Data organisation ABC

Siiri Fuchs, Hanna Koivula, Tuija Korhonen, Tanja Lindholm, Päivi Rauste, Liisa Siipilehto

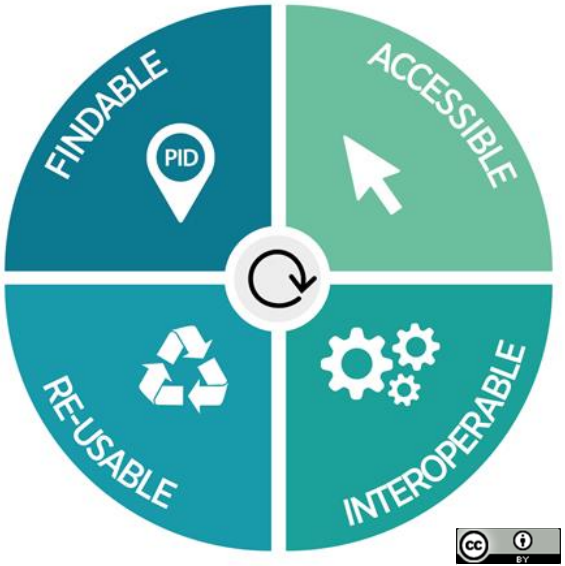
CC BY 4.0 Siiri Fuchs, Hanna Koivula, Tuija Korhonen, Tanja Lindholm, Päivi Rauste, Liisa Siipilehto

Contents

Well-organised and documented data is the foundation for the entire research lifecycle!

When data is well organised and documented, it is easy to share, open and re-use.

By following a few basic principles of data processing and documentation, you will be many steps closer to FAIR data.



CC BY

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Why organise and document?

1. Well-designed file names and folder structures make it easier to find and track data files
2. Data is easier to use, share, open, access and preserve.
3. Standardized practices increase the value of the data and its reusability.
4. Good documentation reduces the risk of data being misinterpreted.
5. Time spent on careful documentation saves time both during the project and in the opening phase.
6. Good metadata increases the discoverability of data.

Well organised and well documented

Good documentation includes:

1. Data collection methods: sampling, how the data was collected, what equipment and software was used
2. Quality assurance methods
3. File and folder structure
4. Version control
5. Information about the terms of access and use or the confidentiality of the data
6. Names, labels and descriptions of variables, datasets and their values
7. An explanation or definition of the codes and classification systems used
8. Definitions of any special terms or abbreviations used
9. Codes for missing values and the reasons for them

Designing the folder structure

- For whom are you designing a data organisation system? For yourself, as a researcher, for the needs of the project, for partners?
- Is the project short term or long term? Is a simple folder structure sufficient or do you need a more complex structure?
- Who should have access to the folders? List all the people (including potential ones) who will need access to the files on your system.
- Making copies and backups and controlling access is efficient with the help of an organised folder structure (e.g. if the project contains sensitive data, a particular folder can be protected with a password).

Functional folder and file structure

- Create a separate folder for each project by name and date (e.g. abbreviation + year).
- In addition to data files, create separate folders for project management, methods, text files, etc.
- Consider who should have access to the folder
- Use unique names for files and folders.
  - The names of the files in different folders must not be confused when the folder structure is disassembled (do not use the same file names in different folders and on the other hand name the files so that you know which folder they belong to (some identifier)).
- The right balance between shallow and deep folder hierarchy helps to find the right file.
  - Avoid folders within folders. For example, don’t create different folders for different years, despite the fact that you want to keep the years separate. Instead, name the files uniquely and keep them in the same folder.

Good folder structure & content

A good structure includes at least the following elements:

1. Unique main folder for the project
2. Codes
3. Data
4. Readme document containing all important information about the project (there can be more than one). There must be at least one readme at the top level of the folder, covering at least administrative matters.

Data can be divided into different folders, for example like this:

- Individual main folder for the project
  - Code
  - Data
    - Raw data
    - Edited data
    - Final data
- Readme

https://mitcommlab.mit.edu/be/commkit/file-structure/#ChooseScaleAim

https://github.com/mitcommlab/Coding-Documentation/blob/master/File-Structure-Case-Studies.md#case-study-2-a-simple-hierarchy

Examples of folder structures


```
project/
  code/                code needed to go from input files to final results
  data/                raw and primary data (never edit!)
     raw_external/
     raw_internal/
     meta/
  doc/                 documentation of the study
  intermediate/        output files from intermediate analysis step
  logs/                logs from the different analysis steps
  notebooks/           notebooks that document your day-to-day work
  results/             output from workflows and analyses
     figures/
     reports/
     tables/
  scratch/             temporary files that can safely be deleted o
  README.txt           file and folder description
```

From: RDMKit, Elixir.

Case Study 2: A Simple Hierarchy

```
PROJECT/
├── bin/            <- compiled binaries.
├── data/
│   ├── raw/
│   └── clean/
│
├── figures/        <- figures used in place of a "results" folder.
├── scripts/
│   ├── process/    <- scripts to maniuplate data between raw, cleaned, final stages.
│   └── plot/         <- intermediate plotting.
│
├── src
│   ├── model1/     <- various experimental models.
│   ├── model2/
│   └── model3/
│
├── LICENSE
├── Makefile
└── readme.md
```

From: Three examples of file structures for different project types

20221202_Mountain_EXP2_Kilpis_DATA_V01.xls

Naming files

- The file name is the main identifier of the file:
  - Short & descriptive file names tell what the file contains and make it easier to organise the data, but only if the naming is consistent.
- The file naming convention should be **agreed at the beginning of the project** by the whole group and the names should include elements relevant to the project. It is important that everyone in the project follows an agreed file naming convention.

Example elements to include in filenames

- Creation date
- Project number / experiment number / acronym
- Data type: sample ID, analysis, conditions, changes, etc.
- Place / coordinates
- Author's name / initials
- Version number
- The last three letters indicate the file format (e.g. .xls, .rtf, .mov, .tif, .doc)

From:
https://docs.csc.fi/data/datasets/metadata-and-documentation/#data-organization
https://rdmkit.elixir-europe.org/data_organisation#what-is-the-best-way-to-name-a-file

Tips for naming files

20221202_Mountain_EXP2_Kilpis_DATA_V01.xls

1. Balance the number of elements: too many make it difficult to understand, too few make it generic.
2. Arrange the elements from general to specific.
3. Use meaningful abbreviations.
4. Use an underscore (_), hyphen (-), or capital letter to separate elements in the name. Do not use spaces or special characters: ?!& , * % # ; * ( ) @$ ^ ~ ‘ { } [ ] < >.
5. Use as date format (ISO8601): YYYYMMDD (year, month, day), and time, if necessary, HHMMSS (hours, minutes, seconds).
6. Include the version number in the name if necessary: at least two numbers (V02) and extend if necessary for minor fixes (V02-03). The first zeros allow the files to be ordered correctly.
7. Write down your naming convention and explain the abbreviations in your documentation (e.g. Readme file).
8. If you have to rename numerous files in order to manage the files more easily, it is possible to use applications, e.g. Bulk Rename Utility (Windows, free), Renamer4Mac (Mac).

Example: Mountain project, experiment 2, Place Kilpisjärvi, file created 2.12.2022

- Name of the file: 20221202_Mountain_EXP2_Kilpis_DATA_V01.xls
- Explanation: Time_ProjectAbbreviation_ExperimentNumber_Location_TypeOfData_VersionNumber

20221202_Mountain_EXP2_Kilpis_DATA_V01.xls

20221202_Mountain_EXP2_Kilpis_DATA_V02.xls

Version control helps you keep your data organised!

- Version control can be done either manually, or by adding a sequential number (_v03) to the end of the filename, or automatically, which is the preferred method.
  - Automated version control can be implemented using software such as Git, GitHub or GitLab (your organisation may offer an integrated solution).
- You can also use cloud storage solutions, which usually offer automatic file version control. There are several version control systems available, both open source and commercial, closed implementations.

20221202_Mountain_EXP2_Kilpis_DATA_V01.xls

20221202_Mountain_EXP2_Kilpis_DATA_V02.xls

Things to remember when versioning

- Version control produces a (modified) copy of the file, uniquely marked with a version number. Version control allows reverting to a previous version, which is important for data traceability, the tracking of edits and error correction.
- When making new versions of data files, it is important to keep a copy of the original raw data.
- When sharing data, it can also be useful to provide both unprocessed and processed versions of your data and include either code or explanations of how the final version is produced
- Plan and agree which versions of the data will be archived and/or published.

README.txt - file

```
|-- README.txt
|-- data
|    |-- raw
|    |-- process
|    |-- metadata
|-- scripts
```

- The README- file binds the parts of the dataset together. It can be used to collect and describe the data history (lineage), i.e. the relationships between individual files, collection methods, data quality information, intended use and restrictions.
- *README-file records consist of the documentation and data quality information generated during data processing (and version control).*
- Acts as a repository of ‘tacit knowledge’ for you and your research team.
- Makes data easier to publish and understand.
- Provides guidance on the re-use of data.

In addition, file or database level metadata describes how the files that make up the dataset relate to each other; what format they are in; and whether they replace or are replaced by previous files. A folder-level readme.txt file is the classic way of accounting for all the files and folders in a project.

https://carpentries-incubator.github.io/spreadsheet-humanities-lesson/02-common-mistakes/

README file

```
|-- README.txt
|-- data
|   |-- raw
|   |-- process
|   |-- metadata
|-- scripts
```

- A ReadMe file is a simple text file, such as a .txt
- It should be clearly named using good naming conventions (not mandatory)
- It should contain information about when it was created or updated
- The date is in ISO 8601 format: YYYYMMDD
- The file contains the author's name and contact details

For datasets, the minimum requirement is information on the following

- When was the data collected/produced?
- Data licence
- Links to publications using the data
- Links to openly available datasets
- Reference to dataset
- Descriptions of the datasets
- Reference to a file containing e.g.:
  - description of variables or classifications (**code book**)
  - external vocabularies used in the data
  - method descriptions and linkage of data/files to them

Keep track of your analysis

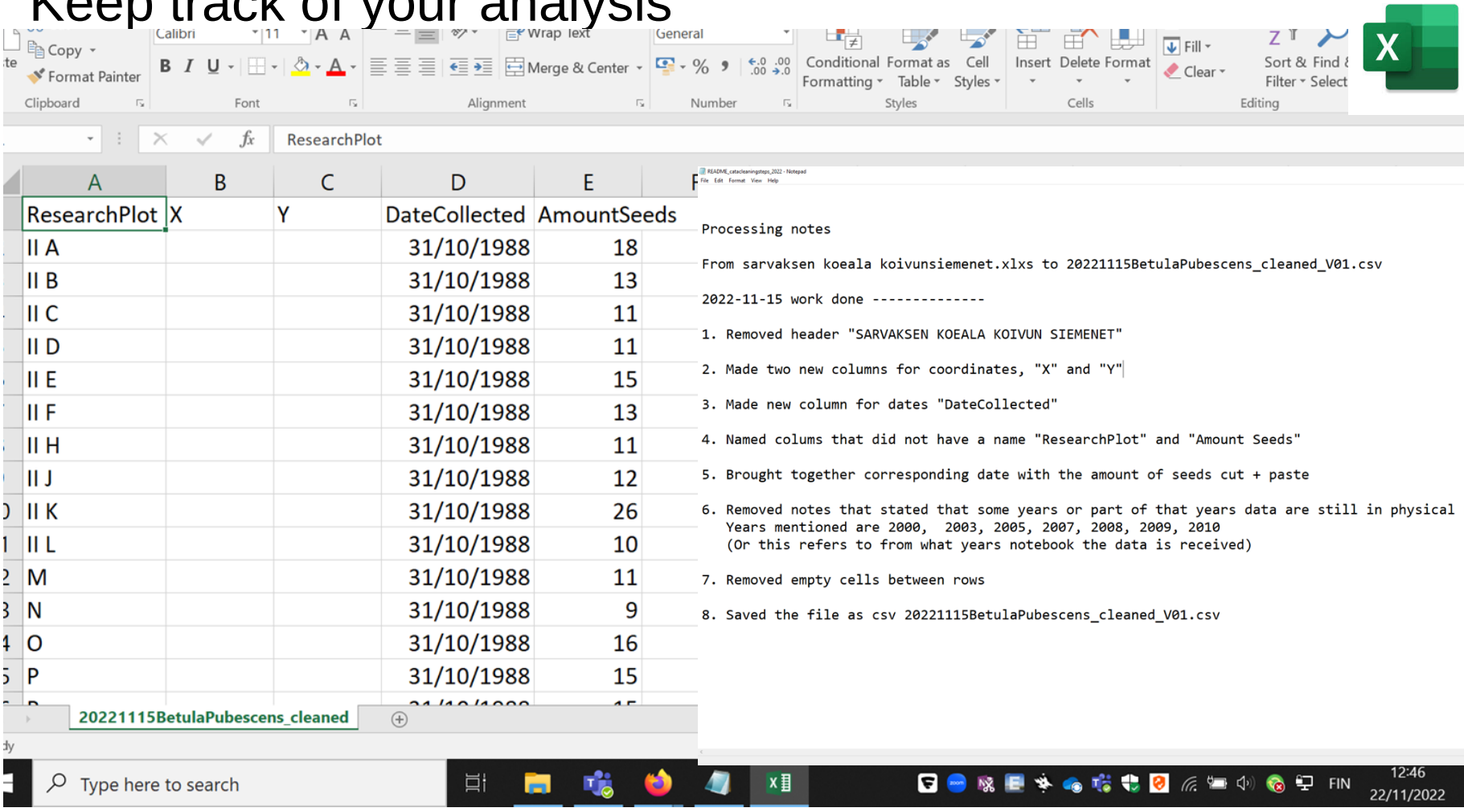
To reproduce your analysis and to remember what you did, you should

- Create a new file with your cleaned or analysed data. Never modify the original dataset!
- Keep track of the steps you took in your cleaning or analysis.
    - What are the steps did you take from the raw data to the analysis data. For example, use a plain text file (txt) to document the steps you took and store it in the same folder as the data file.
- It is recommended to use R, Python or another programming language to clean up your data. Above your code write clearly what the different steps do # this does this and that

This might be an example of how to do it with the plain text file:

Text adapted from: Hoyt, P., et al.(2019): Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869

Keep track of your analysis



ResearchPlot	X	Y	DateCollected	AmountSeeds
II A			31/10/1988	18
II B			31/10/1988	13
II C			31/10/1988	11
II D			31/10/1988	11
II E			31/10/1988	15
II F			31/10/1988	13
II H			31/10/1988	11
II J			31/10/1988	12
II K			31/10/1988	26
II L			31/10/1988	10
M			31/10/1988	11
N			31/10/1988	9
O			31/10/1988	16
P			31/10/1988	15

Exercise 1

If necessary, the trainer can add the necessary instructions and links on how to proceed with the exercise.

How to structure data in spreadsheets

A few rules about how to keep your data tidy:

1. Put all your variables in columns — such as “weight” or “temperature”.
2. Never put units in the same cell with numeric values, but report them somewhere, such as in a variable name.
3. Put each observation in a separate row.
4. Don’t put several pieces of information in the same cell. This makes it easier to use or sort the data.
5. Leave the raw data as it is — don’t change it!
6. Export the cleaned data into a text-based format, such as CSV (comma-separated values) format. This ensures that anyone can use the data and is required by most data repositories.

Remember! Columns = Variables, Rows = Observations, Cells = Data (value).

Text adapted from: Hoyt, P., et al.(2019): Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869

Remember! Columns = Variables, Rows = Observations, Cells = Data (value)

	A	B	C	D	E	F	G	H
1	NestID	Species	BZ	Day	Month	Year	Dayofyear	
2	A_001	ACCGEN	HB	19	6	1979	170	
3	A_002	ACCGEN	HB	18	6	2001	169	
4	A_003	ACCGEN	HB	6	7	1984	188	
5	A_004	ACCGEN	HB	26	6	1987	177	
6	A_005	ACCGEN	HB	25	6	1995	176	
7	A_006	ACCGEN	HB	24	6	2010	175	
8	A_007	ACCGEN	HB	21	6	1981	172	
9	A_008	ACCGEN	HB	27	6	1983	178	
10	A_009	ACCGEN	HB	27	6	1986	178	
11	A_010	ACCGEN	HB	26	6	1985	177	
12	A_011	ACCGEN	HB	29	6	2003	180	
13	A_012	ACCGEN	HB	6	7	1992	188	
14	A_013	ACCGEN	HB	17	6	1993	168	
15	A_014	ACCGEN	HB	15	6	1979	166	
16	A_015	ACCGEN	HB	2	7	1980	184	
17	A_016	ACCGEN	HB	18	6	1992	170	

Data Example from: Hällfors, Maria et al. (2020), Data from: Shifts in timing and duration of breeding for 73 boreal bird species over four decades, Dryad, Dataset, https://doi.org/10.5061/dryad.wstqjq2ht

One Tab - One Table, One Table - One Tab

Using multiple tables

- Do not create multiple data tables within a spreadsheet!. It may look nice, but it will confuse the computer.
- By creating multiple tables within one spreadsheet, you make false associations between things for the computer, which sees each row as an observation.
- You are potentially using the same field name in multiple places, making it harder to clean up your data.

Using multiple tabs

- Using multiple workbook tabs may seem an easy way to organise data or to create metadata, but it has consequences:
- For example, you are more likely to accidentally introduce inconsistencies into your data if you start recording data in a new tab every time you take a measurement.
- Even if you manage to prevent all inconsistencies from creeping in, you are still adding an extra step to your analysis. If you need to combine this data into a single data table, you will need to explicitly tell the computer how to combine the tabs.
- Also, many data repositories require your data to be entered in a single tabular file, such as csv.

The next time you enter data, and want to create another tab or table, ask yourself if you could avoid adding that tab by adding another column to your original spreadsheet. Even if your spreadsheet gets really long, you can always freeze the column headings so that they remain visible if you have a spreadsheet with many rows.

Text adapted from: Hoyt, P., et al.(2019): Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869

Using formatting, comments and units in excel sheet

1. Never use any formatting, such as highlighting, borders and such to convey information.
2. Firstly, a computer cannot read this information, and secondly information is not stable
3. Add a column each piece of information you want to convey.
4. Comments should also be in a separate column.
5. Do not use units after numbers. You can either express the information in the variable name or you can have a new column for units.
6. Remember not to use any special characters when adding comments or units.

Text adapted from: Hoyt, P., et al.(2019): Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869

Naming the variables

Give variables descriptive names!

Remember:

1) Names should not contain spaces, numbers or special characters. These confuse computers.
2) Instead of using spaces, use underscores (_)
3) You can also use Capital Letters to separate words, for example ExampleFileName
4) Names should not be too long!
5) Remember to document your naming convention, especially if you use abbreviations. This will enable you and others to fully understand what they mean.

Text adapted from: Hoyt, P., et al.(2019): Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869

Naming variables, an example

	A	B	C	D	E	F	G	H
1	NestID	Species	BZ	Day	Month	Year	Dayofyear	
2	A_001	ACCGEN	HB	19	6	1979	170	
3	A_002	ACCGEN	HB	18	6	2001	169	
4	A_003	ACCGEN	HB	6	7	1984	188	
5	A_004	ACCGEN	HB	26	6	1987	177	
6	A_005	ACCGEN	HB	25	6	1995	176	
7	A_006	ACCGEN	HB	24	6	2010	175	
8	A_007	ACCGEN	HB	21	6	1981	172	
9	A_008	ACCGEN	HB	27	6	1983	178	
10	A_009	ACCGEN	HB	27	6	1986	178	
11	A_010	ACCGEN	HB	26	6	1985	177	
12	A_011	ACCGEN	HB	29	6	2003	180	
13	A_012	ACCGEN	HB	6	7	1992	188	
14	A_013	ACCGEN	HB	17	6	1993	168	
15	A_014	ACCGEN	HB	15	6	1979	166	
16	A_015	ACCGEN	HB	2	7	1980	184	
17	A_016	ACCGEN	HB	18	6	1992	170	

Hällfors, Maria et al. (2020), Data from: Shifts in timing and duration of breeding for 73 boreal bird species over four decades, Dryad, Dataset, https://doi.org/10.5061/dryad.wstqjq2ht

Not filling in zeros

Why bother writing in the number zero if, for example, your observations in the survey are mostly zeros?

1) There is a difference between a zero and a blank cell in a spreadsheet. For the computer, a zero is data that you have measured or counted. An empty cell means it hasn't been measured and the computer will interpret it as an unknown value (null value).
2) Most programs are likely to misinterpret empty cells as null values that you intended to be zeros. This can naturally cause problems with your analysis and misinterpretations.

Text adapted from: Hoyt, P., et al.(2019): Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869

Using problematic null values

Never use 0 as a null value! Statistical programs will not recognize that this is intended to represent missing values.

There are a few possibilities to choose from, depending on the final use of your data and how you intend to analyse it.

The important thing is to use a **clearly defined** and **consistent** null indicator.

Table 1. Commonly used null values, limitations, compatibility with common software and a recommendation regarding whether or not it is a good option. Null values are indicated as compatible with specific software if they work consistently and correctly with that software. For example, the null value "NULL" works correctly for certain applications in R, but does not work in others, so it is not presented in the table as R compatible.

Null values	Problems	Compatibility	Recommendation
0	Indistinguishable from a true zero		Never use
Blank	Hard to distinguish values that are missing from those overlooked on entry. Hard to distinguish blanks from spaces, which behave differently.	R, Python, SQL	Best option
-999, 999	Not recognized as null by many programs without user input. Can be inadvertently entered into calculations.		Avoid
NA, na	Can also be an abbreviation (e.g., North America), can cause problems with data type (turn a numerical column into a text column). NA is more commonly recognized than na.	R	Good option
N/A	An alternate form of NA, but often not compatible with software		Avoid
NULL	Can cause problems with data type	SQL	Good option
None	Uncommon. Can cause problems with data type	Python	Avoid
No data	Uncommon. Can cause problems with data type, contains a space		Avoid
Missing	Uncommon. Can cause problems with data type		Avoid
-,+,.	Uncommon. Can cause problems with data type		Avoid

https://datacarpentry.org/spreadsheet-ecology-lesson/02-common-mistakes/index.html

Ethan P White, Elita Baldridge, Zachary T. Brym, Kenneth J. Locey, Daniel J. McGlinn, Sarah R. Supp, Nine simple ways to make it easier to (re)use your data Vol. 6 No. 2 (2013): Special Issue - Data Sharing in Ecology and Evolution

Text adapted from: Hoyt, P., et al.(2019): Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869

Inclusion of metadata in the data table

- It is essential to create proper metadata in order to be able to read the data.
- However, metadata should not be included in the data file itself, as this information is not data and its inclusion may affect how computer programs interpret your data file.
- It is recommended that you store a separate metadata file in the same directory as your data, preferably in a plain text format with the same name that clearly associates it with your data.
- A plain text format allows you to encode comments, units, information about null values etc. Any information that is needed to read your data.

Text on metadata adapted from the online course Research Data MANTRA by EDINA and Data Library, University of Edinburgh. MANTRA is licensed under a Creative Commons Attribution 4.0 International License and from the online course Data Organization in Spreadsheets for Ecologists licensed under CC-BY 4.0 2018–2023 by The Carpentries

Dates in data

Always use widely known ISO8601 standard: YYYYMMDD (year, month, day) and when necessary HHMMSS (hours, minutes, seconds).

It is highly important to use four numbers for the year.

For example recording year only with two numbers, such as 17 or 99, can represent either year 1917 or 2017 and 1899 or 1999.

Remember that there are datasets that go back hundred years and possibly even longer.

	A	B	C	D	E	F	G	H
1	NestID	Species	BZ	Day	Month	Year	Dayofyear	
2	A_001	ACCGEN	HB	19	6	1979	170	
3	A_002	ACCGEN	HB	18	6	2001	169	
4	A_003	ACCGEN	HB	6	7	1984	188	
5	A_004	ACCGEN	HB	26	6	1987	177	
6	A_005	ACCGEN	HB	25	6	1995	176	
7	A_006	ACCGEN	HB	24	6	2010	175	
8	A_007	ACCGEN	HB	21	6	1981	172	
9	A_008	ACCGEN	HB	27	6	1983	178	
10	A_009	ACCGEN	HB	27	6	1986	178	
11	A_010	ACCGEN	HB	26	6	1985	177	
12	A_011	ACCGEN	HB	29	6	2003	180	
13	A_012	ACCGEN	HB	6	7	1992	188	
14	A_013	ACCGEN	HB	17	6	1993	168	
15	A_014	ACCGEN	HB	15	6	1979	166	
16	A_015	ACCGEN	HB	2	7	1980	184	
17	A_016	ACCGEN	HB	18	6	1992	170	

Data example from: Hällfors, Maria et al. (2020), Data from: Shifts in timing and duration of breeding for 73 boreal bird species over four decades, Dryad, Dataset, https://doi.org/10.5061/dryad.wstqjq2ht

Text adapted from: Hoyt, P., et al.(2019): Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869

Exercise 2

If necessary, the trainer can add the necessary instructions and links on how to proceed with the exercise.

References

- Slides 6-9
  - MIT School of Engineering Communication Lab;
    - https://mitcommlab.mit.edu/be/commkit/file-structure/#ChooseScaleAimshttps://mitcommlab.mit.edu/be/commkit/file-structure/#ChooseScaleAim
    - https://github.com/mitcommlab/Coding-Documentation/blob/master/File-Structure-Case-Studies.md#case-study-2-a-simple-hierarchy
    - https://github.com/mitcommlab/Coding-Documentation/blob/master/File-Structure-Case-Studies.md#three-examples-of-file-structures-for-different-project-types
  - RDMkit Elixir: https://rdmkit.elixir-europe.org/data_organisation#how-do-you-organise-files-in-a-folder-structure
- Slides 10-13
  - CSC: https://docs.csc.fi/data/datasets/metadata-and-documentation/#data-organization
  - Elixir: https://rdmkit.elixir-europe.org/data_organisation#what-is-the-best-way-to-name-a-file
  - Siiri Fuchs, & Mari Elisa Kuusniemi. (2018). Making a research project understandable - Guide for data documentation (1.2). Zenodo. https://doi.org/10.5281/zenodo.1914401
- Slides 20, 24
  - Data Example from: Hällfors, Maria et al. (2020), Data from: Shifts in timing and duration of breeding for 73 boreal bird species over four decades, Dryad, Dataset, https://doi.org/10.5061/dryad.wstqjq2ht
- Slides 14, 16, 19, 21-23, 25-28
  - **The Carpentries.** Text mainly adapted from online course Data Organization in Spreadsheets for Ecologists licensed under CC-BY 4.0 2018–2023 by The Carpentries
    - Peter R. Hoyt, Christie Bahlai, Tracy K. Teal (Eds.), Erin Alison Becker, Aleksandra Pawlik, Peter Hoyt, Francois Michonneau, Christie Bahlai, Toby Reiter, et al. (2019, July 5). datacarpentry/spreadsheet-ecology-lesson: Data Carpentry: Data Organization in Spreadsheets for Ecologists, June 2019 (Version v2019.06.2). Zenodo. http://doi.org/10.5281/zenodo.3269869
    - https://datacarpentry.org/spreadsheet-ecology-lesson/
    - https://datacarpentry.org/spreadsheet-ecology-lesson/02-common-mistakes/index.html
  - EDINA and Data Library, University of Edinburgh: https://rdmkit.elixir-europe.org/data_organisation#what-is-the-best-way-to-name-a-file
  - Tidy data: https://www.jstatsoft.org/article/view/v059i10
  - Data organization in spreadsheets: https://peerj.com/preprints/3183/
  - Ethan P White, Elita Baldridge, Zachary T. Brym, Kenneth J. Locey, Daniel J. McGlinn, Sarah R. Supp, Nine simple ways to make it easier to (re)use your data Vol. 6 No. 2 (2013): Special Issue - Data Sharing in Ecology and Evolution