An Exercise in Iteration: Transcribing the 1950 United States Census with DataScribe

This case study explores the process of creating projects and datasets to transcribe returns from the 1950 United States census. The final structure of the DataScribe projects were determined only after a great deal of iteration and experimentation.

Table of Contents

The Census Returns

The United States National Archives and Records Administration (NARA) released the 1950 census record at roughly the same that the Roy Rosenzweig Center for History and New Media (RRCHNM) released DataScribe. I was looking for a test dataset which was both interesting and public domain images: The census was a perfect fit.

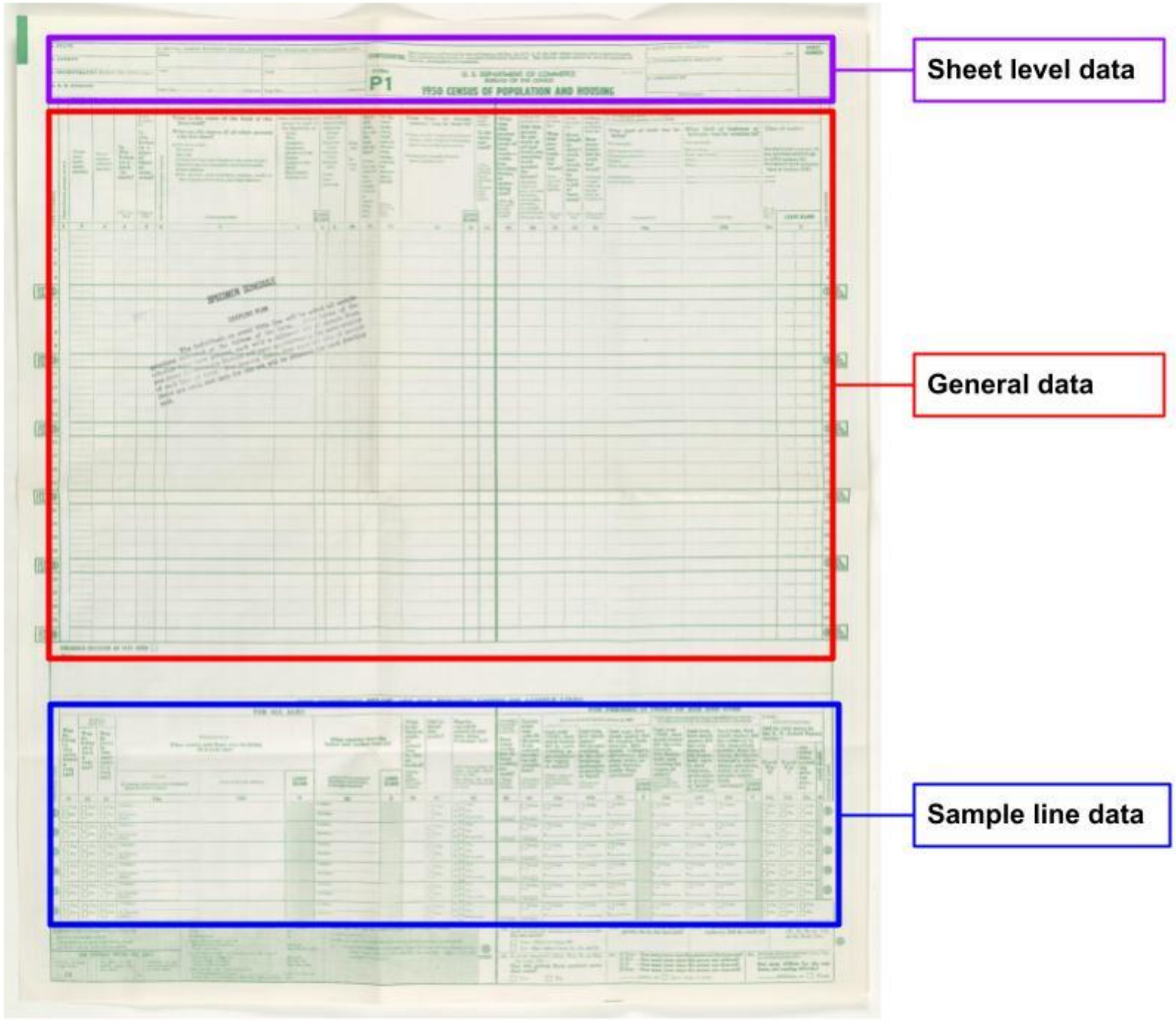
The 1950 Census offers a large corpus. Not only are there the general population returns - the ones which most people will see when they go searching for a family member or their current address - but there are also returns for territories and Americans abroad. The 1950 Census Website from NARA provides completed general returns (information about individuals) and also offers digitized, full-color, blank versions of all of the forms <https://1950census.archives.gov/howto/blank-forms.html>.

In order to work with the Census returns, it is necessary to understand something of their structure. The material published by NARA on the 1950 Census website includes only the general population returns - that is, the forms listing the names and demographic data of individuals. The data is organized first by state or territory, then by locality (usually a city or county) and finally by enumeration district. On the FAQ of their website, NARA explains that an enumeration district: “was a geographic area that a single enumerator (census taker) could complete within two weeks in cities or within 30 days in rural areas. Each ED has a two-part number, such as 10-15. The first number (prefix) is the number of the county (or county equivalent) and the second number was a specific geographic area within that county.”[1]

There are a few different forms used in the data which is available from NARA. Returns for the continental United States - the then 48 states plus the District of Columbia - all used the P1 form. This form has three sections. At the top was a header area with sheet-level data including the location information, date, sheet number, and a space for the name of the census taker. The second section was individual data for all of the residents in the enumeration district, one person per line. Six lines per sheet were designated as sample lines. For each individual on the sample line, the census taker was supposed to complete additional demographic questions. Most

[1] https://1950census.archives.gov/howto/faq.html#ed

returns using the P1 form have pages starting at 1 and a series of pages starting at 71. The pages in the 70s were used for individuals who were not at home when the census taker first went through the neighborhood.[2]



P1 Form with various sections

The overseas territories each had a different form. The demographic information captured was a combination of the general and sample line questions from the P1 form. The overseas forms

[2] https://1950census.archives.gov/howto/faq.html#missing

omitted the sample line section completely. The main variation between the forms for each overseas territory seems to have been the predefined racial categories in the Race question and the default options for where someone was born, which were adapted for each specific location.

Intellectual Objectives

Understanding the structure of the data in the original sources is just one aspect of planning work with DataScribe. In order to set up an effective structure of projects, datasets, and data forms, I needed to know what the end goals were for the transcribed data.

First and foremost, there was never any intention of a comprehensive transcription of all of the census documents from all states, territories, and Indian reservations. This study was only ever meant to be exploratory. The primary objective was to create a useful model for working with DataScribe by conducting samples from a larger dataset, resulting in datasets with which researchers could experiment to create analyses or visualizations.

I initially planned two uses for the transcribed data, at least from the continental United States. First, to compare manual transcription to the automated transcription that powers the search engine on the NARA website. These transcriptions are available as part of the data download from NARA. Secondly, I thought it would be interesting to be able to create analyses which went from the macro level of national or state demographics to the micro level of street or even individual data.

After transcribing a few sheets, another avenue for analysis presented itself: people who weren’t at home. The fact that there are the additional pages (those numbered 71 and higher) for people who weren’t at home means that it’s possible to find out some demographic data for people who were absent from their houses on the first (and sometimes second) day that the enumerator came through their neighborhood. Combining this with information about the date - what day of the week were they away from home - could be the basis for some very interesting analysis about lifestyle patterns.

For the overseas returns, I was interested in the possibility of analyzing families and family groups of American citizens living overseas. Although the individual returns for Americans abroad (meaning foreign service officers stationed in sovereign nations), the territories with military bases offer an opportunity to look at who resided on- or off-base and what kinds of family groups ended up moving overseas.

Preparatory work

Because these were meant to be small samples, rather than comprehensive transcriptions, I used a more labor-intensive workflow to generate items in Omeka S which could be used in DataScribe. The process included downloading individual images from 1950census.org, renaming those files, creating a spreadsheet or Tropy project to generate metadata, uploading files to a directory on my server for use with the File Sideload, creating item sets in Omeka S, and finally using the CSV import module to create the items and put them in the respective item sets.

Getting the files

Downloading the image files was in some ways the most time-consuming process. The 1950 Census website did not offer a way to download an entire enumeration district's images at once. As a result, I had to manually download each file from an enumeration district by clicking to the page, clicking download, selecting the largest resolution (to enable better clarity when zooming in), and changing the file name on download. The website does not generate unique file names for each image, so I gave them unique names as I downloaded the file.

In order to get some geographical diversity and increase the number of datasets with limited time, I asked other members of the DataScribe team to download image files for one enumeration district from a place in the continental United States which had relevance to them personally or to their scholarship. Two of my colleagues obliged by providing me with images from Minneapolis, Minnesota, and Syracuse, New York.

The necessity of downloading each image individually means that I and my colleagues tended to choose smaller enumeration districts – those with 10-30 pages – rather than the larger districts which run to over fifty pages.

Organization and metadata

Once the files were downloaded, I created the metadata for the items. This was important to me even in an experimental data project because I wanted anyone to ensure proper credit was given to the source of the images. Adding robust metadata also meant that it would be easier to use a search plus bulk edit function in Omeka S if I wanted to create new datasets (see below for more on this).

Initially I simply created a spreadsheet with columns for the metadata properties I wanted to capture. Most of these were consistent for an entire sheet and some for all of census items.

- Title: Formatted as State-Locality-ED-page#, ed “Virginia-Fairfax-30-62_09”
- Creator: “Bureau of the Census”
- Rights: “Public Domain”
- Description: All items contained “This series contains the 1950 census which attempted to enumerate every person living in the United States on April 1, 1950, although some persons were missed. The enumeration began on April 1, 1950, and was completed within four weeks.”
    - For later items, I added a multi-value separator and then inserted the enumeration district description.
- Identifier: “”43290879” (the NARA identifier)
- Publisher: National Archives and Records Administration
- State: written out, ex “Virginia”
- Locality: state or city, ex “Fairfax”
- Enumeration District: two digit, ex “30-62”
- Source: “https://1950census.archives.gov”
- File: filename and filepath formatted to work with the File Sideload module.

These columns corresponded to properties in a resource template I created for census documents. For the most part the headings correspond directly to Dublin Core properties. For those that do not, the mappings were:

- state = spatial coverage;
- locality = provenance; and
- enumeration destruction = Is Part Of.

I used the File Sideload <https://omeka.org/s/modules/FileSideload> module to allow me to import the media files at the same time that I created the items with CSV import.

Initially I made item sets for each locality, combining different enumeration districts into the same item set on import. However, as the project progressed and I got a better sense of possible workflows, I broke out some of the enumeration districts into their own item sets. I did this by creating a set for the specific enumeration district, running an advanced item search where “Is Part Of” contained the enumeration district number, then bulk-editing those items to add them to the new item set.

Creating projects, datasets, and forms

The bulk of the intellectual work on this project took place in the creation and revision of datasets, forms, and projects. For the most part, transcription was only taking place as part of DataScribe workshops.[3] My focus was thus on ensuring that the structure and workflows were good examples of what DataScribe can do, and on creating datasets, forms, and guidelines which were relatively user-friendly without a lot of time to train transcribers.

Initial organization

When I first started, I organized the projects according to the (anticipated) intellectual goals of the overall project and the datasets by the structure of the historical form data. In practice, this meant that I created two separate DataScribe projects: one for the continental United States and one for the overseas territories. Within the projects, I created datasets by state or territory. My reasoning on leaving the datasets at such a broad geographic level was that I was working with sample-sized sets, not comprehensive, so we would not end up with datasets with thousands of items.

I decided to split the information on each census P1 form (the continental US) into two separate datasets/forms: one for the general returns and one for the sample lines. I wanted to be able to analyze the demographics of the sampled population separately.

[3] See the Resources section at the end for materials from the workshops.

The early structure of the project looked something like this:

- DataScribe Project: Continental US
    - Dataset: Virginia General Returns
    - Dataset: Virginia Sample Lines
    - Dataset: Washington DC General Returns
    - Dataset: Washington DC Sample Lines
- DataScribe Project: Overseas US
    - Dataset: Panama Canal Zone
    - Dataset: Guam

One of my reasons for making two datasets for the P1 forms was noticing that children and infants were not infrequently on the sample lines and I wondered how often that happened. I set up the Sample Lines form to repeat some of the information from the General form so that 1) there was enough information to run analysis on the dataset alone and 2) there were multiple datapoints which could be used to merge the Sample Lines and General datasets after export.

In the earliest iteration of the form, I omitted some of the information from the header (questions e, g, and h). I used the same question order as the original form but not always the same wording, and only used select options for a handful of the questions. My guidelines were fairly general, focusing mostly on “do your best with the handwriting” and clarifying not to use “is missing” unless there was physical damage to the form.

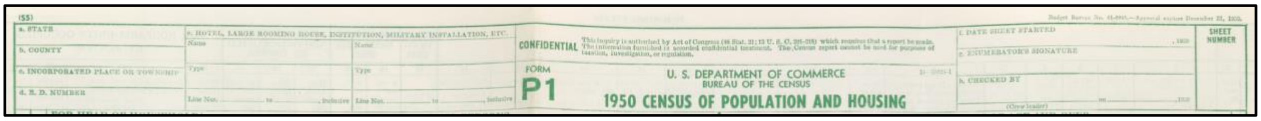
Current organization

Over a series of months, I tested my transcription forms, invited colleagues on DataScribe to contribute forms and test the transcription, and used the projects in multiple workshops with transcribers who were encountering the 1950 Census and DataScribe for the first time. The projects, datasets, and forms went through many iterations as a result. Each encounter with the project brought up new questions which required me to either explain the reasoning behind a choice or adjust the structure of a piece of the project.

There are now multiple DataScribe projects which form part of the overall 1950 Census project. I am able to do this in part because the colleagues with whom I share a testing install are agreeable to all of us having multiple projects. The different projects have helped me break out the datasets according to the needs of the project’s ultimate intellectual objectives *and* the needs of transcribers and reviewer workflows.

One of the first things that we noticed when doing a group transcription on some P1 forms in a DataScribe workshop were the number of fields in the sheet metadata which were repeated for every row in the item. This included information like the state, locality, and enumeration district. Workshop participants were confused about whether they needed to re-enter this information for every record, and remarked that doing so was tedious. In response, I have created three separate projects: a “sheet data” project, one project for Virginia, and one for Washington, DC.

The “sheet data” project has datasets by locality, with multiple enumeration districts per dataset, but there is only one record per item. The form for these datasets transcribes all of the information that is sheet-level - that is, the items labeled with letters on the census forms. Breaking the sheet data into a separate dataset allows it to be transcribed quickly and then merged with the returns data after export, preserving the transcribers’ time and effort.



(55)

Budget Bureau No. 41-0901.—Approval expires December 31, 1950.

a. STATE

b. COUNTY

c. INCORPORATED PLACE OR TOWNSHIP

d. E. D. NUMBER

e. HOTEL, LARGE ROOMING HOUSE, INSTITUTION, MILITARY INSTALLATION, ETC.

Name Name

Type Type

Line No. .... to ...., inclusive Line No. .... to ...., inclusive

CONFIDENTIAL This inquiry is authorized by Act of Congress (46 Stat. 21; 13 U. S. C. 201-218) which requires that a report be made. The information furnished is accorded confidential treatment. The Census report cannot be used for purposes of taxation, investigation, or regulation.

FORM P1

U. S. DEPARTMENT OF COMMERCE
BUREAU OF THE CENSUS
1950 CENSUS OF POPULATION AND HOUSING

f. DATE SHEET STARTED ...., 1950

g. ENUMERATOR'S SIGNATURE

h. CHECKED BY .... on ...., 1950
(Crew leader)

SHEET NUMBER

Header information of the P1 form

The Virginia and Washington DC projects serve to illustrate some of these issues and model possible paths forward for anyone else wanting to work with the 1950 Census returns. The Virginia project has datasets by locality, with enumeration districts combined. The form for the Virginia returns includes some of the sheet-level data. The Washington DC datasets are broken down by enumeration district and their forms only require sheet number and date. The juxtaposition acts as an object lesson in making choices about workflow and data, and thinking about what work can be done after export from DataScribe as well as in the forms.

I had to rebuild the overseas territory item sets because I did not initially realize the importance of keeping the A and B sides of a sheet together. The districts sampled from the Panama Canal Zone and Guam are fairly small - only 4-6 sheets per district - and so I have kept them as larger datasets grouped by locality. These datasets have a limited number of fields from the sheet level data, collecting only the enumeration district number and the page number and side.

The current structure of projects and datasets is roughly along these lines:

- 1950 Census Sheet Data (project)
    - NY Onondaga sheet data
    - VA Fairfax sheet data
- 1950 Continental US (project)
    - VA Fairfax General returns
    - VA Fairfax sample lines
- 1950 Census Washington DC
    - Washington DC 1-174
    - Washington DC 1-173
- 1950 Census Overseas
    - Guam - Sumay
    - Panama Canal Zone - Balboa
    - Panama Canal Zone - Cristobal

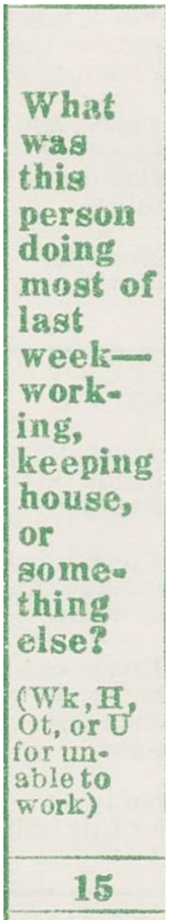
Creating and revising the forms

My initial attempt to create a DataScribe form based on the P1 form raised a number of questions. Did I want to copy the text of the questions exactly or summarize? For the yes or no questions did I want to use a checkbox or a radio button? Which, if any, of the standardized answers should I make select dropdowns for? How did I want to deal with "other" entries for these questions?

One of the first changes I made in revising my initial form was to add the numbers and letters used on the original census forms to my DataScribe form labels. Even as the creator of the form, I found it easy to lose my place in the middle of transcribing a row. While not every field in the DataScribe form has a number -- some questions on the original took more than one DataScribe field to cover -- it has proven very helpful to have a clear and quick point of reference between the original document and the form. Adding the question numbers also allowed me to shorten some of the question text to be more concise and gender-neutral, for example going from "How old was he on his last birthday" to "How old on last birthday?" (question 11 on the P1 form).

Dealing with the apparent binaries in the form also posed a challenge: did I want to use a checkbox or a radio button? Checkboxes would, by default, export as binary (1 or 0), while a radio button would export as “yes” or “no”. I decided to use the checkboxes for fields where it only mattered if there was a yes, for example the questions about military service. I used radio buttons for those instances where a “no” was informational and not simply the absence of a “yes,” for example in questions about employment and job-seeking.

Standardized answers are my term for those abbreviations provided by the Census Bureau either in the question text or in a footer on the form. Some of these are fairly legible to the novice transcriber, such as “M” and “F” for male and female. Others can be a bit difficult to decipher, especially if you have not looked at the suggested answers. For example, the image at right shows question 15 on the P1 Form, which asks what the person was doing most of the previous week. The answers a transcriber would find in this fairly narrow column are “Wk” (working), “H” (keeping house), “Ot” (other), and “U” (unable to work). The variable handwriting of the census enumerators, however, means that an “H” could easily resemble a “N” or “M”, and “Wk” might look more like “Wt”. In these instances, I determined that having a select dropdown of the Census Bureau approved answers would prevent confusion on the part of transcribers and ensure cleaner data in an export. This proved particularly helpful with the education questions in the sample lines. Firstly because those abbreviations are printed below the returns table rather than in the body of the question. In addition, the letters used in these abbreviations include C and S combined with numbers and, depending on the handwriting, read as 55 rather than S5, for example.

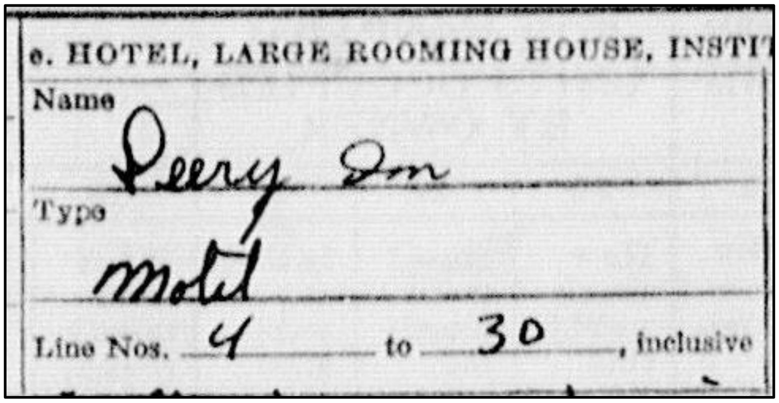


What was this person doing most of last week—working, keeping house, or something else?

(Wk, H, Ot, or U for unable to work)

15

As I worked through creating and transcribing multiple datasets from multiple enumeration districts, I found that I needed to include information on the original forms which I had omitted. One of the fields in the sheet-level data section was “E. Hotel, large rooming house, institution, military installation, etc.” The first few enumeration districts I worked with -- in Fairfax, Virginia, and Washington, DC -- had no information in that field and so I did not find it helpful to include. However, when I started working with enumeration district data from the Panama Canal Zone and Guam -- specifically sheets including military barracks -- I realized what that field was for and why it might be useful. I found at least one instance of Field E in use in Virginia, in the case of persons enumerated who were living or at least staying at a motel in Alexandria. The important thing to capture for Field E is not only the institution named but the lines which it covers, since it may be only a few lines.



e. HOTEL, LARGE ROOMING HOUSE, INSTI

Name
Peery Inn

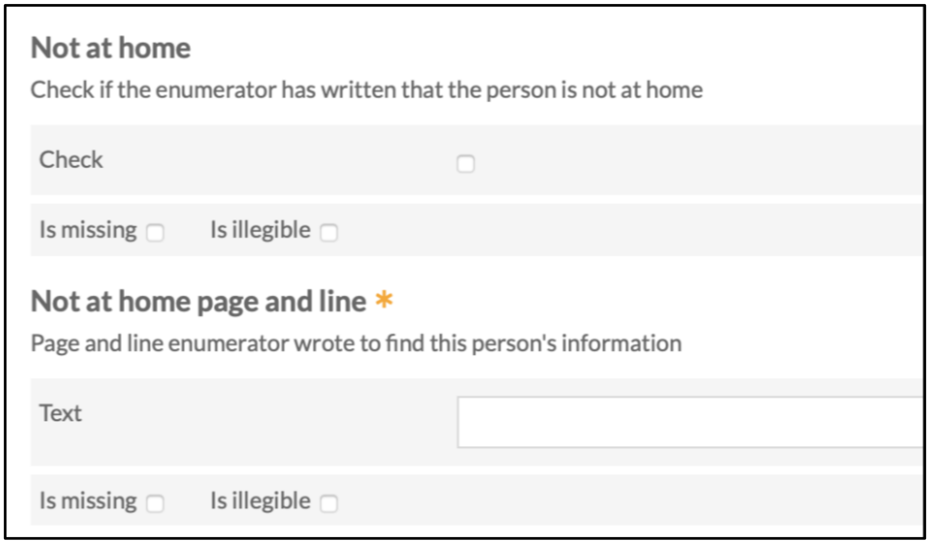
Type
motel

Line Nos. 4 to 30, inclusive

Field E for Fairfax Virginia ED 30-46, sheet 34

I decided it was important to capture Field E information, to some extent, in the general returns form. I created a checkbox on the DataScribe form labeled “Line falls in range covered by box e”. At the moment, I have the Field E text information as a field on the form as well, however that could be limited to the sheet data form and rejoined programmatically after export. I prefer to keep the information with the general returns record so that the institution information remains attached to the record and is readily available for analysis without any sort of remediation.

Another piece of information which I did not originally capture on the DataScribe form is the lines for people who are not at home. Those people are, in theory, counted on pages 71 and higher for each enumeration district. After the first workshop, however, I started to think about how to capture the information of when these lines show up, in order to facilitate cross-referencing, particularly when different members of a single household are split over multiple sheets. The enumerator usually wrote “Not at home” and often added in later the sheet and line numbers for the missing persons. My current solution is to add two fields just before the field for question 6 -- name of individual -- since that is the space where “not at home” was most often recorded.



Not at home

Check if the enumerator has written that the person is not at home

Check □

Is missing □ Is illegible □

Not at home page and line *

Page and line enumerator wrote to find this person's information

Text

Is missing □ Is illegible □

Not at home checkbox and text field

The first field is a checkbox for “not at home” which will allow for a quick quantitative analysis of how many people were absent. The second field is a short text entry for the transcriber to enter whatever the enumerator wrote in the line. In theory, this will help the process of matching people on later pages with their households.

The process of refining the forms for my datasets, both adding in missing fields and removing some in order to simplify the workflow, was easier thanks to the fact that you can export DataScribe forms as JSON files. The structure of the exported form is fairly straightforward if one is familiar with the syntax of other coding languages. This code snippet is the JSON version of the checkbox and text field:

```
{
       "data_type":"checkbox",
       "name": "Not at home",
       "description": "Check if the enumerator has written that the person is
not at home",
       "is_primary": false,
       "is_required": false,
       "data": {
           "checked_value": "Not at home",
           "unchecked_value": "At home",
           "checked_by_default": "0",
           "label": null
```

```
        }
    },
    {
        "data_type": "text",
        "name": "Not at home page and line",
        "description": "Page and line enumerator wrote to find this person's
information",
        "is_primary": false,
        "is_required": false,
        "data": {
            "minlength": null,
            "maxlength": null,
            "placeholder": null,
            "pattern": "",
            "default_value": null,
            "label": null,
            "datalist": []
        }
    },
```

The labels for the various options clearly correspond to the options when building the form in DataScribe itself - checked by default, minimum and maximum length. For some fields -- particularly the variations in the standardized answers -- I found it easier to duplicate and edit a JSON export of the form rather than make the changes in DataScribe itself. I now have a master JSON file with every single field I have come up with for the P1 form, which I can then duplicate and edit to set up various datasets based on that form.[4] I try to ensure that I have triple-checked for any errors in the JSON file before uploading the form because DataScribe does not throw an error on dataset creation if you upload a file with errors, it simply does not create the form.

Even as I refined the forms for each dataset, breaking out individual enumeration districts to cut down on repetition for transcribers, I kept a few fields which retained sheet-level information. I wanted there to be enough information that an exported file of the data with a bad filename would still have enough information to trace it back to the source.

[4] I have published this json file in the GitHub repository mebrett/datascribe1950cs

Guidelines

Although I was the only regular transcriber of these projects, they were designed to be used in workshops with people encountering DataScribe for the first time. As such, the guidelines needed to be robust, both to help transcribers unfamiliar with the Census documents and to model good guideline practice.

As I wrote the guidelines, I took advantage of the fact that DataScribe lets you format this text with headings, lists, and even links. I tried to create a usable structure to the guidelines so that transcribers would be able to find answers quickly. It is currently structured with four sections: general information; how to deal with blank fields; a bulleted list of how to find specific pieces of information on the original document; a section specifically explaining Field E. Two of these three sections have headers to help with navigation.

The opening section states what part of the sheet is being transcribed (either the main section or the sample lines). It also explains when to use “is missing” and “is illegible” - these flags are built in to DataScribe but the application of them varies by project. For this project, transcribers should only use the “is missing” when a required field or the enumeration district field are empty. The guidelines encourage transcribers to leave a note for reviewers explaining which field or fields (identified by number) contain illegible text.

The section with bullet points on where to find different sections of the sheet is meant to serve as a reminder for transcribers, supplemental to in-person (or video based) training that involves a slower review for the P1 form. One of the main points for this section is to remind them to use the rotation and zoom features in order to find the street address for each line, as the street names are written perpendicular to the rest of the data.

In the section of the guidelines for Field E, I start with an explanation of the field’s purpose and how it might be used. Since DataScribe allows for some HTML in the guidelines, I added a link to the sheet with the Peery Motel so that transcribers could reference an instance where there is data in Field E.

The section discussing blank fields grew out of transcriber questions. For instance, on most of the forms, the enumerator only wrote the address for the head of household. Subsequent lines for other members of the household are left empty. A few transcribers asked how to deal with these empty fields - should they be left blank? Because this information is important and could be easily lost from the data, the guidelines were updated to clarify that address information should be repeated for subsequent rows.

A different transcriber question stemmed from the above but added in the complication of households spanning multiple sheets. When the members of a household extended on to the next sheet, enumerators did not necessarily repeat the address information or even the last name on the new sheet. This presents a challenge for users with transcriber permission levels if the previous sheet has been assigned to a different user; they cannot easily access the previous sheet *in DataScribe* in order to see the information.

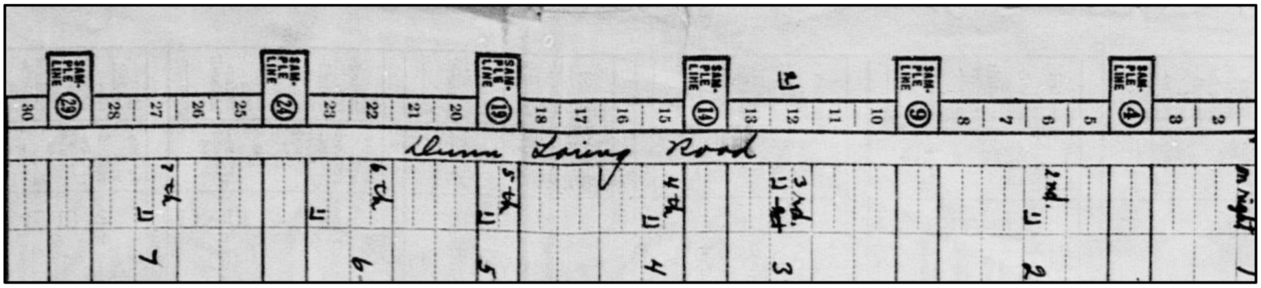
There were two possible solutions. The first was to document the process of going to the previous item, then clicking the Omeka item link in the item metadata sidebar, and from there viewing the media attached to the item. This method involves leaving the DataScribe work area, albeit only briefly, and would need to be clearly explained *and demonstrated* during transcriber training. The second option was to tell transcribers that empty address fields on the first few lines should be flagged as “is missing”, so that reviewers would know to fill in this information. Reviewers have more permissions in DataScribe and could consult the previous sheet without having to lock or unlock it.

After a conversation with my colleagues, I decided to go with the second option. While the first might be more sensible in an ongoing project with a small team, the transcribers for this project are generally workshop participants with limited time to learn the platform and widely varying degrees of knowledge about Omeka S. It was therefore simpler to go with the option which allowed them to work within the DataScribe transcription form.

Conclusions

The 1950 Census returns are a useful, public domain data source for learning and teaching DataScribe. The forms pose more complicated questions than I initially anticipated but this makes the data even more useful as a teaching tool. A teacher or workshop leader can make deliberate choices about how to present the data - for example, leaving the sheet-level fields in for every row of the general return - in order to demonstrate some of the choices which scholars must make when transcribing structured data.

My main recommendation for anyone planning to use this in a teaching setting is to ensure that someone - yourself or your participants - is familiar with the area of the enumeration districts you use. The screenshot below is of a page in the Fairfax, Virginia, returns which we used in our workshops (as George Mason University is in Fairfax). One colleague was stumped as to the street name. For those of us who had been in northern Virginia for a while, it clearly reads “Dunn Loring Road”, but our familiarity with local place names allowed us to see the D where others might read a Ll or U.



Dunn Loring Road

Resources

I created a repository on GitHub with forms and sample guidelines for people who are interested in working with the 1950 Census, either in a teaching setting or to experiment with DataScribe < https://github.com/mebrett/datascribe1950cs >.

For the workshops, we used a form building worksheet and gave each groups one of the overseas territories forms to work through. The form building tutorial with that worksheet is available at the DataScribe site <https://datascribe.tech/resources/tutorials/buildform/ >.