Agroecology Partnership: Data Management Guidelines

LifeWatch ERIC

August 13, 2025

Table of contents

Introduction

Welcome to the Data Management Guidelines documentation of the Agroecology Partnership.. This online documentation expands on the official Agroecology Partnership Data Management Plan (DMP) and complements it with more practical and technically detailed information.

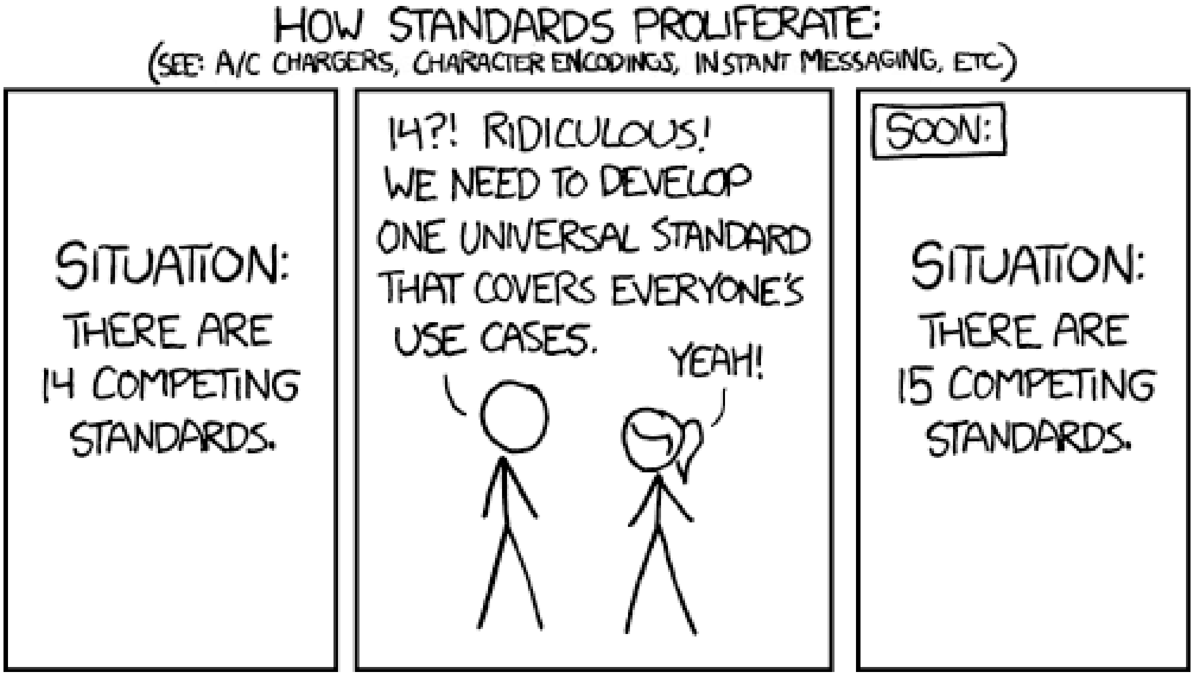
While the DMP outlines the full strategy for data handling across the partnership and it is updated every 2 years, the present documentation is meant to be updated as we learn from doing. Its objective is to support partners with hands-on guidance, concrete examples, and step-by-step explanations of the data pipeline — from collection to publication — including how to make data FAIR (Findable, Accessible, Interoperable, and Reusable). This space aims to be more to the point than the DMP, helping teams implement the plan more effectively in day-to-day work. Furthermore, being a public document, we aim to offer the wider agronomy community with state-of-the-art data management practices in the Agroecology domain.

For any questions, please write us a line at agroecology-data@lifewatch.eu.

Part I

Data Standarization

Data on Agroecology are multidisciplinary: There are biological data, socio-economic, geographical and many more. The main goal is to standardize data to the preferred community standard. However, sometimes there is not a clear winning standard, or the winning standard may not meet the FAIR principles. This chapter describes the preferred data standards for each data type collected in the Agroecology Parnership. In some cases, when there is not an obvious standard, we suggest evidence-based way of structuring data that fits the FAIR principles.


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Source: xkcd

1 Tabular data

Tabular data are often referred as **data structured in rows and columns, where each row contains values for a set of properties and each column represents a specific property of the things described by the rows** (Tennison and Kellogg 2015). Following certain best practices and well-studied structures when creating tabular data make it easier to analyze, visualize and reuse - by others or by your future self.

1.1 Best practices for Tabular Data

Tabular data can **follow very different structures**. There is a lot of flexibility in how you name your columns or what values you put in every cell. This flexibility is at the same time a **double-sided knife**: follow the right recommendations, and your data will be easy to analyze. Do the wrong decisions, and your same data could become not usable. Best practices go around columns names and consistency. Here we will describe common mistakes and how to solve them:

1.1.1 File formats

While you are free to use proprietary spreadsheet software such as Microsoft Excel or Google Sheets to work on your data, **we strongly recommend storing datasets in `.csv` or tab-delimited formats**. This ensures compatibility across software and, most importantly, protects your work from accidental obsolescence. Imagine you open an old project and find that, after a software update, some formulas no longer work, column types have shifted, or the file won't even open.

Open formats like `.csv` are readable by virtually any tool, now and decades from now. If a problem ever arises, there's a global community of developers who can build solutions to recover your data. Proprietary formats are fine for analysis, but their internal rules can change without notice, and you might only discover it when it's too late.

In short: **use Excel or Sheets to crunch numbers if you like, but keep your raw data in `.csv`** and publish both raw and processed datasets in `.csv` whenever possible. Avoid `.xlsx` as your master copy. And you can always open `.csv` in Excel for analysis without locking your work into a single program's rules.

1.1.2 Handling Empty cells

Most software handling text, tabular files, are able to recognize empty cells. This is, a cell where a value is not added. In Excel, these show as blanks, while in other tools like R or Python they show as "NA" or "NULL". **We recommend to leave empty cells when a value is not known**. A common mistake is to add 0's in a numeric variable when a value is unknown. This could distort further analysis as 0's are considered meaningful values. Common file formats that allow empty values are `.csv`, `.txt` or `.xlsx`.

Certain formats don't allow adding empty values. In this case, it is recommended to add a placeholder such as `-9999` in numeric variables, or `NA` in text variables. This blank value must be documented properly. For instance, the binary format `NetCDF` does not allow empty cells, but you can document empty values by using the attributes `missing_value` or `_FillValue`.

1.1.3 Column names

The names used to describe your variables should be **clear and consistent**. Avoid spaces or special characters, and adopt standard units. Examples:

- `max_temp_celsius` instead of `Max Temp Celsius`
- `soil_ph` instead of `Soil-pH`

We recommend to use lowercase and underscores _ instead of spaces. This style is known as `snake_case`. Other well-known style is `camelCase`, where spaces are avoided and words are spitted by using a Capital letter to emphasize the different words.

1.1.4 Table schema

A table schema is a data dictionary that defines each variable in your data table, recording information such as the data type of the column (numeric, date, string, boolean…), units or possible values. You can save this in a separated file and send it together with your data. We elaborate about this in the following chapters where we also show some examples and available standards and tools.

1.1.5 Adhere to community standards

Think of data like a written language: if everyone makes up their own grammar, nobody can understand each other. In data, this applies not only between humans but also between humans and machines. **We rely on computers to analyze our data, but unlike us, they're terrible at guessing meaning**. To them, data are just zeros and ones. In science, the key is in the context, so **we need to shape our data so both people and machines can read them correctly.**

The easiest way to do that? **Follow community standards**. Chances are, you're not the first person working in your field. Others have faced the same problems and worked hard to solve them. Communities dealing with certain types of data — geospatial, surveys, biodiversity, genetics — have already agreed on rules and formats so that other researchers and software can make sense of the data. Take dates as an example. You could write them in many ways: `DD/MM/YYYY`, `MM/DD/YYYY`, or even with month names. But most software understands them best in the format `YYYY-MM-DD`. This is not arbitrarty: it's an international standard (ISO 8601) designed after much discussion and testing.

In the next chapters, we will look at several community standards relevant to Agroecology, and we will recommend which ones to use for maximum clarity and interoperability.

1.1.6 Separate raw data from calculations

Raw data should always live apart from any data cleaning, wrangling, or analysis. Even if your raw file has typos, strange characters, or awkward formatting — leave them as they are. They're part of the original record.

Do all cleaning and analysis in a separate file or script (R, Python, Julia, Excel formulas…) and share that too. **This will enable anyone can reproduce your process and get the same results**. This is of course, unless you are unsure of your analysis and embarrassed to show to others. In that case you should proofread until you are happy.

Why keep the artifacts? Because it preserves a trustworthy copy of the original, makes your work reproducible, and gives you a safe point to return to if something later goes wrong.

1.2 What is Tidy data?

In practice, following the principles of tidy data can make tabular datasets easier to work with (Wickham 2014). **In tidy data, each row represents a single observation, each column a variable, and each table a distinct type of observation**. For example, consider this table with grape yield and wine production across three European regions . A non-tidy table might look like this:

farm	grape_yield	wine_production
Bavaria (DE)	7000 kg/h	60 hl/ha
La Rioja (ES)	5000 kg/h	40 hl/ha
Peloponnese (GR)	6000 kg/h	50 hl/ha

Here, different variables are spread across columns, and units are written inside the values. While this structure is easy to understand for most of us, it is harder to analyze for machines:

- **Units mixed with values:** The measurement units are embedded in the cell content, making it difficult to perform numeric calculations without first extracting and converting the values to numbers.
- **Variable names as column headers:** Each variable requires a separate column, so adding new variables (e.g., sugar content, harvest date) would require adding new columns, breaking automation.
- **Inconsistent row structure:** If some observations are missing a value, it's harder to handle missing data systematically.
- **Limited scalability:** Combining multiple datasets or reshaping data for plotting, statistical analysis, or modeling requires extra preprocessing steps.
- **Filtering and grouping complexity:** Aggregating or comparing values across variables requires multiple steps instead of simple column-based operations.

A tidy version of the same data represents each measurement as a separate row with explicit columns for the variable type, value, and unit, so that **each row contains a single observation of a single variable**.

Farm	Variable	Value	Unit
Bavaria (DE)	Grape Yield	7000	kg/h
Bavaria (DE)	Wine Production	60	hl/ha
La Rioja (ES)	Grape Yield	5000	kg/h
La Rioja (ES)	Wine Production	40	hl/ha
Peloponnese (GR)	Grape Yield	6000	kg/h
Peloponnese (GR)	Wine Production	50	hl/ha

This tidy format makes it easy to filter, group, or plot by crop type, farm, or yield, and it aligns with modern data analysis workflows. Following these principles helps ensure that tabular data are both **clear** and **practical** for downstream use.

2 Survey data

Recommendations for organizing survey data in the Agroecology partnership following a wide, spreadsheet format approach. This structure is inspired by the data structure of the American National Election Studies (ANES 2020), the recommendations of (Zimmer, Powell, and Velásquez 2024), while applying a tidy data approach (Wickham 2014) with usability on mind and the enabling of a later transformation into Open Linked Data as structured in The Survey Ontology (Scrocca et al. 2021).

2.1 Survey project structure

We propose a project structure that uses `csv` files as a way of formally describing the survey, both questions, answer and other information. This structure may be accompanied of a text description of the survey for further context. A minimal example of the file directory would look like:

```
.
    data/
      codebook.csv
      responses.csv
    docs/
       survey_descriptor.pdf
```

The data files are inside the data directory, and the different documents including the description of the survey are inside the docs directory.

This structure can be further extended, depending on the needs of the survey. For instance, a README file can be added to the root of the project with a short explanation of the survey. When uploading to Zenodo, this can be written instead in the Description of the metadata. If the survey is uploaded to GitHub, we recommend to use the markdown format for the README file. Also on GitHub, it is recommended to include a LICENSE file, choosing the license of the data.

In addition, any notebook or spreadsheet used to analyze the data can be added to the analysis folder, and reusable code scripts can be added to the scripts directory.

```
.
   README.md
   LICENSE.md
    data/
      codebook.csv
      responses.csv
    docs/
      survey_descriptor.pdf
      survey_analysis.pdf
    scripts/
      clean_data.R
    analysis/
       survey_analysis.xlsx or survey_analysis.ipynb
```

2.2 Survey project description file

A survey is always quite specific, and cannot be fully understood without a good explanation of what is the context and the rationale behind the different questions. It is also a sign of respect to the respondents to explain what is the goal of your survey. We propose to include this description in a `survey_descriptor.pdf` file in the `docs/` directory.

2.3 Survey project data files

Surveys are typically are typically stored as `pdf`, `docx` or `xlsx`. Having interoperability on mind, we propose to use **text delimited files** such as `csv`. These type of files are largely used in data science. they have several advantages, including easy machine-readability and being an open format with no owner, which ensures data will remain readable and understandable by many different software for a long time. We propose the following specifications:

- Separate columns using semicolons `;`
- Use double-quotes `"` to quote strings
- Use `UTF-8` as encoding.

We avoid using colons `,` as separators because these can be used in free text, open questions. They would affect the structure of the data. We recommend to prohibit the use of semicolons `;` and double-quotes `"` in any case that is not separating columns and delimiting strings respectively. We encourage you to use software solutions that allow blocking these characters in free-text answers provided by users. If blocking the use of these characters it is not possible, we urge you to ask respondents to not use them while filling up your survey.

2.3.1 Codebook

Codebooks are files that **explain the questions** formulated in the survey. An unique identifier (a code) is assigned to each question, linking the information about the questions with the answers provided by the participants. But the codebook can be used as well during the design of the survey.

The example below shows an hypothetical survey codebook about the user satisfaction using an online platform.

`./data/codebook.csv`

Code	Label	Type	QuestionText	Values	Cardinality	QuestionType
Q1_age	Age	integer	What is your age?		1..1	SingleChoice
Q2_gender	Gender	string	What is your gender?	Female|Male|Other	0..1	SingleChoice
Q3_satisfaction	Satisfaction	integer	How satisfied are you with the platform?	1=Very unsatisfied|2=Unsatisfied|3=Neutral|4=Satisfied|5=Very satisfied	1..1	SingleChoice
Q4_improvement	Improvement	string	What would you improve in the platform?		0..n	FreeText

Each row in the codebook describes a survey question. Below is an explanation of each column:

- **Code**: A unique identifier for the question. It must be unique within the survey.
- **Label**: A short, human-readable label or name for the variable that can be used in spreadsheets or statistical software.
- **Type**: The data type of the answer. Common types include “integer”, “string”, “boolean”, or “date”.
- **QuestionText**: The full text of the question as it was asked in the survey.
- **Values**: A list of possible values for closed questions. Options are separated by vertical bars (|), and value labels can be assigned using the equals sign (=). For open or free-text questions, this field is left empty.

- **Cardinality**: Indicates how many answers are allowed. The cardinality pattern is`min..max`, where the first number is the minimum required answers and the second is the maximum allowed. `n` denotes “no fixed upper limit,” so `1..1` means exactly one answer, while `0..n` means the question may be skipped or answered multiple times.
- **QuestionType**: Describes the nature of the question. Typical values include “SingleChoice”, “MultipleChoice” or “FreeText”

2.3.2 Responses

Answers to the survey are recorded in the `./data/responses.csv` file. **Every row is the answers of a participant**, and **every column is named after the `code`** in `codebook.csv`. This allows to link easily the information about the questions without getting the responses file full of details that difficult the analysis.

`./data/responses.csv`

respondent_id	Q1_age	Q2_gender	Q3_satisfaction	Q4_improvement
001	34	female	4	I think the platform is user-friendly
002	29	male	5	Needs better support for collaboration.

We recommend to include an unique identifier for every respondent, here named as `respondent_id`. This allows to annomymize the survey without loosing the link to private information about the respondents that might have been collected (e.g. name, email, address)

2.3.3 Participants and private information

Sometimes your survey might collect personal or sensitive information — like names, emails, or locations. If that’s the case, you **should never share this data publicly**. In the Agroecology Partnership, we follow the **GDPR** (the General Data Protection Regulation), which means that **private data must be handled carefully and securely**.

We recommend creating a separate file (not to be published!) called `participants.csv` inside your `data/` folder. This file is **for internal use only** — for example, if you need to follow up with participants or validate something later. To keep things tidy and safe, give each person a unique code, which we’ll call `respondent_id` and store personal info (like name or email) in separate columns. Use clear, human-friendly column names, and write them using `snake_case` (i.e. separate the words of the column names with underscores `_`).

respondent_id	name	email	country	preferred_language	wants_emails
001	Jane Doe	jane.doe@example.com	France	fr	FALSE
002	John Doe	john.doe@example.com	USA	en	TRUE

The table above is an example (with mocked information) of how the `participats.csv` would look like. This file helps you **anonymize your actual survey data**, because in the main `responses.csv`, you'll only keep the `respondent_id`, and this is nothing that could personally identify someone.

References

ANES. 2020. "Time Series Study Full Release: User Guide and Codebook." https://electionstudies.org/wp-content/uploads/2022/02/anes_timeseries_2020_userguidecodebook_20220210.pdf.

Scrocca, Marco, Daniele Scandolari, Giulia Re Calegari, Irene Baroni, and Irene Celino. 2021. "The Survey Ontology: Packaging Survey Research as Research Objects." In *Proceedings of the 2nd Workshop on Data and Research Objects Management for Linked Open Science – Co-Located with ISWC 2021*. https://doi.org/10.4126/FRL01-006429412.

Tennison, Jeni, and Gregg Kellogg. 2015. "Model for Tabular Data and Metadata on the Web." World Wide Web Consortium; W3C Recommendation. https://www.w3.org/TR/2015/REC-tabular-data-model-20151217/.

Wickham, Hadley. 2014. "Tidy Data." *Journal of Statistical Software* 59 (10): 1–23. https://doi.org/10.18637/jss.v059.i10.

Zimmer, Samantha A., Ryan J. Powell, and Iván C. Velásquez. 2024. *Exploring Complex Survey Data Analysis Using r: A Tidy Introduction with {Srvyr} and {Survey}*. Chapman & Hall/CRC Press. https://tidy-survey-r.github.io/tidy-survey-book/.