On Datasharing

Matt Marx (mattmarx@bu.edu)

15 April 2020

1 Introduction

Researchers share data for various reasons. Some journals require that authors make available data and programs for purposes of replication. Foundations and government agencies may stipulate that grantees post datasets as a condition of their grant. In these cases, scholars share data because they have to. Motivated less by a desire to promote cumulative research and more by a need to satisfy a requirement, they may do the minimum to "check the box." Publishers have established requirements, such as those put forth by the American Economic Association's Data Editor, to uphold a minimum standard such that published papers can be replicated.

This document is instead aimed at those who *want* to share their data. Motivated by others taking on cumulative projects, they want to make the data easy to retrieve and work with. At the same time, they want to minimize the amount of time spent answering questions from users. The path of least resistance may seem to be simply post a datafile on one's own website, but this approach can be problematic if someone moves institutions. What if the data needs to be updated – how will previous versions (which some scholars may have used in their projects) be archived? What sort of documentation should be provided? (The discussion below presumes that the researcher will be posting data files and not setting up an API for retrieval.)[1]

1.1 Datafile preparation

If you have a single file not more than a few gigabytes in size, this step can be skipped. If your file or files are larger, you face the decision of how and whether to segment and/or compress your data. The right approach can make it easier for others to make use of the data and reduce your support load.

[1] Disclaimer: these recommendations are largely based on my experience with posting data regarding patent-to-paper citations at relianceonscience.org and not on a large-scale, systematic study.

1.1.1 Segmentation

If you have larger files to post, it may make sense to segment and link them. For example, the main file of the Microsoft Academic Graph is 60G, which can take a long time to download and is probably too large for most researchers to load on their laptops. I suggest keeping (uncompressed) datafiles under 16G, 8G if possible. One way to do so is to structure your files as a relational database, with unique identifiers that link various files together.

The USPTO distribution of patent data at patentsview.org is an excellent example of segmentation. An ID for each patent is used to link together multiple files, one per field. For example, instead of having a file of patent assignee, where the assignee is a space-consuming text field, the file is patent assigneeID, where assigneeID is a numerical identifier that can be resolved in a separate file linking assigneeID assignee name.

1.1.2 Formats

Many researchers work with data using programs like Stata, R, or SAS and may be inclined to post data in one of those formats. Proprietary formats often store data, especially text data, more efficiently than a plain-text file because redundant entries can be internally represented with pointers. Thus file sizes can be smaller, and users can use files directly instead of having to import (larger) plain-text files and convert into their program of choice.

The downside of using proprietary formats, of course, is that those using programs that do not support whatever format the data were posted in need to convert it. Pressure may mount to post the data in multiple proprietary formats. Posting plain text files solves this problem, although doing so will probably lead to (much) larger file sizes. This is especially true with datasets containing redundant text strings.

1.1.3 Compression

If your files are small enough you can avoid compression. This is ideal for two reasons. First, depending on the compression approach you take you may get support requests from users who have trouble uncompressing files. Second, in some repositories users can preview uncompressed data before downloading it whereas compressed files will be opaque.

If you choose to compress your files, you'll need to choose from the myriad available methods. Some are more effective for text-based data; others are optimized for binary data. Even for a single distribution, you might achieve the smallest overall package size by mixing and matching compression methods. However, use of nonstandard methods can generate confusion among users and support requests for you.

Although it did not result in the smallest file sizes across the board, I followed patentsview.org in using "standard" .zip compression (via the 'zip' command in unix, or in Windows: right click, Send to, Compressed (zipped) folder).[2]

1.2 Documentation

Documenting the data you have shared can spur diffusion/adoption and lessen the volume of support requests. If your data is associated with a published paper, it may be appealing to simply point users to the published paper. If your paper is paywalled, however, this creates friction, as does having the data description buried in the middle of a large document. If however your paper described the data in a separate online appendix, you may be able to point users to the online document or a locally-hosted copy. Otherwise, I recommend creating a dedicated document that describes the data.

Every variable in the datafile(s) should be described individually, with its **type** (i.e., numeric, string, boolean) and **purpose**. If there are just a few possible values for the variable, enumerate them. If there is a numerical range, describe the upper and lower boundaries. If the variable was derived from another source, provide a link to the original data.

Source code is the ultimate documentation for users who want to re-create your data or understand every step in its assembly. Providing a link to a GitHub repository lets users know exactly what they are getting. Providing source code may result in support requests if your code contains hardcoded pathnames or is otherwise not well-architected.

1.3 Licensing

When making your data available, you may choose to (or need to) place conditions on their use. If your data depend on other data provided under a particular license, you may be required to pass-through those terms. You may also place your own terms on the distribution. For example, you may want to disallow commercial entities from using your data for profit.

In terms of license pass-throughs, as an example, the data at relianceonscience.org are derived from publicly available patent data, PubMed (also public), and the Microsoft Academic Graph. The latter source is under the Open Database Commons Attribution license (ODC-By), which allows anyone to use the data for any purpose as long as they acknowledge the original source by citing the relevant paper(s). Thus users of my data must cite the Microsoft Academic Graph as well as my own paper.

1.4 Choosing a Repository

Of course, the points above presume that you have somewhere to put your data. One possibility is to host the data on your own website, whether hosted by your institution or by a separate entity (i.e., wix or squarespace). The former can be a problem if you change institutions and the latter if you change providers; neither guarantees that the data will always be available.

[2] I have had some Mac users say they could not uncompress such files, but not many.

Another approach is to find a *persistent* repository where your data will remain available. There are many such repositories; Harvard's installation of Dataverse is probably the best-known. Others include Inter-university Consortium for Political and Social Research (ICPSR) and Zenodo (operated by CERN). A full inventory and comparison of all available repositories is beyond the scope of this document; below, I'll discuss factors to consider and use these few repositories to illustrate.

Of course, capacity of the repository is a key consideration. Most repositories have a default limit on a) size of individual files, b) overall size of the archive, or c) both. In some cases, exceptions to the default may be granted upon request. Check the documentation for the repository you're interested in using.

1.4.1 Curated vs. Open

Some repositories including ICPSR offer to curate your data, including review of the dataset and preparation of documentation. This is an excellent option, especially if you have a dataset to distribute that will not change. If however you have a 'live' dataset that will be updated over time and/or may require bugfixes, you may instead prefer a repository that allows you to update the dataset on-demand. ICPSR offers a non-curated option, openicpsr.org, for researchers who prefer this approach. Dataverse and Zenodo are also non-curated repositories.

1.5 Archiving & version control

If your dataset is "final" in that it will never be updated, this is not an issue. If however, you plan to make improvements or keep the dataset up to date, users will want to know that they can return to an old version of your dataset. Perhaps they submitted a paper using the original version you posted, but cannot find the file they downloaded. Or, they may want to compare versions of your dataset. This is difficult if the repository contains only the latest version.

ICPSR and Zenodo, among others, stamp each upload with a unique DOI such that the version of the dataset downloaded by a given user can be retrieved. Note that not every user may recall the DOI they downloaded, or the date on which they downloaded the data! If the repository does not automatically give each release a new version number, you will want to update this manually.

Of course no repository can guarantee that it will be available forever. You may want to consider the institution standing behind the repository when choosing, or make your data available in multiple repositories.

1.6 Access control & tracking

Again, posting the data on one's personal website may be the most expeditious method of sharing a dataset, but this may leave you with no idea regarding who or how many people have used the data. Particularly if you are funded by a foundation or agency interested in diffusion/impact, it may be important to keep track. Moreover, if you have a particular point of view on the appropriate use of your data, you may want to screen potential users.

One approach is to gate access to your dataset, restricting its use to particular people or for particular purposes. For example, Jan Bena's excellent disambiguation of patent assignees is hosted at a private repository and one must request access to the data including explaining the intended use. This approach surely slows diffusion but enables you to know who has retrieved your data. You might impose conditions disallowing sharing of data with non-registered persons, as unenforceable as such provisions may be.

Other sites are not selective in terms of usage but still require registration. OpenICPSR and the FIVES repository are both examples of this approach. It is unclear to me whether those who post data are able or allowed to see who has downloaded the data. Doing so does impose a slight friction on potential users.

The most "open" approach is not to require registration. This can be accomplished by posting data on one's own website, but then it may be difficult to track how often the data have been accessed. Dataverse and Zenodo both maintain a model where users need not register before retrieving data, but counts of downloads are kept. This approach forfeits insight into the nature of the user base but removes frictions.