Uncovering Hidden Datasets in DRUM

David Durden & Allison Buser // UMD Libraries Research & Innovative Practice Forum // June 3, 2021

UNIVERSITY LIBRARIES

Introduction

The Digital Repository at the University of Maryland (DRUM) has been accepting research data deposits for over a decade. If one searches DRUM with the search term, "datasets," using the "All of DRUM" search facet, over 4,000 results will be returned. In contrast, searching with "datasets" as a keyword will return 27 results, and searching by type using "dataset" will return 88 results. These discrepancies are explained by both the uncontrolled usage of Dublin Core Metadata Terms during the self-submission process and the tokenized search capabilities of Solr, which provides full-text search results across most documents deposited in DRUM.

However, knowing why the discrepancies exist prompted the authors to ask: Exactly how many datasets are actually in DRUM? The authors attempted to answer this question by analyzing metadata records from DRUM through two approaches: 1) Using DSpace Admin Tools and analyzing results in Microsoft Excel, and 2) Exporting selected metadata elements from Solr and analyzing dc.type and dc.format.mimetype values using R Studio.

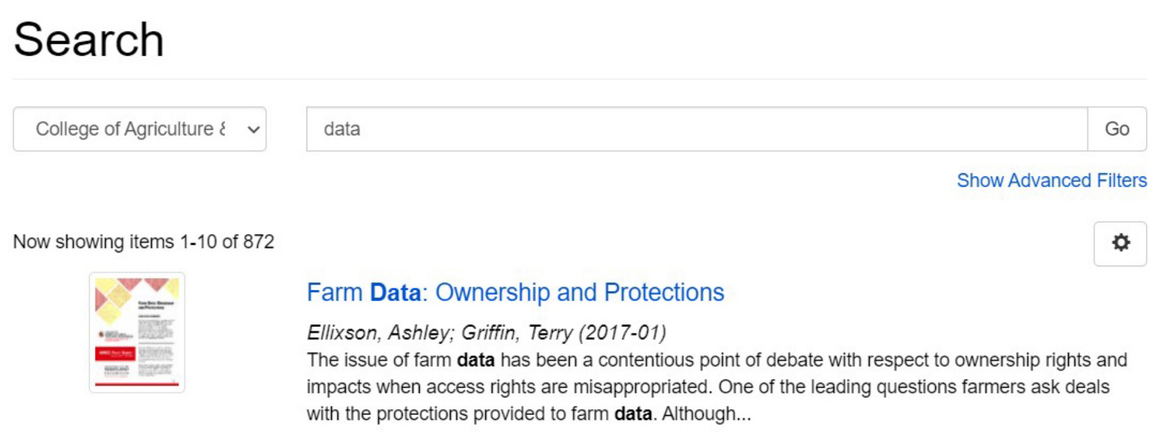
Approach 1: Admin Tools & Excel

This approach to discovering hidden datasets utilized DRUM's front-end search tools and functions in Microsoft Excel. DRUM includes a tool which allows users with an administrative account to export an Excel sheet of metadata for all records that appear in a specific search. While hidden datasets are not easily identifiable in this sheet based on metadata alone, some metadata clearly disqualifies a record from containing datasets. Creating rules based on disqualifying metadata in Excel's conditional formatting feature allows obvious non-datasets to be easily highlighted and eliminated. Through process of elimination, this approach sought to produce a more manageable and accurate list of potential hidden dataset records that could then be manually examined at the item level.

Method

1. Keyword selection. There are almost 90 known datasets in DRUM, deposited using the term 'dataset' as their dc.type metadata. Many of these dataset records have associated articles published in journals outside of DRUM. From further analysis of these external records, it was observed, perhaps obviously, that each of the published works associated with the known datasets use some form of the word "data" in their text. Based on this knowledge it was assumed that most works with hidden datasets attached to them in DRUM might behave similarly.

2. DRUM query and metadata download. The term 'data' was queried in each DRUM collection (see figure 1). The metadata of all returned records was then downloaded in an Excel file using the 'Export Search Metadata' admin tool.



Search

College of Agriculture &

data

Go

Show Advanced Filters

Now showing items 1-10 of 872

Farm **Data**: Ownership and Protections

Ellixson, Ashley; Griffin, Terry (2017-01)

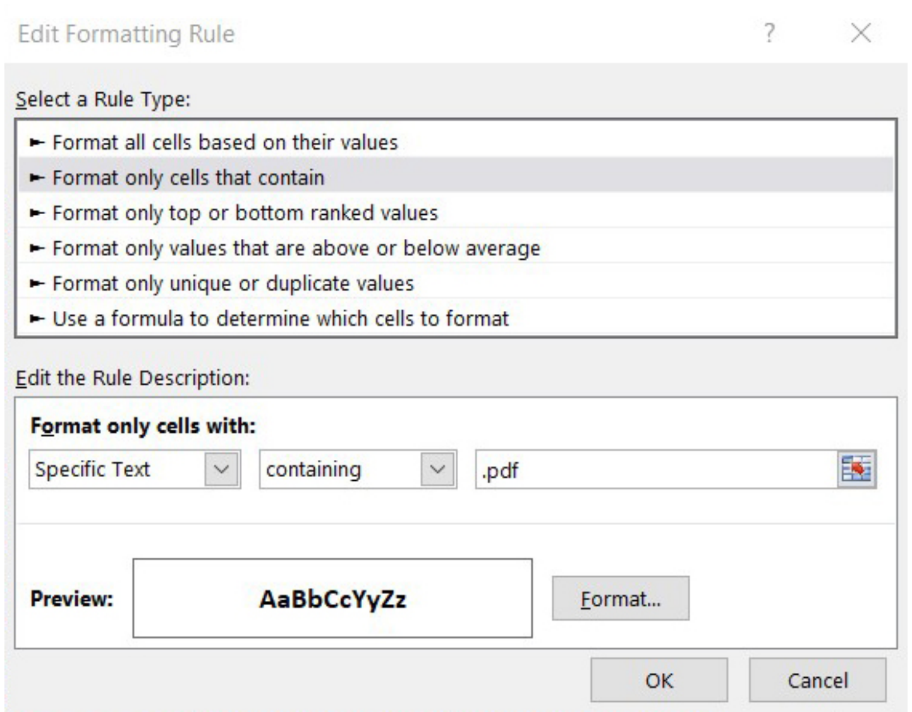
The issue of farm **data** has been a contentious point of debate with respect to ownership rights and impacts when access rights are misappropriated. One of the leading questions farmers ask deals with the protections provided to farm **data**. Although...

Figure 1. Querying the College of Agriculture and Natural Resources collection returned 872 records for metadata download.

3. Filter known datasets. Within the Excel file, the dc.type metadata column was filtered first and eliminated any records labelled as 'dataset', as these are already known.

4. Apply conditional formatting to single bitstream records with PDFs. The category dc.description.provenance automatically captures information such as the number of bitstreams and the names of the files uploaded to records. Conditional formatting rules were primarily applied to this column to identify records that could be eliminated from consideration. The first rule instructed Excel to red highlight any record with the specific text 'No. of bitstreams: 1.' A subsequent rule changed any cell containing '.pdf' to bold text. The records with the combined red highlight and bold text formatting were then eliminated from consideration (see figure 2).

5. Adapting conditional formatting rules to ProQuest ETDs. Many records in DRUM are electronic theses and dissertations imported from ProQuest. These ETD records usually include at least two bitstreams, the thesis or dissertation file and a 'DATA.xml' file with further metadata. Another conditional formatting rule was applied in dc.description.provenance to yellow highlight records with the specific text 'No. of bitstreams: 2.' A second rule applied strikethrough text to records containing the specific text 'DATA.xml' (see figure 3). Again, the records with the combined formatting were eliminated from consideration.



Edit Formatting Rule

Select a Rule Type:

- ► Format all cells based on their values
- ► Format only cells that contain
- ► Format only top or bottom ranked values
- ► Format only values that are above or below average
- ► Format only unique or duplicate values
- ► Use a formula to determine which cells to format

Edit the Rule Description:

Format only cells with:

Specific Text | containing | .pdf

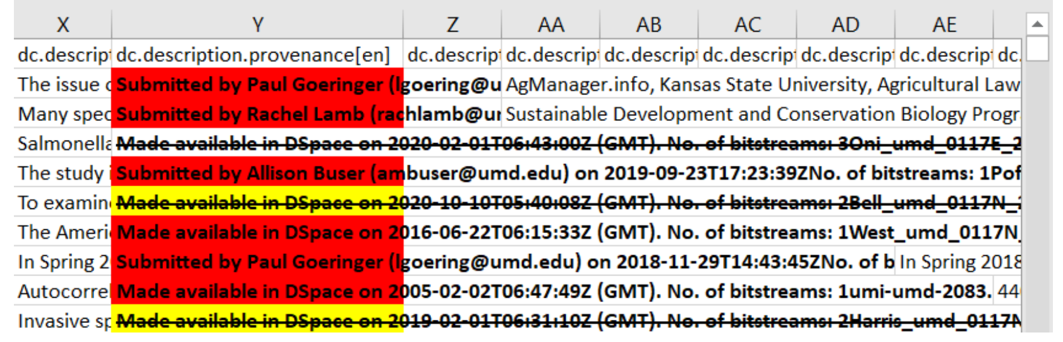
Preview: **AaBbCcYyZz**

Figure 2. Conditional formatting rule applying bold text to any record containing '.pdf' in the dc.description.provenance column.

6. Repeat and adapt rules for each collection. This process was applied to the metadata returned by queries of each collection in DRUM (with the exception of the Dissertations and Theses collection, as its records are all cross listed in other collections). Generally, these two sets of rules eliminated the majority of collection records containing the word 'data.' Some collection specific elimination rules were created as well. For example, School of Music ETDs that only contained audio files in addition to the usual PDF and XML files were removed.

7. Organize Remaining Records. Records marked for closer examination were added to a central list. A final conditional formatting rule was applied to highlight duplicate records mapped to multiple DRUM collections. Highlighted records were then removed and a final list of qualifying datasets was compiled for later manual analysis.

Results



X	Y	Z	AA	AB	AC	AD	AE
dc.descrip	dc.description.provenance[en]	dc.descrip	dc.descrip	dc.descrip	dc.descrip	dc.descrip	dc.descrip dc.
The issue c	**Submitted by Paul Goeringer (lgoering@u**		AgManager.info, Kansas State University, Agricultural Law				
Many spec	**Submitted by Rachel Lamb (rachlamb@ui**		Sustainable Development and Conservation Biology Progr				
Salmonella	**~~Made available in DSpace on 2020-02-01T06:43:00Z (GMT). No. of bitstreams: 3Oni_umd_0117E_2~~**						
The study	**Submitted by Allison Buser (ambuser@umd.edu) on 2019-09-23T17:23:39ZNo. of bitstreams: 1Pof**						
To examin	**~~Made available in DSpace on 2020-10-10T05:40:08Z (GMT). No. of bitstreams: 2Bell_umd_0117N_~~**						
The Ameri	**Made available in DSpace on 2016-06-22T06:15:33Z (GMT). No. of bitstreams: 1West_umd_0117N**						
In Spring 2	**Submitted by Paul Goeringer (lgoering@umd.edu) on 2018-11-29T14:43:45ZNo. of b**						In Spring 2018
Autocorre	**Made available in DSpace on 2005-02-02T06:47:49Z (GMT). No. of bitstreams: 1umi-umd-2083.**						44
Invasive sp	**~~Made available in DSpace on 2019-02-01T06:31:10Z (GMT). No. of bitstreams: 2Harris_umd_0117N~~**						

Figure 3. Excel file with conditional formatting applied to the dc.description.provenance.column.

From the approximately 17,900 records that are returned by a general search for the keyword 'data' in DRUM, this method marked 258 for manual examination. Unsurprisingly, the largest number of records marked for manual examination came from the College of Computer, Mathematical & Natural Sciences collection, followed by Engineering. This corresponds with trends displayed by already known dataset records in DRUM. There were several advantages and disadvantages to this method. The term "data" is broad and used with the expectation of capturing more potential dataset records in a search. However, it also returned a considerable number of non-qualifying records, making elimination time consuming. Larger batches could take 30-40 minutes to complete and left more room for human error. This method also excludes potential records that do not contain the term "data".

The greatest advantage of this method was the ability to adapt and improvise rules of elimination according to variations in the metadata. Working directly with the complete metadata files also provided a greater sense of how dataset and non-dataset records can be expected to appear in DRUM overall.

Approach 2: APIs & R Studio

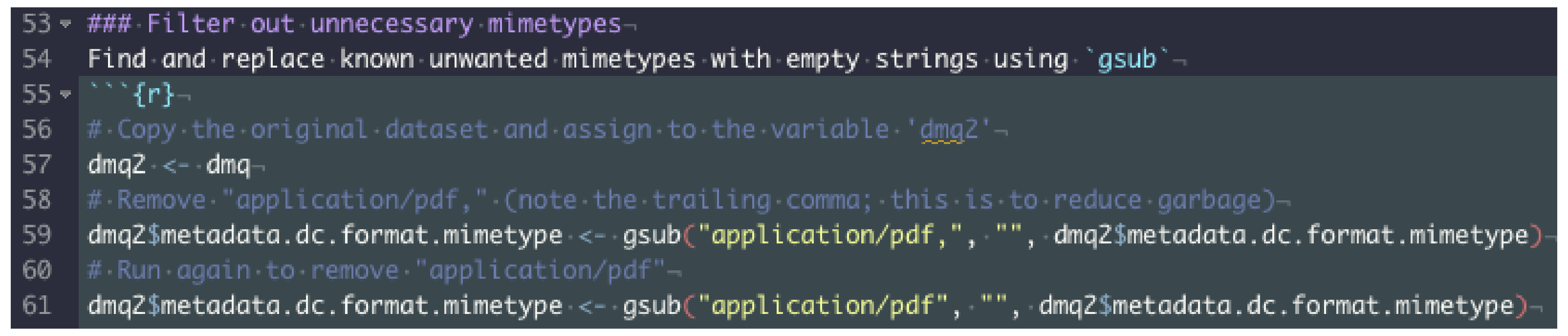
An alternative to using DRUM's Admin Tools to query metadata records is the OAI-PMH Solr core. While the OAI-PMH API endpoint can be queried, it was simpler to obtain selected metadata elements for all records in DRUM by querying the OAI-PMH Solr core directly. This approach utilized exploratory techniques to iteratively visualize and search through approximately 25,000 metadata records in DRUM.

Method

1. Query the Solr OAI core. This Solr core contains the necessary Dublin Core element dc.format.mimetype which is necessary to assess what type of files are associated with a particular DRUM record. Only four elements were needed: item.id, metadata.dc.identifier.uri, metadata.dc.format.mimetype, and metadata.dc.type. These four elements were set using the field limit (fl) in Solr. The total number of metadata records is known, and at the time that these metadata were harvested, there were 24,590 records. The results were then stored as a CSV. All of these parameters were URL-encoded and the query was posted to the Solr API.

2. Store the output as a data frame in R. Data frames are two-dimensional array-like structures that are used to organize data into columns and rows while storing additional attributes about the data. Storing the data as a data frame allows for using many of the built-in features of R. Once the query output was stored in a data frame, the data could be summarized, visualized, and edited. A preview of the data reveals that records have multiple mimetypes, or media types, which indicate the format and contents of the associated files. Using the metadata.dc.format.mimetype element, unnecessary mimetypes can be filtered out of the results set to narrow the number of records that may contain research datasets.

3. Filter unnecessary mimetypes. A list of unique mimetype values was created so that filtering can be applied. Using this list and the gsub() function, which replaces all instances of value, known non-data formats were removed from each record and replaced with an empty string value (see figure 4). Examples mimetypes to be removed include, application/pdf, application/msword, and application/vnd.ms-powerpoint. Once all of the unnecessary mimetypes were removed, there were multiple records that now have no associated mimetype. This indicates that the record only contained non-data mimetypes and can be removed from the results set.

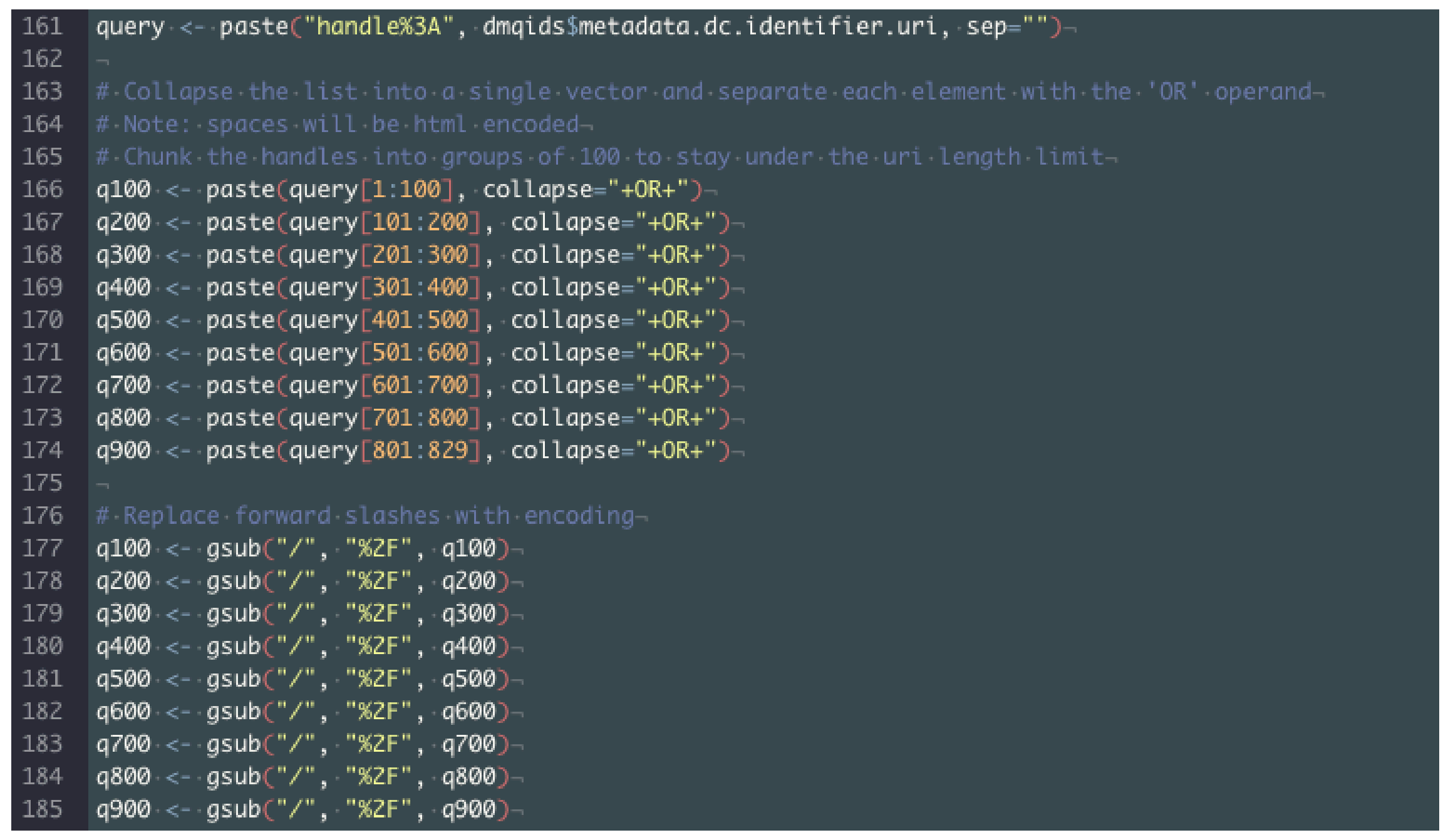


~~~
### Filter out unnecessary mimetypes
Find and replace known unwanted mimetypes with empty strings using `gsub`
```{r}
# Copy the original dataset and assign to the variable 'dmq2'
dmq2 <- dmq
# Remove "application/pdf," (note the trailing comma; this is to reduce garbage)
dmq2$metadata.dc.format.mimetype <- gsub("application/pdf,", "", dmq2$metadata.dc.format.mimetype)
# Run again to remove "application/pdf"
dmq2$metadata.dc.format.mimetype <- gsub("application/pdf", "", dmq2$metadata.dc.format.mimetype)
~~~

Figure 4. Filtering unnecessary mimetypes using the gsub() function.

4. Remove musical recordings. A frequency table was then created using the dc.type element; this lists each object type and counts the number of times it occurs in the results set. The majority of types returned from the original Solr query are primarily articles, theses, and dissertations. Supporting datasets added to a primary research product such as an article or dissertation is common and these types cannot be excluded from the analysis. Musical records, however, are unlikely to include research data and can be removed from the results set. There were 35 musical recordings of three different types removed from the results set

5. Create a subset of handles and query a different Solr core to obtain filenames and file extensions. A subset of 829 records has been created to filter out obvious records without research data, the handles for each record may be extracted and used to perform another query on the drum-search Solr core. The full uri (https://hdl.handle.net/1903/xxxx) is not needed for the query and can be edited using gsub() to create a new subset of just the record ids in the format 1903/xxxx. Once these ids are stored in a new data frame, the Solr query can be created. This query limits the returned fields to include handle and stream_source_info, which contains detailed information about the related bitstreams or files. The new query will be broken up into 100 record chunks in order to not overwhelm the Solr core (see figure 5). Once all of the queries are posted and results are returned as CSV files, the next step is to combine these into a single data frame. This new results set is then merged with the previous data frame to create a new data frame that contains type, mimetype, and stream_source_info elements..



```
query <- paste("handle%3A", dmqids$metadata.dc.identifier.uri, sep="")

# Collapse the list into a single vector and separate each element with the 'OR' operand
# Note: spaces will be html encoded
# Chunk the handles into groups of 100 to stay under the uri length limit
q100 <- paste(query[1:100], collapse="+OR+")
q200 <- paste(query[101:200], collapse="+OR+")
q300 <- paste(query[201:300], collapse="+OR+")
q400 <- paste(query[301:400], collapse="+OR+")
q500 <- paste(query[401:500], collapse="+OR+")
q600 <- paste(query[501:600], collapse="+OR+")
q700 <- paste(query[601:700], collapse="+OR+")
q800 <- paste(query[701:800], collapse="+OR+")
q900 <- paste(query[801:829], collapse="+OR+")

# Replace forward slashes with encoding
q100 <- gsub("/", "%2F", q100)
q200 <- gsub("/", "%2F", q200)
q300 <- gsub("/", "%2F", q300)
q400 <- gsub("/", "%2F", q400)
q500 <- gsub("/", "%2F", q500)
q600 <- gsub("/", "%2F", q600)
q700 <- gsub("/", "%2F", q700)
q800 <- gsub("/", "%2F", q800)
q900 <- gsub("/", "%2F", q900)
```

Figure 5. Creating URL-encoded Solr queries in 100 row batches.

6. Visualize the data and explore the records using text searching. A cursory examination of the data indicates that there are several records that are missing both type and stream_source_info. Visualizing these records shows that a substantial amount are datasets and images (figure 6). A visualization of those records without values in the stream_source_info column indicates that the majority of these records are classified as articles and dissertations (figure 7).

The second visualization supports the notion that there are potentially qualifying research data files associated with traditionally text-based documents, like articles and dissertations. Records are then separated into two sets depending on whether there are multiple files associated with the record. Then using the logical grep function in R, grepl(), file extensions can be filtered to exclude known non-known data formats such as PDF, DOC, DOCX, PPT, and PPTX. The grepl() function returns TRUE if the provided patterns match and all records returning TRUE can then be stored in a separate data frame; the same can be done for records that return FALSE. An examination of the contents of records with only a single associated file are all confirmed to not contain possible data formats. This means that the only records worth looking at are those that contain multiple files.

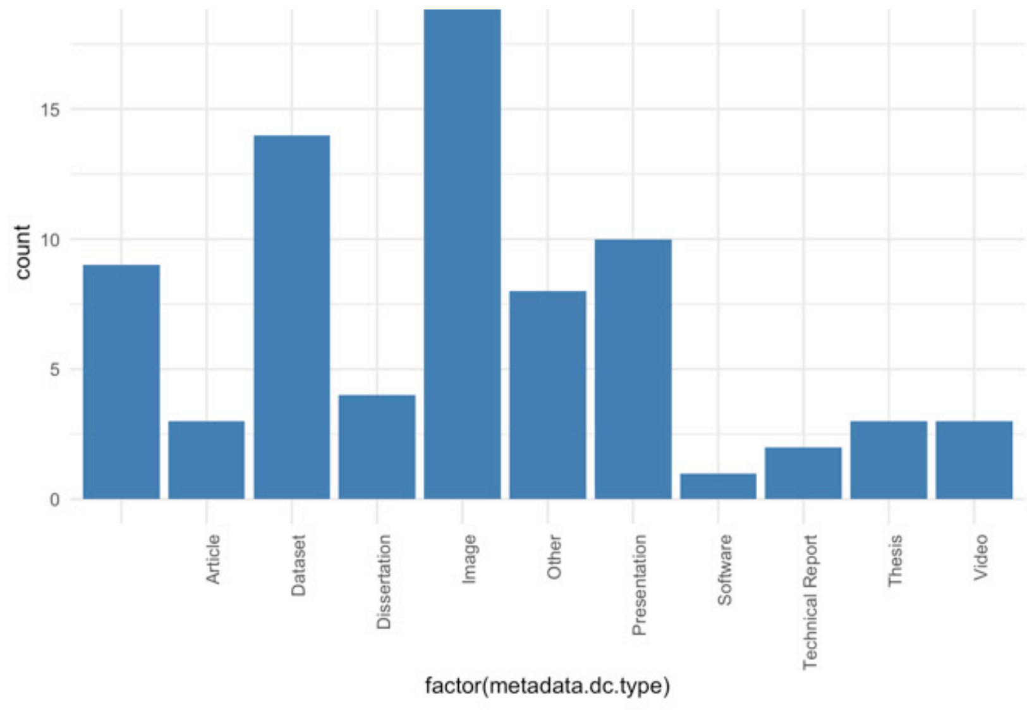


Figure 6. Visualization of records that are missing stream_source_info values.

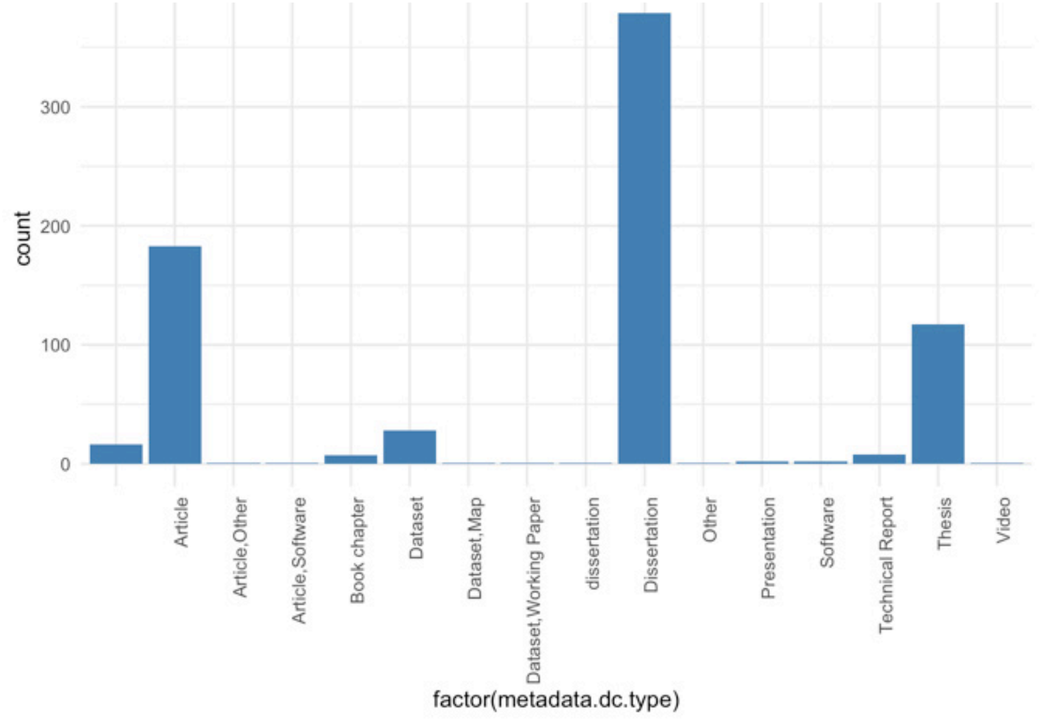


Figure 7. Visualization of records with stream_source_info

Using only those records with multiple files and iterating through the results creates a set of unique combinations of multiple file extensions. From this, there are only six file extensions across 504 records that are not text documents: CSV, XLS, XLSX, ALN, HTM, and TXT. Of these, 439 are some version of an Excel spreadsheet.

Results

A cursory review of the affected records indicates that the majority of these Excel files have been added as supporting research documentation to various dissertations and articles. While the original hypothesis of "hidden data" has been confirmed, it is unclear if these data files truly qualify as machine-readable research datasets given that many of these are mixed-use spreadsheets containing a variety of graphs, tables, statistical calculations, and bibliographic data.

Conclusion

Once the authors completed their queries and analyses, it was discovered that the individual results from each approach were not identical. The results created by querying the Solr cores were larger than those created by querying through DSpace Admin Tools. The total number of results across both sets was almost a thousand records that would have to be manually inspected. For the initial manual review of hidden datasets in DRUM, the authors found 127 records common to both results sets. This review is ongoing, but initial results have confirmed the presence of legitimate machine-readable datasets; however, the majority of the results are mixed-use spreadsheets that do not qualify.

About the Authors

David Durden is the Data Services Librarian at the University of Maryland Libraries. He has an MLIS in Archives and Digital Curation from the University of Maryland iSchool and an MA in Musicology from Brandeis University.

Allison Buser is a Graduate Assistant in Digital Progams & Initiatives at the University of Maryland Libraries. She is a graduate student in the UMD iSchool in the dual masters program in History, and Library and Information Science (HILS), and has a BA in History and a certificate in Museum Studies from the University of Iowa.