Data Versioning Use Cases

Web Sources

W3C Data on the Web Best Practices

https://www.w3.org/TR/dwbp/#dataVersioning

Datasets published on the Web may change over time. Some datasets are updated on a scheduled basis, and other datasets are changed as improvements in collecting the data make updates worthwhile. In order to deal with these changes, new versions of a dataset may be created. Unfortunately, there is no consensus about when changes to a dataset should cause it to be considered a different dataset altogether rather than a new version. In the following, we present some scenarios where most publishers would agree that the revision should be considered a new version of the existing dataset.

Wikipedia Page on Software Versioning

https://en.wikipedia.org/wiki/Software_versioning

Software versioning is the process of assigning either unique *version names* or unique *version numbers* to unique states of computer software. Within a given version number category (major, minor), these numbers are generally assigned in increasing order and correspond to new developments in the software. At a fine-grained level, revision control is often used for keeping track of incrementally different versions of electronic information, whether or not this information is computer software.

Modern computer software is often tracked using two different software versioning schemes—an internal version number that may be incremented many times in a single day, such as a revision control number, and a *released version* that typically changes far less often, such as *semantic versioning*[1] or a project code name.

Semantic Versioning

http://semver.org/

In the world of software management there exists a dread place called "dependency hell." The bigger your system grows and the more packages you integrate into your software, the more likely you are to find yourself, one day, in this pit of despair.

In systems with many dependencies, releasing new package versions can quickly become a nightmare. If the dependency specifications are too tight, you are in danger of version lock (the inability to upgrade a package without having to release new versions of every dependent package). If dependencies are specified too loosely, you will inevitably be bitten by version promiscuity (assuming compatibility with more future versions than is reasonable). Dependency hell is where you are when version lock and/or version promiscuity prevent you from easily and safely moving your project forward.

As a solution to this problem, I propose a simple set of rules and requirements that dictate how version numbers are assigned and incremented. These rules are based on but not necessarily limited to pre-existing widespread common practices in use in both closed and open-source software. For this system to work, you first need to declare a public API. This may consist of documentation or be enforced by the code itself. Regardless, it is important that this API be clear and precise. Once you identify your public API, you communicate changes to it with specific increments to your version number. Consider a version format of X.Y.Z (Major.Minor.Patch). Bug fixes not affecting the API increment the patch version, backwards compatible API additions/changes increment the minor version, and backwards incompatible API changes increment the major version.

I call this system “Semantic Versioning.” Under this scheme, version numbers and the way they change convey meaning about the underlying code and what has been modified from one version to the next.

RDA Sources and Use Cases

RDA Data Citation Recommendation

https://www.rd-alliance.org/group/data-citation-wg/outcomes/data-citation-recommendation.html

Digitally driven research is dependent on quickly evolving technology. As a result, many existing tools and collections of data were not developed with a focus on long term sustainability. Researchers strive for fast results and promotion of those results, but without a consistent and long term record of the validation of their data, evaluation and verification of research experiments and business processes is not possible.

There is a strong need for data identification and citation mechanisms that identify arbitrary subsets of large data sets with precision in a machine-actionable way. These mechanisms need to be user-friendly, transparent, machine-actionable, scalable and applicable to various static and dynamic data types.

Data Versioning: For retrieving earlier states of datasets the data needs to be versioned. Markers shall indicate inserts, updates and deletes of data in the database.

da|ra Registration agency for social and economic data

da|ra (Registration agency for social and economic data) provides recommendations on versioning:

https://www.da-ra.de/fileadmin/media/da-ra.de/PDFs/TechnicalReport_2014-18.pdf

p. 14/15

The GESIS Leibniz Institute for the Social Sciences (www.gesis.org) is compliant with this recommendation and is operating a dree-digit-versioning.

Major.Minor.Revision

Major number starts with „1“, Minor and Revision number start with „0“ separate with „.“

First version of a data file is „1.0.0“.

1. Increase of the first digit if new data is added (e. g. waves, samples etc.)
2. Change of the second digit if corrections are made, which influence the analysis (e. g. change of values of respondents)
3. If the documentation is changed or emended (typing error or more detailed text added etc.) only the third digit will be increased

This versioning is based on the recommendations of the Data Documentation Initiative (DDI). DDI-Lifecycle 3.2

http://www.ddialliance.org/Specification/DDI-Lifecycle/3.2/drafts/IVMR_DRAFT.pdf ; page 2/3 “Versioning”

In the GESIS Data catalogue (DBK) the versioning and corresponding errata are documented. Description (in German only) please see here:

http://www.gesis.org/fileadmin/upload/forschung/publikationen/gesis_reihen/gesis_methodenberichte/2012/TechnicalReport_2012-01.pdf

page 13/14

DIACHRON project

http://www.diachron-fp7.eu/

DIACHRON is an integration project that addresses certain issues arising from the evolution of the data such as:

- Detect the changes of that happen to datasets (tracking the evolution)
- Archive multiple versions of data and cite them accordingly to make the reference of previous data feasible (archiving and citation)
- Retrieve and query previous versions (time traveling queries)
- Validate and repair various data deficiencies (curation problem)
- Identify the cause of the evolution of the datasets in respect with the real world evolution of the entities the datasets describe (provenance problem)
- Provide various quality metrics so as to enable quality assessment of the harvested datasets and determination of the datasets versions that need to be preserved (appraisal)

The DIACHRON solution aims not only to store previous versions for preservation in case of future need of them, but to create a live repository of the data that captures and highlights data evolution by keeping all data (current and previous) accessible, combined with a toolset that handles the full life cycle of the Data Web.

United States Geological Survey (USGS, Draft Policy)

Guidance on Documenting Revisions to Data Releases:

This guidance describes a formal revision process for datasets and associated metadata that have been released as an information product and require change.

Not covered in this guidance are USGS approved databases or Web data services for data that are expected to change continuously or on a schedule, with additions and updates made over time. Examples of these systems or services include NWIS, USA-NPN, and BISON. These data products have processes in place for data quality evaluation prior to data being loaded.

Revision of a data release is warranted, for example, when an error is detected and needs to be corrected (deleted, changed) for future use of the data. When correcting data errors, changes are made to the data only where needed, but no other alterations are made to either the structure or content. Another example case for revision is the release of data in stages in order to meet project timelines, so that the amount of data provided in an information product increases through subsequent versions.

If, in the revision, data are corrected or added, the data release must again be reviewed for quality and accuracy, and the modifications must be documented as described below. For substantial or major revisions (defined below, in the Version

Numbering section), the review process should be documented in the Information Product Data System (IPDS).

The revision process is described below for the following four cases:

1. Correcting an error
2. Appending new data
3. Extending the data structure
4. Archiving deprecated data

Following these cases, guidance is provided for assigning revision numbers.

Correcting an Error:

If an error is found that is not in the data itself, such as a misspelling in a data header or a site location name, replace or update the erroneous file and update the metadata record and any additional documentation to reflect the update.

If an error is found in the data, the author should correct the data release. If the error is large enough to affect outcomes of future data use, create a new data release record in the IPDS. A new version of the corrected data should note that the revised version (as opposed to previous versions) is current. The landing page should describe the error and point users to the new version of the data. The original or previous version of the data should also be preserved in case it is needed to understand previous uses, and in accordance with records management disposition schedules and litigation holds requirements. The revision process will result in a new IPDS record, an updated metadata record, updates to the online documentation including a revision history, and a new incremental version number (e.g., version 1.1, refer to "Version numbering" below.). All review requirements apply to the new version. Once revised data are released, the citation should reflect the new incremental version of the data as shown in the example citations below.

If the error could affect existing USGS scientific conclusions, consult your local Bureau Approving Official in the Office of Science Quality and Integrity (https://internal.usgs.gov/fsp/toolbox/approvingofficials.html) for guidance.

Examples of the citation change on data release landing page:

Original citation:

Klunk, O.T., 2012, Bathymetry of the Bermuda Triangle: U.S. Geological Survey data release, https://doi.org/10.5066/XXXXXXXX.

Revised citations:

Klunk, O.T., 2012, Bathymetry of the Bermuda Triangle (ver. 1.1, July 2012): U.S. Geological Survey data release, https://doi.org/10.5066/XXXXXXXX.

Klunk, O.T., 2013, Bathymetry of the Bermuda Triangle (ver. 2, May 2013): U.S. Geological Survey data release, https://doi.org/10.5066/XXXXXXXX.

Note that the title and DOI do not change but that the citation changes by adding version information. Additionally, it is possible that the year of the publication will also change.

On the landing page of the data release, include text reflecting the revision.

First release: 2012
Revised: July 2012 (ver. 1.1)
Revised: May 2013 (ver. 2.0)

Revision history:
Additionally, there should be a revision history text file available that explains exactly what changed in each revision. (See the revision history file in the example provided below for appending new data.)

Appending New Data:

Addition of data to released datasets, such as updating a project's data release with data from a new time period, place, or new field activity, requires most of the same steps as an original data release. In addition to the inclusion of new data, errors in previously released data may also be corrected. The following are required when new data is added: a new IPDS record, updated citation, updated metadata record, updated revision history, and text on the landing page reflecting the new version. **NOTE:** The new IPDS data release record is used to ensure requirements of SM 502.7 and SM 502.8 have been met. No new digital object identifier should be created. That is, use the existing DOI for the revised data release.

For an example, see Pendleton, E.A., Ackerman, S.D., Baldwin, W.E., Danforth, W.W., Foster, D.S., Thieler, E.R., and Brothers, L.L., 2016, High-resolution geophysical data collected along the Delmarva Peninsula, 2014, USGS Field Activity 2014-002-FA (ver. 4.0, October 2016): U.S. Geological Survey data release, https://doi.org/10.5066/F7MW2F60

Extending the Data Structure

There are cases in which the data structure is modified to allow the inclusion of new data types through the addition of new tables or fields. The extended structure is then considered a new version. These revisions are appropriate for data releases that are stand-alone research products, rather than the data foundations of scientific reports. In this case, the requirements include: a new IPDS record, updated citation, updated metadata record, updated revision history, and text on the landing page reflecting the new version. Changes should reflect a new version of the data release (e.g., version 2.0, refer to "Version numbering" below.).

NOTE: The new IPDS data release record is used to ensure requirements of SM 502.7 and SM 502.8 have been met. No new digital object identifier (DOI) should be created. That is, use the existing digital object identifier for the extended data structure.

Archiving Deprecated Data:

When data with identified errors are corrected and replaced - for example, as a new incremental version - the version with errors should not be publicly offered, but may be available on request, to future users. Because the errored data may have been used to support conclusions in a publication or a policy decision, there may be future consequences; therefore, it is essential to preserve the original data, for example in a dark archive (an offline location for preservation), with errors intact. The filename and accompanying documentation should make clear that the data are deprecated. This provides a snapshot in time of the data in terms of provenance, while ensuring that they are not recommended for future use. If size constraints make archiving a full copy impractical, some other process should be provided for making the original data available.

Version numbering:

Version numbers consist of two parts, a major and a minor component, separated by a period. In the example "version 1.2," the number to the left of the period, "1," is the major component and represents the number of separate major revisions. The number to the right of the period, "2," is the minor component and represents the number of separate substantial revisions.

The original release is considered version 1.0. Either the major or the minor component of the version number will be incremented when a revision is released. When a major revision is released, the major component increases by one number and the minor component is reset to zero (0). Substantial revisions (see definition below), regardless of how many, do not trigger a change in the major component of the version number. For example, if the data release was revised on seven separate occasions for substantial revisions, the version number will be 1.7.

Minor Revisions:

Minor revisions that are so insignificant that they do not affect the use or interpretation of the data include, but are not limited to: correcting misspelled words in data or metadata; and improvements in presentation of ancillary information on data landing pages. There is no version numbering system for these types of minor revisions and therefore no need to develop a version history document.

Using a ScienceBase (sciencebase.gov) data release page as an example, a minor revision could correct a misspelled word in the title or in the abstract. In another example, the author may wish to revise one of the contacts listed on the landing page. In other words, in a minor revision, the data did not change.

Action: No new version number required.

Substantial Revisions:

Substantial revisions are corrections to the data or metadata that are large enough to affect outcomes of future data use. Such errors typically involve missing or incorrect data values, but could also be missing or unclear annotations in table headings or in metadata records. Substantial revisions might also improve the usability or interpretation of the product content such as a modification in a polygon shapefile, slightly shifting a line so that a western boundary is consistent with another polygon shapefile that was just released.

An example of a substantial revision would be correcting a geospatial file in which a small number of negative longitudes were entered as positive numbers. The revision would change the incorrect longitude values to negative numbers. In a substantial revision, some of the data have been changed.

Action: Create a new minor component number (for example, version 1.0 is changed to version 1.1).

Major Revisions:

Major revisions include changes in the data structure and updates that add or modify substantial amounts of data. Also included are large corrections to data, for example, correcting a data file in which many data values were consistently incorrect as a result of improper processing.

An example of a major revision would be a new release of a bathymetry grid when an error was detected in the processing step that applied tide corrections. The data themselves have undergone a significant change.

Action: Create a new major component number (for example, version 1.0 is changed to version 2.0).