Making a research project understandable

Guide for data documentation

Siiri Fuchs & Mari Elisa Kuusniemi
Helsinki University Library, Data Support

Layout by Emma Niemi

Thanks to Liisa Siipilehto (Helsinki University Library, UH), Juuso Ala-Kyyny (UH), Katja Moilanen (Finnish Social Science Data Archive, FSD), Mari Kleemola (FSD), Jessica Parland-von Essen (CSC), and Arto Teräs (CSC) for valuable feedback, when making this guide.

Version 1.2, 4.12.2018
DOI: 10.5281/zenodo.1914401

Foreword

This guide was made upon request for instructions how to document research data. As no coherent guidance gathering different documentation methods was available, we wanted to make a compact guide for researchers from University of Helsinki. Nevertheless, everyone finding this document useful is free to use it, and all suggestions how to improve and develop this guide are more than welcome. Please send us feedback to datasupport@helsinki.fi.

Contents

1. Introduction

What?

Data documentation means describing the data. This should be planned before starting the data collection by creating a data management plan (DMP) to accompany the actual research plan.

How?

The level and specificity of documentation is something each project can choose themselves. The documentation methods introduced in this guide provide a basic structure of how to document data. By varying these, we hope each project finds it easier to select the right methods for them.

Data documentation includes a variety of documents which describe all data used in a project. The most commonly used methods are gathered to the checklist found on next page. Data documentation can be done by using metadata standards, which are discipline specific forms widely used, or e.g. electronic laboratory notebooks, which create metadata about the project, while keeping the notes up to date. Documentation includes also data dictionaries, codebooks, vocabularies and readme-files, which all take part in explaining what the project data is, how it has been collected, what the abbreviations mean and how the data has been modified.

In addition, to keep all documents related in a project well-structured and findable, one should pay attention to file naming, directory structure, version control and tagging files with specific key words.

Why?

Investing time in documenting the data makes it more understandable for you as well as others. The data will be in better overall shape when all variables, abbreviations and codes have been explained, as well as having readme files explaining the content of folders. This will also decrease the risk of false interpretation of the data.

Having invested in documentation during the project will save time upon publishing the dataset. Especially if the publication process takes long, it is easy to forget details of data collection and processing, if they are not well documented. Good documentation is also very important for the principles of open science: having the data well documented will ease sharing it to collaborators or to fellow scientists after publication.

Checklist for documentation

This guide will introduce the following documentation methods. It can be used as a quick overview of the topic as well as a checklist on what documentation methods have been or will be used in the project.

ELEMENTS OF DATA DOCUMENTATION
Metadata standards & vocabularies • Describe data in a controlled format using vocabularies • Use field and disciplinary specific standards, if suitable standards exist • Should always be favored, if suitable for project
Data management software, i.e. databases and electronic laboratory notebooks • Software take data in and create a database • Makes documentation easier as software usually generate metadata by themselves • Easy to share & control access, usually have safe storage & search tools • Easy error spotting: inputs out of range can be automatically detected
Data dictionaries • Dictionaries explain variables used in a dataset • Codebooks are collections of codes, algorithms and calculations used in a project
Directory structure • Create a folder structure to suit your project needs • If you work with sensitive data, a clear folder system helps also in access control • Balance between shallow and deep folder hierarchy to keep files findable
Tagging files • Tags are keywords assigned to files, which enable organizing and searching files easier • A file can only be in one folder at a time, but it may have an unlimited number of tags
File naming conventions • Create a meaningful but brief system with unique names (in case of directory structure break-down) in the beginning of the project
Version control • Version control makes it possible to return to an older version of a specific file • Automatic version control system preferred
Readme-files • Readme-files are text documents (e.g. format.txt) providing information about data files to ensure they are interpreted correctly • Readme-files can include information such as title, creator, description, location, methodology, dates and file formats
Discovery metadata • Descriptive metadata, “label of the dataset”, where the dataset is explained should always be published regardless of the nature of the data • Persistent identifiers (PIDs) identify citable online resources providing a permanent link to them. Persistent identifiers are used when citing and managing data and information
Research records • Administrative documents (i.e. licenses, usage agreements, consents used) and other research related documents (e.g. research plan, publications, DMP) explaining the context of the actual metadata

1.1. Metadata standards and vocabularies

Metadata standards and vocabularies describe data in a controlled way. Hence, they should be favored when choosing documentation method. Using standards and coherency in documentation will make data more understandable, facilitate its reuse and make combining datasets possible. The easiest way to apply metadata standards is to use a program tool or services which generate metadata in a preferable format (see section 1.2.).

Metadata standards are structured formats that use specific **vocabularies** or **ontologies** in describing the data. This ensures that data is understood the same way regardless who the user is. Metadata standards vary across research fields and disciplines, and some are better established than others. Some data repositories require the use of a metadata standard. Therefore, choosing a repository in the beginning of the project makes it easier to select a metadata schema. It is recommendable to use a standard even though all the study variables do not fit in it.

Finding metadata standards:

- Digital curation centre (DCC) has gathered discipline specific metadata standards and provides tools to implement them.
- FAIRsharing is a curated, informative and educational resource on data and metadata standards, inter-related to databases and data policies.
- EMBL-EBI Ontology lookup service is a repository for biomedical ontologies that aims to provide a single point of access to the latest ontology versions.
- Data vocabularies (Tietomallit), a service for managing and publishing data vocabularies. It contains data component libraries, i.e. data specifications for harmonizing information used jointly by different actors.

1.2. Data management software, databases & electronic lab notebooks

There are tools for making data management easier; for example **data management software** are developed to take various data in, convert it into a database, and generate metadata by themselves, when new inputs are made. Inputs, which are way out of range, can also be more easily detected with data programs making the data more reliable. Before starting to use a software, check that it uses some standard to avoid incompatibility problems with the data later on.

An example of a software is REDCap, which is a secure web application for building and managing online surveys and databases. While REDCap can be used to collect virtually any type of data, it is specifically geared to support online or offline data capture for medical research studies and operations.

Be aware that many manufacturers have software which produces its own kind of metadata or formats, making the data incompatible with other programs. However, usually manufacturers do have the standard metadata version/format available. Checking this is worthwhile.

Researchers can organize and store experimental procedures, notes, protocols and data using an **electronic lab notebook**. These can be used on computers or mobile devices. The advantage of these in data documentation is that metadata is created automatically in the software. Additionally, electronic notebooks can be used remotely, they can be shared, and they have secure data storage, access control, and search-tools.

Splice-bio made a list of best electronic lab notebooks available (spring 2018):

- Scinote
- Benchling
- Rspace
- More options and reviews in the end of the Splice article.

Another option is Jupyter Notebook, which is an open source web application for creating and sharing many kinds of documents. See also the Penna tool provided by the Finnish Social Science Data Archive for written data.

2. Documentation during research project

The ways of documenting data depend on the project: what methods, devices, programs have been used, what are the communicational needs of the research group, as well as are there any field specific traditions to be taken into account. All processes are different, and hence, there is no single correct way of documentation.

To give some general advice for documentation, follow these rules:

1. If possible, use metadata standards and controlled vocabularies.
2. If available, use data management software, to make documenting easier.
3. Get familiar with the following topics in this chapter, if standards and software are not suitable for your project:
   - Data dictionaries (&Codebooks)
   - Directory structure
   - Tagging files
   - File naming
   - Version control
   - Readme-files

2.1. Data dictionaries

Data dictionaries, sometimes referred as codebooks, **explain variables** used in a dataset. In order to avoid misunderstanding, codebooks can also be understood as collections or description of codes, algorithms and calculations used in a project.

Sheet_1

Show rows with cells including:

Variable	Variable name	Mesaurement unit	Allowed values	Description
Participant ID number	ID	Numeric	001-999	ID number assigned to participant in sequential order
Group number	GROUP	Numeric	1-30	Group assigned to participant based on ID number
Age in years	AGE	Numeric	18.0-65.0	Age of participant in years
Date of birth	DOB	mm/dd/yyyy	1-12/1-31/1951-1998	Participant's date of birth
Gender	SEX	Numeric	1 = male 2 = female	Participant's gender
Date of survey	SURVEY	mm/dd/yyyy	01/01/2015 – 01/01/2016	When the participant completed the survey
Self-reported consumer spending	SPEND	Numeric	0-100,000,000	Self-reported average yearly expenditure
Market sentiment	SENTIMENT	Numeric	1 = negative 2 = neutral 3 = positive	Sentiment towards US domestic economy
Actual GDP growth	GDP	Numeric	-5.0-5.0	Average US yearly GDP growth

Figure 1. Data dictionary from Open Science Framework Help

Information explaining variables used in the data

A data dictionary should begin with basic information of the study. This includes information such as title, researcher name, table of contents and a description of the purpose and format of the data dictionary.

Example of information in a data dictionary:

- **Variable name**, as they are named in your spreadsheet, should be one word only
- **Variable label**, a description of the variable
- **Definition of the variable** so that it is understood in the same way regardless of the user, define the variable preferably by using a vocabulary
- **Units of measure**, i.e. minutes, milligrams, meters etc.
- **Value ranges and allowed values**, e.g. 1-10, female/male
- **Variable universe**, from which group/set is the variable information from, e.g. if the variable is based on a question that is asked only from women or under 50 years old participants.
- **Value codes/labels**, if variables are coded numerically, what do the codes represent (e.g. 0=no, 1=yes for yes/no variables)
- **Question text**, i.e. the exact survey question.
- How the variable was measured (e.g. nominal, ordinal, scale)
- How the variable is recorded in raw data (i.e. numeric, string) and e.g. how many decimals does it have.
- **Summary statistics**, if applicable.
  - Categorical variables: frequency counts showing the number of times a value occurs
  - Continuous variables: minimum, maximum, median values
- **Missing data**, how are missing values coded in the data.
- Notes

(References: Open science framework help, McGill university, Penn libraries and ICPSR sites.)

2.2. Directory structure

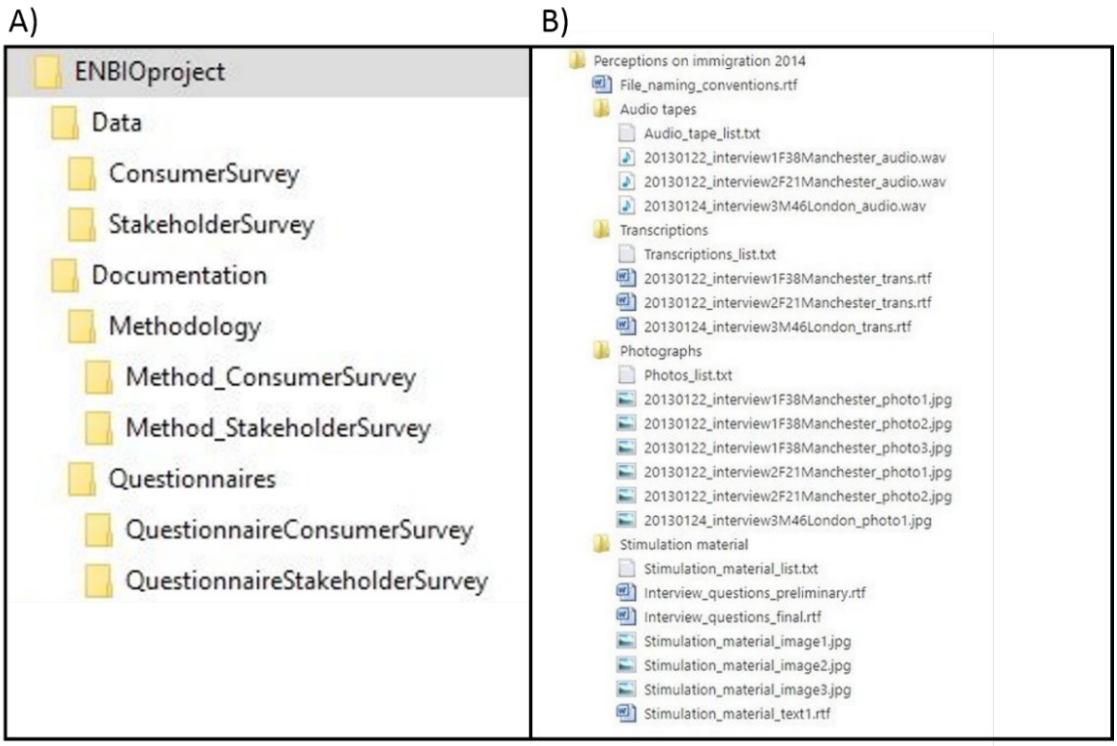
Create a directory structure (= folders) to suit your project needs. If you work with sensitive data, a clear folder system helps also in access control.

When planning a **hierarchical directory structure**, take in consideration:

- What sort of data will you have?
  - Are there many subprojects which need their own folders?
  - How should different raw data, cleaned data, methods, documentation, manuscripts or presentations be organized?
- Balance between shallow and deep folder hierarchy to keep files findable.
  - Too deep needs many clicks to get to the right file.
  - Too shallow can end up having too many files in one folder.
- Avoid overlapping categories and make meaningful folder names.

If you have sensitive data be thoughtful when naming files and folders, since this kind of structural information and metadata might sometimes be visible for outsiders without access to the data itself. Do not use personal names or social security numbers in file names. Preferably also use unique file and folder names to safeguard against corruption that might happen if directories or contexts are lost.

Folder structure example from Cessda eric data management site:



Example A) *For this survey, data and documentation files are held in separate folders. Data files are further organized according to data type and then according to research activity. Documentation files are organized also according to the type of file and research activity. This helps to restrict the level of folders to three or four deep and not to have more than ten items in each list.*

Example B) *The data contain audiotapes of the interviews, interview transcripts, stimulation material shown to the research subjects and photographs taken by the subjects. Data files are files connected to the same interview event conducted on 22nd January 2013. The latter part of the name reveals the specifics of the file. In this case, "audio" means audio tape and "trans" a transcription of the audio tape. However, background information must never be stored in the file name only.*

2.3. Tagging – find similar files quick from different folders

Tags are keywords assigned to files. A file can only be in one folder at a time, but it may have an unlimited number of tags. Tags are a simple way to add metadata to files and organize them flexibly.

Best practices:

- Simple system with simple tag names
  - Keep tag names short: one or two words maximum
  - Be consistent with names, i.e. plural-singular/ capital letters/ symbols: Report vs report vs reports
  - You can have e.g. a high hierarchical tag combined with a lower level tag: manuscript + [project name in one word], raw data + [type of the data, e.g. x-ray]
- Search for guidance for different software.
  - General tagging guide can be found from MIT Libraries page.
  - Tagging Windows files

2.4. File naming conventions – the title tells it all!

Plan your naming system in the beginning of your project and create a meaningful but brief system. Never use same name twice; in case of directory structure breakdown, you may lose data.

Naming tips to keep in mind:

1. Balance with the amount of **elements** in the name: too few making it too general vs. too many hinder understandability. Limit the name to 32 characters or less.
2. Use meaningful **abbreviations**.
3. Order the elements **from general to specific**.
4. Use the underscore (_) as element delimiter and hyphen (-) or capitalizer to delimit words within an element. Don't use special characters: & , * % # ; * ( ) ! @$ ^ ~ ' { } [ ] ? < >.
5. **Time** should be ordered: year, month, day (YYYYMMDD) or more specifically if needed: hours, minutes, seconds (HHMMSS).
6. For **version control** use the letter V followed minimum by two digits (V06), and extend it if needed for minor changes (V06-02). Remember the leading zeros to make sure files sort correctly.
7. Write a **readme-file about the naming system** and explain abbreviations (example below).
8. Make your research group & collaborators use the file naming system.

Tips gathered from Purdue University Data Management site & article by Vincent Santaguida (Exadox).

An example of a file naming system:

The system

- Project title in one word or abbreviation and number: e.g. HB1 (Honey bee project 1)
- Experiment name: e.g. EXP1 (experiment 1)
- File type: e.g.
  - DATA for data files
  - RM for readme-files
  - CB for codebooks
- Date of the file: 20180620 (YYYYMMDD and time if necessary HHMMSS)
- Version of the file: e.g. _V04

The file name when system is used

HB1_EXP2_DATA_20180611_V01-02

2.5. Version control – track changes in your data

Version control makes it possible to return to an older version of a specific file. It is possible to use an automatic software (always preferable) or manually keep track of files.

Automatic software that creates and organizes versions:

- Software available at University of Helsinki:
  - Wiki, a confluence platform, where you can invite also members outside University of Helsinki
  - GitLab version control platform is primarily meant for version control of application codes. To start a project at GitLab, you need an UH account, but you can use the software also with HAKA-identifier or by registering as a new user
- Git (instructions for its use)

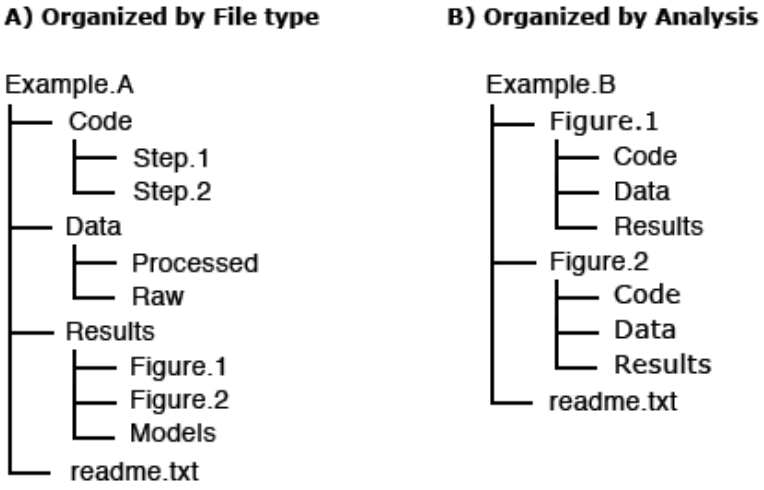
Manual control, where user manages all files:

- Can be used in small project, where only one user is managing the files
- Important to name files correctly: e.g. version indicator in the end of the file name (V02-03)
- Generate an archive folder for old versions

2.6. Readme-files

Readme-files are text documents (e.g. format .txt) providing information about data files to ensure they are interpreted correctly. These become especially important when sharing and publishing data, but they are also helpful to your future self.

The figure below demonstrates where a readme-file should be placed in a folder/directory (source: Dryad):



Example of topics in a README file:

- TITLE: Name of the dataset or research project that produced it.
- CREATOR: Names and addresses of the organization or people who created the data.
- DESCRIPTION of the data package and folder overview.
- LOCATION: Where the data relates to a physical location, record information about its spatial coverage.
- METHODOLOGY: How the data was generated, including equipment or software used, experimental protocol, other things you might include in a lab notebook.
- INFORMATION ABOUT DATA FILES:
  - IDENTIFIER: Number used to identify the data files, even if it is just an internal project reference number.
  - LOCATION: Where to find data files and additional information such as the data dictionary explaining variables used.
  - DATES: Key dates associated with the data, including project start and end date, data modification data release date, and time period covered by the data.
  - SUBJECT: Keywords or phrases describing the subject or content of the data.
  - FILE FORMATS: What file formats have been used.

Other relevant information such as the funder and usage rights can be put to another readme-file, which specifies research records, information that is important for standalone understandability of the data (see 3.3.).

3. Documentation for publishing data

Metadata for publishing enables the **discoverability** of the data, whereas all metadata mentioned in chapters one and two are created for the understandability of the data. Metadata in this case is more like the "label of the dataset", where the dataset is described. Metadata should always be published regardless of the nature of the data. This applies also to datasets which cannot be published as Open Access due to confidentiality issues (personal data/sensitive data).

What metadata exactly is needed for publishing depends on where the dataset is published as well as how long will it be preserved.

3.1. What to include to the dataset when publishing it?

Every repository will give detailed instructions what is needed to publish a dataset. This is only a general list what a dataset can include:

- Metadata referred as "the label of the dataset", describing basic information of the dataset is always **mandatory**, such as title, description, owners, usage rights, methodology, and persistent identifier. This makes the dataset findable for reuse.
- The data files in understandable formats. Possible cleaned as "tidy data", where each data file has only one kind of variables.
- The data files **must be accompanied with**:
  - Standard metadata, if created
  - Appropriate readme-files explaining how the data has been collected & modified, as well as what each data file contains
  - Data dictionaries and codebooks for making variables understandable
  - Codes and algorithms used
  - Vocabularies
  - Licenses used
  - Version control information, i.e. is this the first public version of the dataset
- Raw data, if possible

In short, include all metadata of the data created during the research project. The better documentations is done during the project, the easier it will be to publish it.

Publishing data in a repository will give the data a persistent identifier and files will be kept in accessible formats, which will increase data reuse. Rights to the data can also be easily managed by licensing the data and using potential embargo periods.

3.2. Persistent identifier (PID)

Persistent identifiers are used when citing and managing data and information. PIDs identify citable online resources, such as datasets or publications, by providing **a permanent link** to them. Even if the object's location in the internet changes, the identifier remains the same and will still link to the data, regardless of the new location.

DOI (Digital Object Identifier) and URN (Uniform Resource Name) are examples of a commonly used persistent identifiers.

How to get a PID?

When publishing data or articles, the publisher/ repository will provide a unique identifier. PIDs can also be received for own data archives. One should never use the same identifier twice; even if an updated version of a data is created, a new PID should be used. Remember that a PID is always a promise that the underlying data will remain unchanged.

3.3. Research records

When publishing or archiving a dataset, especially when preserving the data for long time (e.g. next generations), standalone readability of the data becomes very important as there is nobody from who to ask in case of unclarity. For this, good data documentation done for the understandability (chapters 1 and 2) is yet not sufficient. The data and its metadata needs **research records** alongside to make them usable; these can be administrative documents and other research related descriptions which explain the context of the data.

Research records are documents important for the project such as

- Permission for reuse & licenses
- Usage and data handling agreements
- A copy of a consent form, if used, and information sheet
- Scientific publications from the data
- Research plan
- Data management plan
- Device/ equipment descriptions
- Methodology descriptions

Adding a **readme-file** about above mentioned documents is recommendable. The content could be the following:

- FUNDERS: Organizations or agencies who funded the research.
- RIGHTS: Any known intellectual property rights held for the data.
- LICENSES used.
- LANGUAGE: Language(s) of the intellectual content of the resource, when applicable.
- LOCATION: Where the data relates to a physical location, record information about its spatial coverage.
- METHODOLOGY: How the data was generated, including equipment or software used, experimental protocol, other things you might include in a lab notebook.
- DEVICE AND EQUIPMENT DESCRIPTIONS
- INFORMATION ABOUT RESEACH RECORD FILES:
    - Agreements
    - A copy of a consent form, if used and information sheet
    - Scientific publications from the data
    - Research plan
    - Data management plan