. date of birth or nationality
    - Database must be up in order to get info from it - Freebase is down
    - Can extend datasets iteratively (got country of origin can now find name of leader for that country)
    - Take list of nationality and want MARC 210 codes for country
        - See help for info but want cross section (uses cell instead of value, needs new projectName and then the name of the column in the other project)
    - There are collective functions and Boolean functions available within Refine: e.g. If discontinued, skip
    - Regular expressions = reg ex
        - Very powerful but can be confusing
        - Once you know a few, you'll be able to do a lot
        - There are good websites where you can put in sample data and get the information back

- NOTE: If lots of info is received (or being manipulated) - Refine will get slow

    - Address regulations - Geolocations
    - Geocoding (information entered by people)
        - Normalize things
        - Companies exist to do this but to do it yourself
        - Use faceting and clustering to regulate your data
            - Blvd or Boulevard (find with dictionary and word facet)
        - OpenRefne wiki has whole page on geocoding - turn address into longitude and latitude for example
        - Google Terms and Service are becoming more restrictive on geolocation - now can only plot on their maps - want traffic to their maps which they can monetize (purely business related)
        - Openstreetmap.org may do this, they have an API so they may be able to do this soon or even now, but not as developed as Google's
        - Google fusion tables
        - Allow for searching based on other criteria than the address
    - Text search filter for faceting (regular expressions)
        - [0123456789]\(
            - This will find all instances of a number followed by a(
        - [0123456789]-\(
            - This finds all instances of a number followed by a - by a (
        - ^[0123456789]\(
            - This finds all instances of a number followed by a - by a (at the beginning of a string
        - [0123456789]\($

        - This finds all instances of a number followed by a - by a (at the end of a string
- OCLC APIs are quite closed - must be a member of an institution and such to get a devkey
- In the add column based on something: 'value' means the value in the cell of the given column.
- Use + to add the value to the string of the URL
- Use parseHtml() when API returns XML
    - Newer APIs tend to return JSON
    - The best new APIs return data in multiple formats based on request
        - RDF, JSON, XML are most but there may be others
- WorldCat API is protected
- RDF versions at: id.loc.gov/authorities/names/html
    - Linked data service
- VIAF is linked to from there but can go to: viaf.org/search
    - Shows how different things are cataloged around the world
- OCLC and LC focusing on authors more than titles
- Janitorial work of being a Big Data Scientist
    - Most of the work is finding the data
    - Formatting the data
    - Organizing the data in useful ways

- GitHub - OpenRefine - Issues - File a new issue (for problems)
- Openrefine.org
- Video introductions from GoogleRefine are still generally good for OpenRefine and still available on their site
- Virtual machine image for sharing projects or working on large projects is in the works
- Watch for Browser issues and server issues for these
- Up to 1million rows of design space, but your limit is your computer's memory and such.
- Workload sharing?? Don't have a good system to do this unfortunately.
    - Would have to portion out the dataset in order to let multiple people work on it at the same project at the same time
    - Can also run off the server, but then have to schedule doing work on OpenRefine with other people
- If the memory usage is in the 90% range there is something WRONG
    - Close it out, allocate more memory resources, and try again
- Can add new reconciliation services (beyond just RDF)
    - e.g. Open phylo / Phylogenics
    - Reconciliation services on GitHub/OpenRefine page
        - Try Googling
- Opensource project that provides reconciliation tool that is locally hosted (i.e. not available online)
- Reconciliation and such should become easier overtime

Next session on GitHub

More textural, not overly technical/program-y

Gitenberg (http://gitenberg.github.io/) - took all Gutenberg titles and put them into Git repositories

A way to become more embedded in the process