Effective Research Data Management & Reproducible Research Computing

Institute of Agricultural Policy and Markets, *Retreat 2026*, Bad Herrenalb

Anna Rudaev & Konstantin Kuck

Office for Research Services, Kommunikations-, Informations- und Medienzentrum (630)
University of Hohenheim

March 12, 2026

Efficiency

- consistently organized data
- computational reproducibility
- automation

Quality and Integrity

- traceable analysis
- reduced risk of errors
- transparency
- compliance with regulations

Collaboration

- clear project structure
- version-controlled code (and data)
- shared documentation

Long-Term Impact

- reusable datasets and code
- increased visibility

→ Well-managed data and reproducible workflows make research **transparent**, **re-usable**, and **trustworthy**.

Well-documented research data and reproducible data analytical workflows also help to meet the demands of:

- **Journal publishers** who increasingly require **submission** of the **data** upon which **scientific results are based** to **strengthen transparency and accountability**.
- **Research funders** who increasingly mandate **open access** to **all outputs** of research to foster re-use of data generated in scientific processes.

Outline

1. Foundations of Research Data Management
   - Research Data Life Cycle
   - FAIR Principles
2. Guidelines and legal aspects for Research Data Management
   - Hohenheim Open Science Guidelines
   - Principles of Good Scientific Practice
   - Copyright and licensing
   - Data availability statements and citation of research data
3. Support in the context of Research Data Management
4. Concepts, Tools and IT-Services for Research Data Management
   - Data Management Plan
   - Project organisation
   - Data documentation
   - Reproducibility

1. Foundations of Research Data Management

Research Data Lifecycle

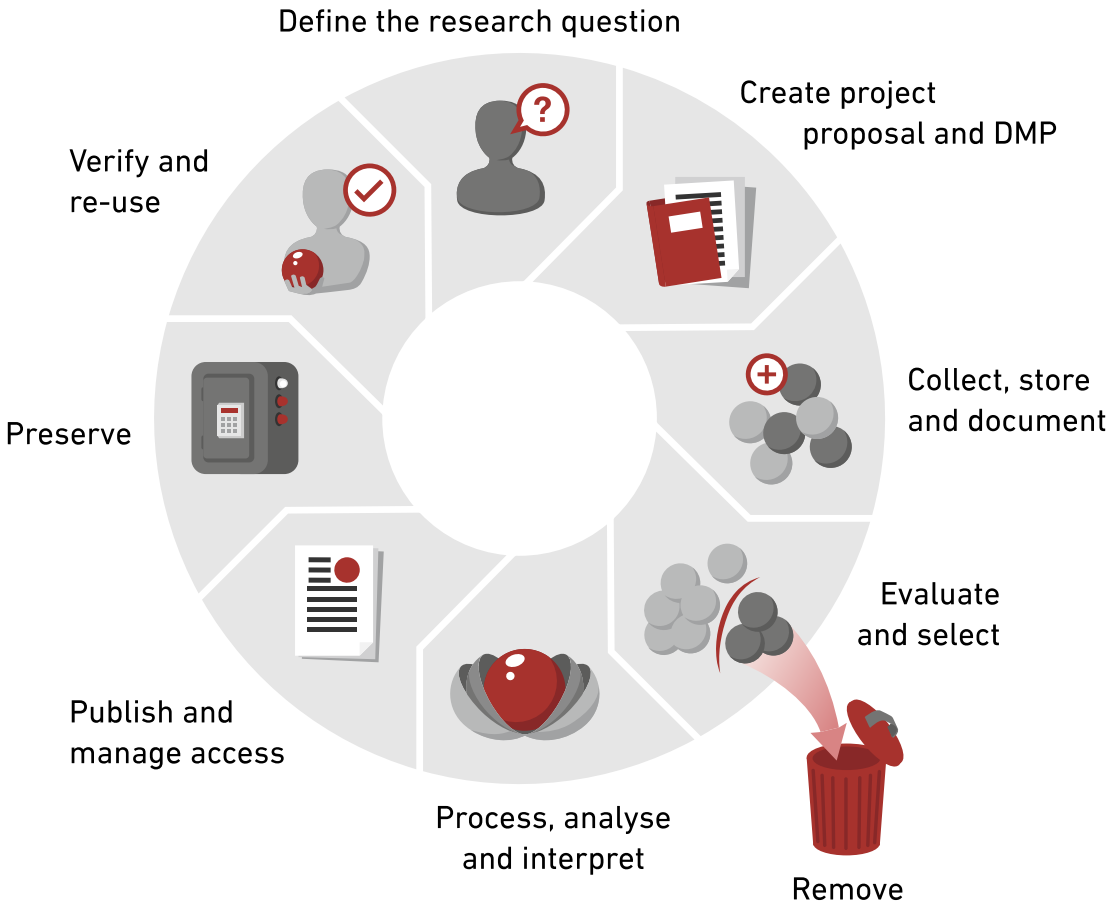


Figure: The research (data) life cycle.

Source: https://library.ethz.ch/en/researching-and-publishing/data-management-and-policies/research-data-management/research-data-life-cycle.html, accessed: July 15, 2024.

In line with the notion of a research data life cycle, Wilkinson et al. (2016a) propose four guiding principles with regard to research data.

Specifically, research data should be "FAIR", in the sense that it is:

- Findable
- Accessible
- Interoperable
- Reusable

Essentially, they aim to enable and facilitate re-use and integrability of data sets generated in the context of different research projects.

Naturally, this has various implications not only with regard to the scientific data generating processes, the research data itself, but also with regard to research IT-infrastructures.

Before going into the details of the **FAIR principles**, we discuss the **concept of metadata** which forms their technical foundation.

Metadata is data **about** (primary) research data.

In contrast to other forms of documenting data, **metadata** is a **structured description** of primary data.

Many file-formats typically have some metadata directly embedded, for instance, image files or Office documents. This often is a valuable source of information about primary data.

In the scientific context, we are often interested in **further augmenting** primary data by **descriptive metadata** with respect to the **study object, tools and data set structure**.

The aim of these structured descriptions is to strengthen the understanding of scientific data and enable and foster their findability and re-use.

Often, discipline-specific metadata standards have been established to help collect relevant metadata. Metadata Standards Catalog by Helmholtz Metadata Collaboration and KIT

Box 2 | The FAIR Guiding Principles

To be Findable:
F1. (meta)data are assigned a globally unique and persistent identifier
F2. data are described with rich metadata (defined by R1 below)
F3. metadata clearly and explicitly include the identifier of the data it describes
F4. (meta)data are registered or indexed in a searchable resource

To be Accessible:
A1. (meta)data are retrievable by their identifier using a standardized communications protocol
A1.1 the protocol is open, free, and universally implementable
A1.2 the protocol allows for an authentication and authorization procedure, where necessary
A2. metadata are accessible, even when the data are no longer available

To be Interoperable:
I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (meta)data use vocabularies that follow FAIR principles
I3. (meta)data include qualified references to other (meta)data

To be Reusable:
R1. meta(data) are richly described with a plurality of accurate and relevant attributes
R1.1. (meta)data are released with a clear and accessible data usage license
R1.2. (meta)data are associated with detailed provenance
R1.3. (meta)data meet domain-relevant community standards

Figure: The FAIR data principles in detail (Wilkinson et al., 2016a).

2. Guidelines and Legal Aspects for Research Data Management at University of Hohenheim

The latest Statutes for Ensuring Good Scientific Practice at University of Hohenheim from 2022 contain relevant regulations in **§7 Quality Assurance Across All Phases**:

- scientific work must always be performed lege artis
- 'Type and scope of research data generated in the research process must be described. Their handling must be organized in accordance with the requirements of the subject in question.'

§12 Documentation regulates the documentation of research data and methods, especially for replication purposes.

Further relevant regulations are in **§17 Archiving**:

- 'When scientific results are made publicly available, the underlying research data (usually raw data) – depending on the respective discipline – is generally made **accessible and traceable for a period of ten years** at the institution where they were generated or in cross-location repositories.'

The Open Science Guidelines at University of Hohenheim from 2024 state i.a. that scientific findings, **data** and publications should be freely accessible and usable by both the academic community and the general public. The goal is open access, open data, and open materials.

Data publications should be made available to the public under the FAIR principles according to the motto 'as open as possible, as closed as necessary,' to the extent permissible and reasonable.

Copyright protection applies to **human creations** that are intellectual content and reach a certain level of creativity.

Research data is not protected by copyright law if they are pure representations of facts, like machine derived raw data.

The author has a copyright only if, e.g. substantial investment was necessary to collect the data.

Qualitative research data is likely protected.
Quantitative research data is likely not protected.

Databases are protected for 15 years after creation.

REMINDER: Data protection

Personal data is protected by **GDPR**. Processing it (collection, storing, transfer, etc.) is strictly regulated by German and EU law.

Make sure that you have a legal basis for collecting and processing personal data.

Avoid personal data if not needed.

Process, store and analyse personal data in a safe environment.

Be aware of unwanted identifiability.

Regularly check if data can be pseudonymized or anonymized.

License = Permission to use a right (to share, save, adapt, etc.).

Licenses can be commercial or free.

Free licenses maximize reuse and visibility, such as Creative Commons Licenses.

Creative Commons License Chooser

NOTE: You can only give a license or waive your rights when your research data is protected under the copyright law!

- **CC0**: Data may be used freely without asking for permission.
- **CC BY**: Data may be used freely, but attribution must be given to the original dataset, and the license must be linked to.
- **CC BY-NC**: Data may be used only for non-commercial purposes.

Data availability statements contain information on data, software, material, or reasons why data is not publicly available.

"Data, including raw data and code, are available at [repository name] [DOI]"

Overview of Examples for Availability Statements, Website Imperial College, London

Data citation is similar to citation of literature:

Author. (Date): Title. Version*. Publisher/Journal/Repository. Type of Resource*. Identifier.

3. Support in the Context of Research Data Management

Table: What support can you find at the University of Hohenheim?

KIM, Office for Research Services	**Consulting** on general RDM and **DMPs** in projects; services and information on **processing, storage and metadata aspects**; support in finding, **citing**, and formatting research data; help with publishing in repositories, licensing, and long-term archiving of data
Department Research and Transfer (AF)	Support in funding, application and transfer (e.g. application advice, contract and third-party funding support)
Ethics Committee	Ethical issues, ethical evaluation (**Ethical Approval**)
Data Protection Staff Unit	Data protection, GDPR, advise on legal and technical data protection issues, **VVT protocol**)

Table: What support can you find at the University of Hohenheim?

Core Facility Hohenheim — CFH (640)	Analysis of composition of soils, plants, foods, biological samples, identification and quantification of specific constituents, structure elucidation (chromatography, spectrometry, spectroscopy, microscopy); **Support** in data collection, data management and **data analysis** in the context **statistical and econometric methods**; bioinformatic evaluations
Biostatistics Unit (340c) Prof Dr Hans-Peter Piepho	Biometrical consulting: support in the context of **experimental sample design** and choice and application of **biostatistical methods**
DataLab Hohenheim — DALAHO (managed by CFH)	**Access** to **proprietary databases**: Refinitiv EIKON and Datastream, S&P Capital IQ, S&P Compustat, etc.

4. Concepts, Tools and IT-Services for Research Data Management and Research Computing

Data Management Plan

A Data Management Plan (DMP) outlines how research data is generated, documented, stored, secured, shared and preserved throughout the course of a project.

- Ensures that data is complete, accurate, and trustworthy
- Saves time and effort in the long term (e.g., in data research, document creation, and data exchange)
- Required by funders (e.g. DFG, VW Foundation, Horizon Europe)
- complies with the Principles of Good Scientific Practice

Create your DMP in the planning phase of your research project and update it regularly during the course of the project.

A DMP generally contains information about

- **description** of the data,
- documentation of **data quality**,
- technical **storage** during the project period,
- **legal obligations and conditions**,
- data exchange, **accessibility** and long-term preservation,
- **responsibilities** and resources.

Example of DMP - online tools:

- Research Data Management Organiser RDMO
  - Result of a DFG-funded project, since 2025 managed by the RDMO Association
  - Open Source, access via forschungsdaten.info
  - big variety of question catalogs from different funders

- DMPtool.org
  - Open Source, hosted at the California Digital Library (CDL) by the University of California Curation Center (UC3)
  - one question catalog from Digital Curation Centre

There are several templates from organisations. For example:

- Horizon Europe DMP Template, 2022, MS Word
- Science Europe Template for DMP, 2021, MS Word

FU-Berlin. Sample DMP, 2022, PDF can also be very useful.

LLMs can assist in the creation of DMPs. In particular, they can help with

- structuring,
- text generation and
- consistency checks

They are less suitable for

- legal statements
- specific infrastructure commitments

Always be aware of the **risks of hallucinations and generalization** (instead of project specific information) when knowledge is lacking!

Two AI Chatbot services are available to members of the University of Hohenheim:

- **GPTalk** (→ https://gptalk.uni-hohenheim.de)
- **Academic Cloud ChatAI** (→ https://academiccloud.de/services/chatai/)

Both services rely on infrastructures operated by the GWDG:

- easy and data protection-compliant access to powerful generative AI
- different large language models (e.g., OpenAI GPT OSS 120B, Devstral 2 123B Instruct 2512, Mistral Large 3 675B 2512, etc.)
- alternative to commercial services like ChatGPT
- chat directly with a selection of different AI models via message, audio recording or text file
- aimed at teaching and studying to administration, research and personal development

Project organisation

There are various suggestions on how to organize all files related to a specific research project:

- The Turing Way
- GIN Tonic Research Team

However, you should have one directory per research project or manuscript that separates

- **Data** (raw and processed data, including metadata)
- **Code** (e.g., R code or shell scripts)
- **Outputs/Results** (i.e., regression output tables, figures, processed data)
- **Documents/Documentation** (related literature, manuscript(s), slides, electronic lab notebook recordings)

Most importantly, be clear and be consistent within your project!

1. Machine readable
   - avoid whitespaces (→ underscores, hyphens, camel case)
   - no special characters (ampersand ('&'), Dollar character, tilde, periods, question or exclamation marks etc.)
2. Human readable (→ descriptive names)
3. Optional: Control ordering by adding a prefix
   - Date (ISO8601 style formatting)
   - Numbering with leading zero (folder naming)
4. In workflows: Define a file naming convention

Source: XKCD, https://xkcd.com/1459/, accessed 2024-07-19.

The information stored in a file is typically encoded in a specific way which is referred to as file format.

Different file formats are used for different types of data (text documents, spread sheets, audio- and video, etc.)

The choice of file format(s) may become important in the context of:

- Exchanging data between different programs
- Data sharing and re-use
- Archiving and Long-Term Preservation

details about file formats

Documenting data

1. README file(s) in text or markdown format
2. Codebook
3. Structured descriptions (metadata), optionally following a discipline-specific metadata standard

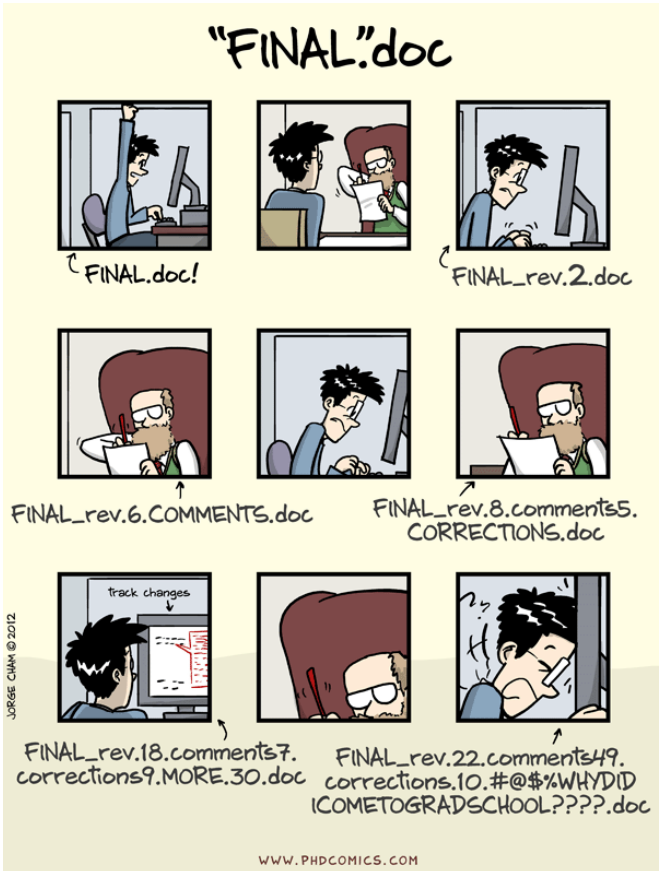
Documenting data

Data documentation depends on the specific research project but should cover the following aspects:

1. Study object
2. Analytical instruments, tools and methods
3. Data set structure

details about documenting data

- Track changes in code (and text) over time
- Indicating different versions of a file by adding v1, v2 etc. to its name may help in some situations but is not an adequate version control
- Use Git to track changes and add a remote repository
- Automate regular backup of all data



Source: Piled Higher & Deeper, https://phdcomics.com/comics/archive.php?comicid=1531, accessed 2026-03-06.

Campus data storage infrastructure (so-called 'project directory'):

- Storage capacity in the order of GBs (10 GB by default)
- Network file shares (CIFS/SMB or NFSv3)
- Share files with members of the University of Hohenheim
- Regular snapshots of the data
- Daily backup (retention period: one month)

Backup service (in cooperation with TIK at University of Stuttgart)

- IBM Storage Protect (operated by University of Stuttgart)
- Backup data from personal laptop, workstation, server
- Free (up to 2 TB per Department), 115 Euro/TB p.a.

SDS@HD – Scientific Data Storage (operated by University of Heidelberg):

- Storage capacity in the order of TBs (10 TB free)
- Network file shares (CIFS/SMB)
- WebDAV
- SFTP/SSHFS

bwSFS-2 – bwStorageForScience-2 (joint with TIK at University of Stuttgart)

- Network file shares (CIFS/SMB or NFSv3)
- Regular snapshots of the data
- Data is mirrored to TIK at University of Stuttgart
- Object-Storage (S3): global file transfer and cold data (joint with FR, TÜ, ST)
- Metadata: Management of metadata (structured documentation of research data)

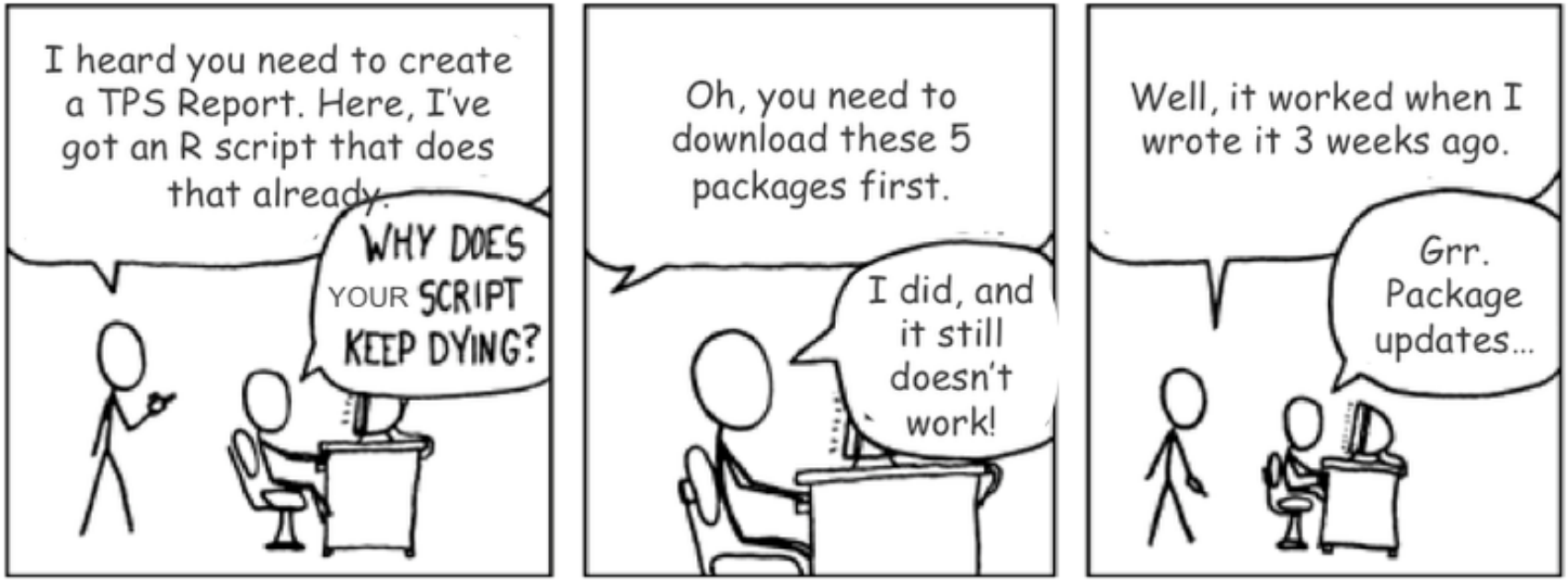
bwSync&Share (operated by KIT):

- Nextcloud-based online storage
- Up to 200 GB quota
- Online Office suite for collaborative work on text documents, spreadsheets and presentations (based on LibreOffice)

Microsoft Teams/Zoom-X

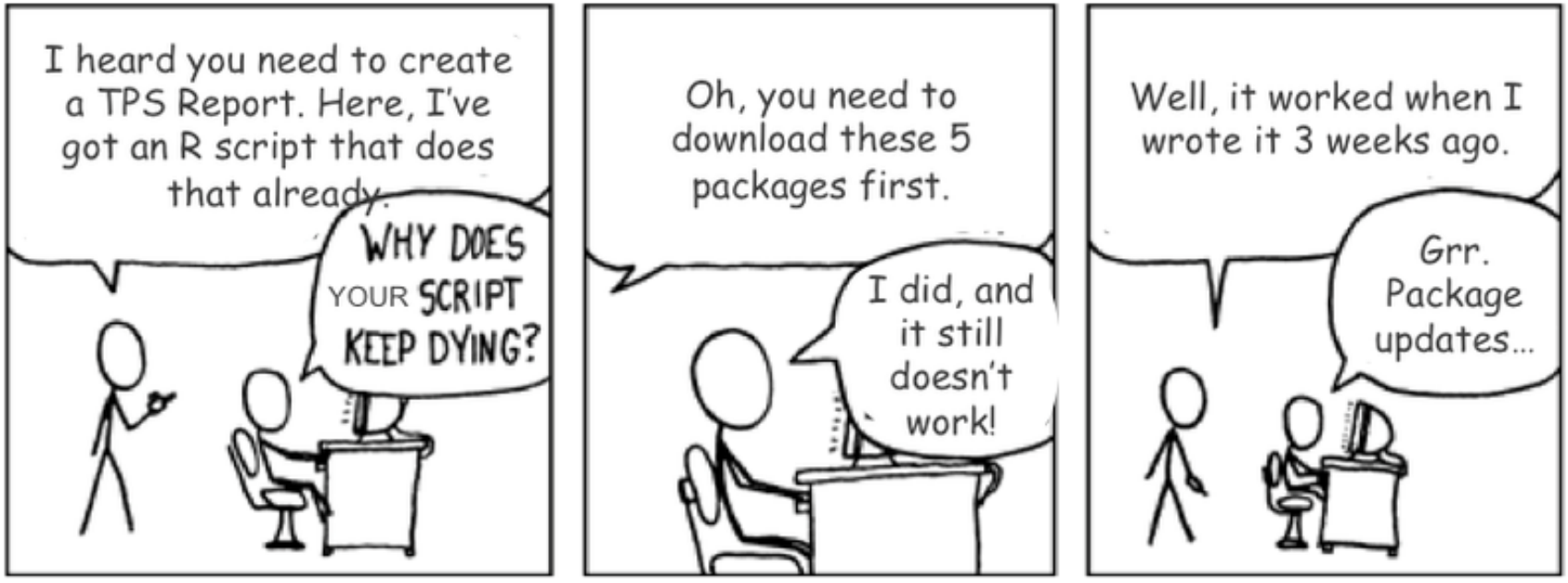
- Video conferences
- Team chat
- Channels

Reproducibility



Source: R-bloggers, http://www.r-bloggers.com/introducing-the-reproducible-r-toolkit-and-the-checkpoint-package/,

adapted from XKCD, https://xkcd.com/234/ accessed 2026-03-09.



Source: R-bloggers, http://www.r-bloggers.com/introducing-the-reproducible-r-toolkit-and-the-checkpoint-package/,

adapted from XKCD, https://xkcd.com/234/ accessed 2026-03-09.

→ Document in detail your code including all internal and external dependencies.

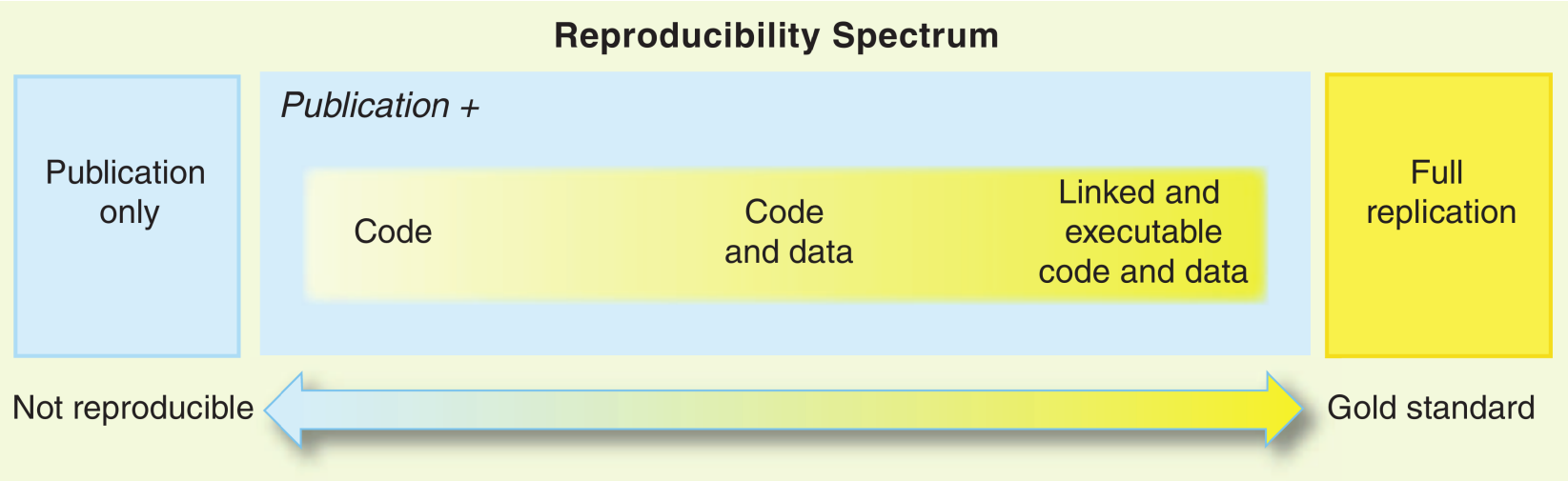
Not reproducible

- Data modified in a spreadsheet
- Click and point analysis
- Tables typed by hand
- Copy and paste graphs and tables to manuscript

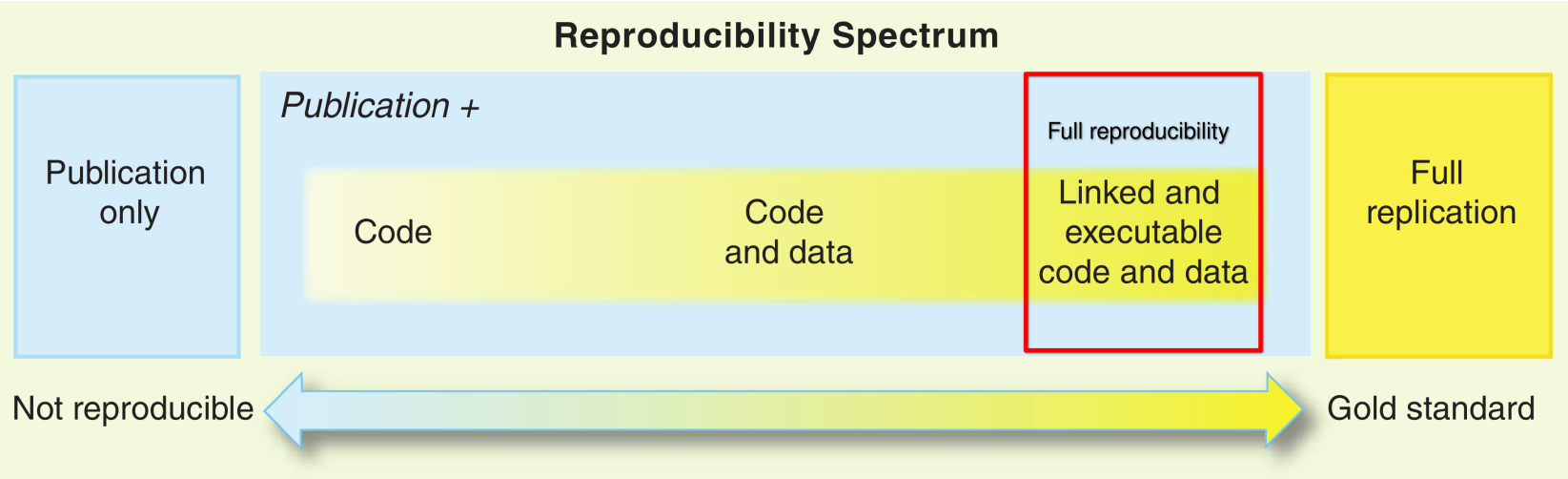
Reproducible

- All data modifications scripted
- All analyses scripted
- Tables generated with scripts
- Graphs and tables linked to manuscripts and reports

$\rightarrow$ Any manual step involved in data analysis is prone to end up not being reproducible.



Source: Peng (2011).



Source: Adapted from Peng (2011).

Depending on the **complexity** of the **analysis**, **full reproducibility** may require the ability to **recreate** the **computational environment** originally used to generate the scientific results.

Code and all dependencies need to be **bundled**:

- Programming environment level: virtualenv (Python), renv (R)
- Package managers: Conda (miniforge)
- Containerization: Podman, Apptainer/Singularity, Docker
- Virtualization (Virtual Machine, VM): VirtualBox, KVM (Linux)

Virtual machine (virtualization cluster)

- operated by KIM
- long-running computations
- limited parallelization
- OS: Windows or Linux VM

Compute Cluster (bwHPC)

- high-performance computing (HPC)
- parallel computations (multi-core & multi-node)
- need for GPU(s)
- high memory demand

bwJupyter

- central, web-based development environment
- need for GPU(s)
- for teaching only

- Clear and consistent organisation of all files related to a specific research project increase efficiency and facilitate collaboration
- Data management plans (DMP) help setting the framework that outlines how research data is handled throughout its lifecycle
  $\rightarrow$ depending on the complexity of the project, an online-tool can be useful
- Documenting research data supports your own work and is a pre-requisite for publishing and sharing
- Documenting code, including internal and external dependencies facilitates re-running an analysis and strengthens transparency with regard to scientific results
- Aim for improvement, not perfection

Thank you for your attention!

Anna Rudaev
Research Data Management Consultant

Garbenstr. 15, 70599 Stuttgart
2nd floor, room no. 245
Phone: +49(0)711-459-22621
e-mail: anna.rudaev@uni-hohenheim.de

Dr Konstantin Kuck
Research Computing Consultant

Garbenstr. 15, 70599 Stuttgart
2nd floor, room no. 247
Phone: +49(0)711-459-22085
e-mail: konstantin.kuck@uni-hohenheim.de

Contact us for consultations on data management and reproducible research computing.

References

Claerbout, J. F. and Karrenbach, M. (1992). Electronic documents give reproducible research a new meaning, *SEG technical program expanded abstracts 1992*, Society of Exploration Geophysicists, pp. 601–604.

Corti, L., Van den Eynden, V., Bishop, L. and Woollard, M. (2014). *Managing and Sharing Research Data: A Guide to Good Practice*, SAGE Publications.
URL: *https://books.google.de/books?id=-gg6AwAAQBAJ*

Eddelbuettel, D. (2021). Parallel computing with R: A brief review, *WIREs Computational Statistics* **13**(2): e1515.
URL: *https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1515*

Kuck, K. and Maderitsch, R. (2019). Intra-day dynamics of exchange rates: New evidence from quantile regression, *The Quarterly Review of Economics and Finance* **71**: 247–257.
URL: *https://www.sciencedirect.com/science/article/pii/S1062976918300322*

Peng, R. D. (2011). Reproducible research in computational science, *Science* **334**(6060): 1226–1227.
URL: *https://www.science.org/doi/abs/10.1126/science.1213847*

van de Wiel, H., Fraga Gonzalez, G., Furrer, E. and Held, L. (2023). Primer: File naming conventions.
URL: *https://doi.org/10.5281/zenodo.10091967*

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E. et al. (2016a). The fair guiding principles for scientific data management and stewardship, *Scientific data* **3**(1): 1–9.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E. et al. (2016b). The fair guiding principles for scientific data management and stewardship, *Scientific data* **3**(1): 1–9.

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L. and Teal, T. K. (2017). Good enough practices in scientific computing, *PLOS Computational Biology* **13**(6): 1–20.
URL: *https://doi.org/10.1371/journal.pcbi.1005510*

Appendix

File names, file naming conventions and directory structures

Organizing the files related to a project in **logical** and **consistent directory structure** allows to keep track of them.

There is no single best way to organise a file system. There are, however some guiding principles (Wilson et al., 2017).

Typically, it useful to have a **separate (project-)directory** for:

- every manuscript, or
- all research on a common topic or data set

Furthermore, it is useful to **separate** within a (project-)directory:

- **Data** (raw and processed data, including metadata)
- **Code** (e.g., R code or shell scripts)
- **Outputs/Results** (i.e., regression output tables, figures, processed data)
- **Documents/Documentation** (related literature, manuscript(s), slides, electronic lab notebook recordings)

back

```
.
├── .gitignore
├── CITATION.md
├── LICENSE.md
├── README.md
├── requirements.txt
├── bin                <- Compiled and external code, ignored by git (PG)
│   └── external       <- Any external source code, ignored by git (RO)
├── config             <- Configuration files (HW)
├── data               <- All project data, ignored by git
│   ├── processed      <- The final, canonical data sets for modeling. (PG)
│   ├── raw            <- The original, immutable data dump. (RO)
│   └── temp           <- Intermediate data that has been transformed. (PG)
├── docs               <- Documentation notebook for users (HW)
│   ├── manuscript     <- Manuscript source, e.g., LaTeX, Markdown, etc. (HW)
│   └── reports        <- Other project reports and notebooks (e.g. Jupyter, .Rmd) (HW)
├── results
│   ├── figures        <- Figures for the manuscript or reports (PG)
│   └── output         <- Other output for the manuscript or reports (PG)
└── src                <- Source code for this project (HW)
```

Figure: Example research project folder structure following the guiding principles outlined in Wilson et al. (2017).

This example project folder structure by Barbara Vreede is available under https://github.com/bvreede/good-enough-project.

```
.
├── AUTHORS.md
├── LICENSE
├── README.md
├── bin                <- Your compiled model code can be stored here (not tracked by git)
├── config             <- Configuration files, e.g., for doxygen or for your model if needed
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
├── docs               <- Documentation, e.g., doxygen or scientific papers (not tracked by gi
├── notebooks          <- Ipython or R notebooks
├── reports            <- For a manuscript source, e.g., LaTeX, Markdown, etc., or any project
│   └── figures        <- Figures for the manuscript or reports
└── src                <- Source code for this project
    ├── data           <- scripts and programs to process data
    ├── external       <- Any external source code, e.g., pull other git projects, or external
    ├── models         <- Source code for your own model
    ├── tools          <- Any helper scripts go here
    └── visualization  <- Scripts for visualisation of your results, e.g., matplotlib, ggplot2
```

Figure: Example research project folder structure following the guiding principles outlined in Wilson et al. (2017).

This example project folder structure by Mario Krapp is available under https://github.com/mkrapp/cookiecutter-reproducible-science.

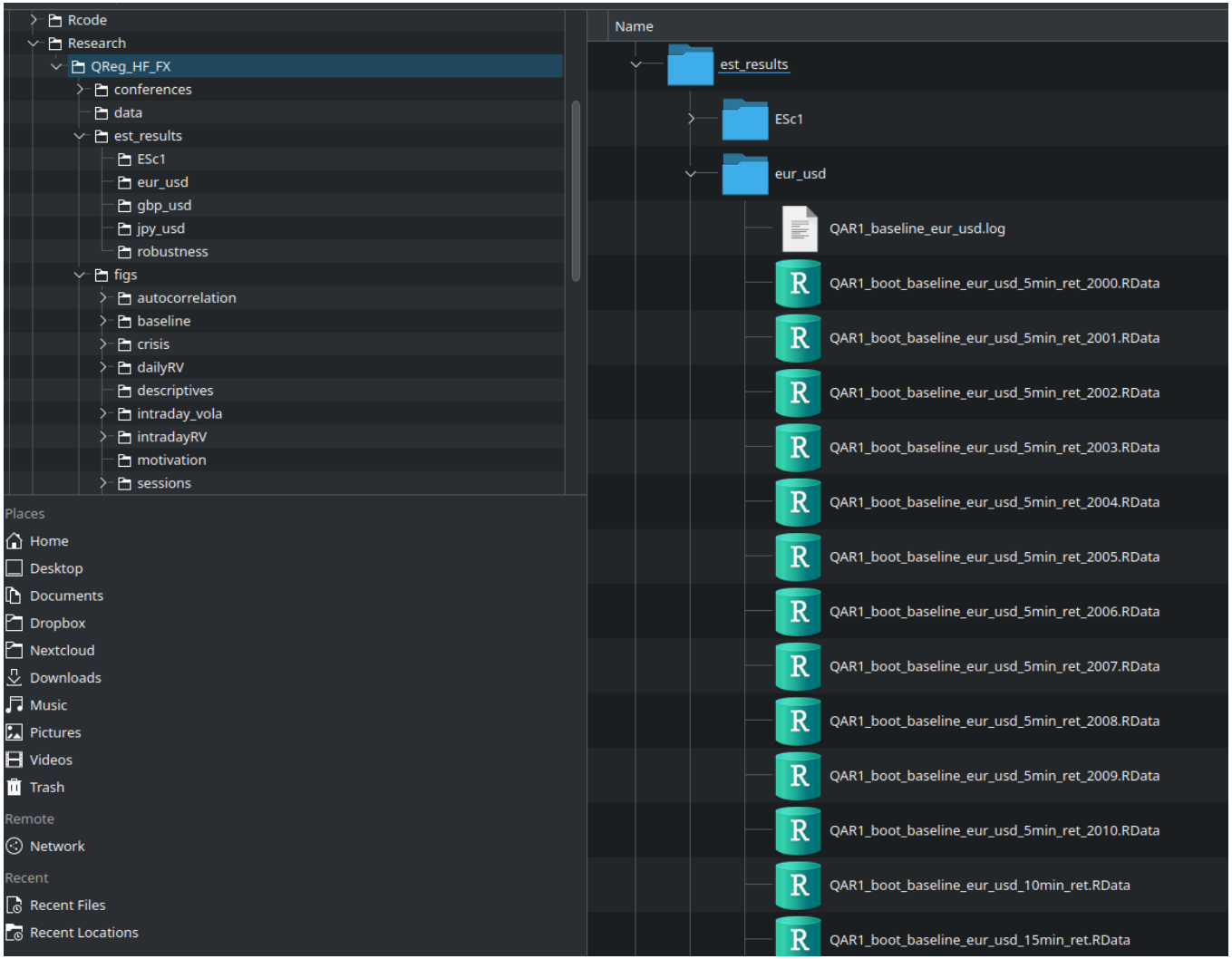


Figure: Research project folder structure used in Kuck and Maderitsch (2019).

A filename typically consists of a **base name** and (separated by a dot) an **extension**, indicating the file **format/type**.

File names should, generally, be **meaningful**, and allow to get an intuition about the contents of a specific file without opening it.

In that sense, the **file name** constitutes an important **metadata**.

From a technical perspective, file names should **not** contain **white spaces** and other **special characters** (e.g. the ampersand, Dollar character, tilde, periods, question or exclamation marks etc).

Furthermore, white spaces to separate different elements in a file name should be substituted by the "camel-case" approach, underscores or hyphens.

back

In addition, it is useful to follow the **general principles** summarized in van de Wiel et al. (2023):

- **Dates** within a file name should follow ISO8601 style formatting (`YYYY-MM-DD`, `YYYYMMDD` etc.)
- **Version numbers** can be included with a prefix and leading zeros (e.g., `v01, v02, ...`), to ensure consistency regarding the order of versions (particularly in the absence of a version control system)
- **Numbering** files sequentially should include leading zeros (i.e., `001, 002,..., 010, 011,...` instead of `1, 2, 3,...`) to ensure consistent file name width and correct ordering in file listings.
- **Relations between files**, e.g. input/output relations, can be indicated by adding appropriate suffixes (e.g., `_raw`, `_clean`, `_out`, ...)
- **File name length** should not exceed 250 characters (OS limitation) and file names should **generally be much shorter**.

If the calendar date is important metadata for a set of files, it should be added as prefix (formatted in ISO8601 style, i.e. `YYYYMMDD` or `YYYY-MM-DD`).

Incorporating the calendar date (of creation, or revision) into the file name ensures that this information (metadata) is preserved, regardless of the means of file transfers or other modifications.

Creation, modification and access timestamps (dates) are important metadata attached to a file (or folder) when stored in the file system (e.g. NTFS, EXT4, BTRFS etc.).

These timestamps are, however, not preserved when transferring a file by e-mail or upload it to S3.

It may be useful to implement a **file naming convention** in collaborative research projects, that defines how files of different file-types are named (van de Wiel et al., 2023).

Specifically, a file naming convention or file naming scheme explicitly **defines which** information about file contents is incorporated in the file name and how it is **structured**.

The implementation of a well-defined file naming convention also supports further **processing** of data in the context of **automated data analytical workflows**.

The Brain Imaging Data Structure (BIDS) describes data organization in the context of human brain imaging research.

It describes not only file names but also folder structure, file formats as well as meta data.

Specifically, a file name consists of a chain of entities and, separated by an underscore, a suffix and an extension.

The entities are a key (an abbreviated name) and a value, separated by a hyphen.

BIDS provides guidance but allows for flexibility and can be adjusted to the requirements of a specific research project.

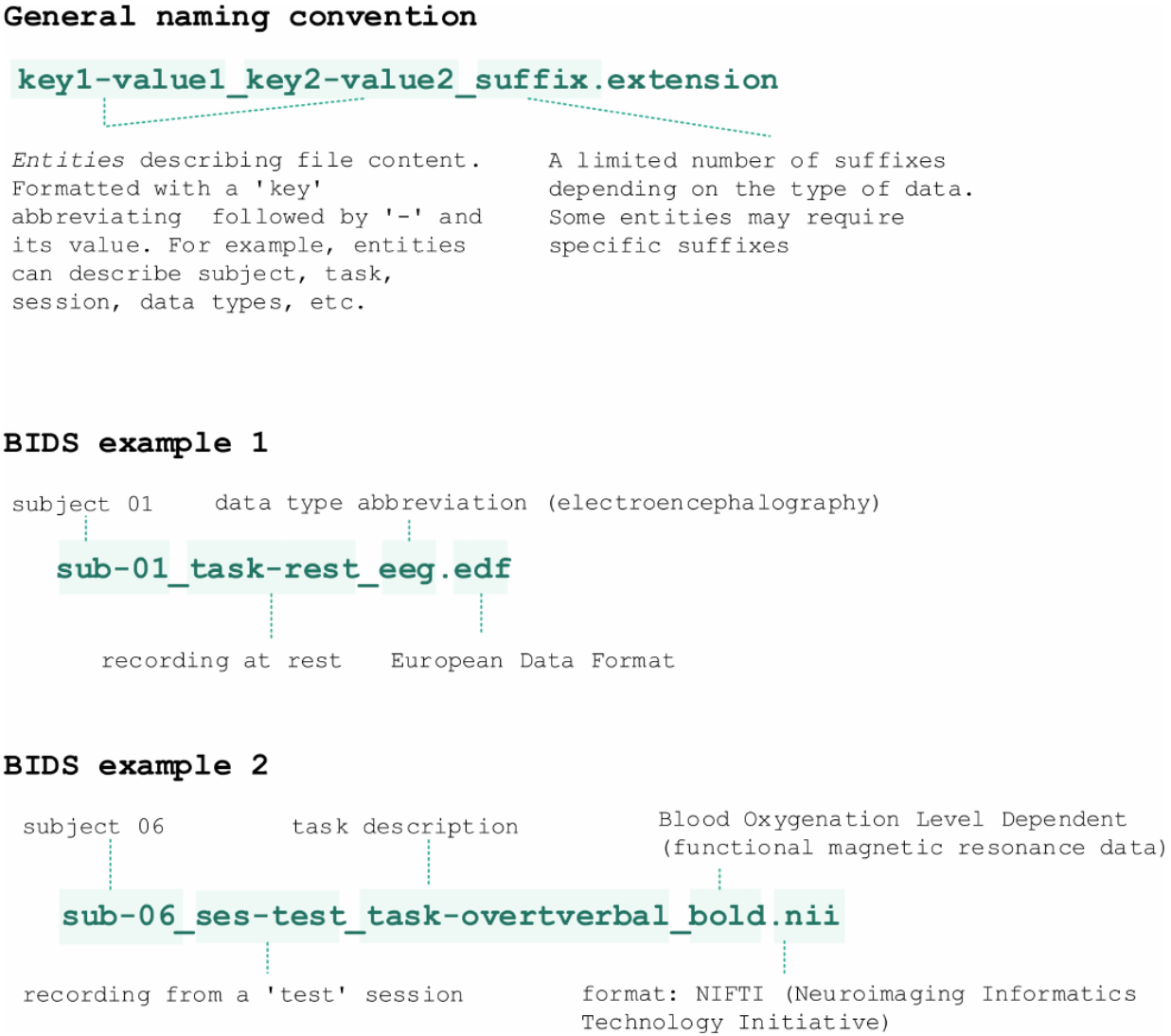


Figure: BIDS file naming scheme examples (van de Wiel et al., 2023).

Not all files within a project need to (and typically) follow a file naming scheme (file naming system). It is, however, best practice to employ meaningful file names.

Furthermore, it is best practice to keep some files (and folders), particularly, raw data read-only (ro) to ensure it remains unmodified.

For example, opening writable spreadsheets in Microsoft Excel may lead to unintended changes due (accidental) clicking in combination with the Auto-Save feature.

File formats

All information stored in the context of a file is encoded in a specific way which we refer to as **file format**, so that a certain computer program can read, interpret and modify the data.

Different file formats are use for different types of data:

- text documents: DOCX, ODT
- tabular data/spreadsheets: CSV, XLSX, ODS
- image data: TIFF, JPEG, RAW
- audio: WAV, MP3, OGG
- multimedia: AVI, MP4
- ...

back

The file formats used to store research data typically depend on the modalities of **data collection**, the **analytical instruments** involved, the **(statistical) analyses** carried out and the **software** used.

Particularly with regard to long-term preservation of the data, **choice of specific file formats**, should also be guided their specific characteristics (Corti et al., 2014):

- interchangeability versus program-specific (e.g., CSV/RData, CSV/SAS)
- loss-less versus lossy (e.g., RData/CSV, XLSX/CSV, DOCX/RTF)
- file size versus (content) quality (e.g., TIFF/JPEG, AVI/MP4)
- open versus proprietary (e.g., ODT/DOCX, GZ/RAR)

A detailed description of file formats can be found at https://www.loc.gov/preservation/digital/formats/fdd/browse_list.shtml.

From a data preservation point of view, the file format should satisfy the following characteristics:

- follow open standards (non-proprietary)
- well documented
- widely used (optimally supported by many tools)
- loss-lessy compressed or uncompressed

Different institutions issued recommendations on file formats for preservation:

- UK Data Service — Recommended formats
- ETH Zürich — File formats for archiving
- MPG — File formats

Data analyses often involve conversion from one file format to another, e.g., to access the data from within another software package (file export/import).

Particularly, if data is exported from a program-specific (proprietary) to another, interchangeable file format, it is important to account for potential information or data quality/precision losses involved in the conversion.

For instance, when exporting a data set from STATA (DTA file format) to a CSV, labeling of factor variables, missing value (NA) indicators and variable descriptions will be lost.

Importing data exported to an interchangeable file format, therefore, must be done very carefully.

Documenting research data

Data documentation should **provide all necessary details** that helps **understanding the data**.

The exact details to include may depend on the specific context but identification of the **relevant details** may be **guided** along the lines of the **following questions**:

1. **What is the study about?**
   - research question and objectives
   - theoretical background or hypothesis
   - population or phenomenon investigated
   - spatial and temporal scope
   - context of the data
2. **Who created the data?**
   - author(s) and institute(s)
   - contact information
   - funding source and project context
   - version history
   - experimental design
   - procedures

back

3. **How were the data generated?**
   - sampling strategy or experimental design
   - instruments or measurement tools
   - computational environment (software and package versions)
   - survey questionnaires, sensors, laboratory equipment
   - protocols and procedures
4. **How are the data structured and how should they be interpreted?**
   - file formats and dataset structure
   - units of observation
   - variable descriptions (codebook/data dictionary)
   - units of measurement
   - relationships between files or tables

back

5. **How were the data processed or modified?**
   - data cleaning
   - treatment of missing values
   - outlier handling
   - derived variables and transformations
   - scripts or software used for processing
   - quality assurance
6. **What are the access conditions and limitations?**
   - licensing and access restrictions
   - ethical considerations and anonymization steps
   - data protection issues
   - known limitations or biases
   - recommended citation

back

- **README file(s) in text or markdown format**
  - all information relevant to understand the data (i.e., project information, details about the data generating processes, software and instruments)
  - separate README files for files/folders that are logically grouped together
  - keep README files consistent within a research project
  - use and adapt a template, e.g., Template by Cornell University
- **Codebook**
  - variables
  - factor codes/levels
- **Structured descriptions (metadata)**
  - e.g. in a spreadsheet or metadata database
  - optional: following a discipline-specific metadata standard
  - aimed at machine-readability
- **File metadata**
  - file generating instrument
  - parameters
  - native file metadata

back

Document research data within different software packages:

- Microsoft Excel/LibreOffice Calc: separate sheet within workbook (the sheet(s) containing the actual data should only contain the data columns)
- STATA: the **built-in Variable Manager** allows to describe variables and set value labels
- R: attributes can be assigned to objects
- ArcGIS geodatabase: metadata can be recorded in the ArcCatalog for shapefiles, layers and tables in a geodatabase

Metadata can also be stored along with the primary data using a container file format like HDF5 or NetCDF.

Version control

- Keep track of changes in files over time
- Review any changes and revert back
- Test changes in local branches (isolated local copies) without affecting the main branch
- Set up an online repository to sync tracked files across computers or work collaboratively (e.g., GitLab)
- Collaborative projects: track what changed and who made changes
- Limited change-tracking is also available in office suites (Microsoft Office, LibreOffice, Google docs)
- Git should not be used in the context of large (binary) files that change often

Further information about the use of Git in the context of research can be found in The Turing Way.

1. Create files (plain text files, e.g. code, markdown, LaTeX)
2. Work on these files, by changing, deleting or adding new content
3. Create a snapshot (Git: commit) of the file status (also known as version) at this time
4. Document what was changed in the version history of that file (commit message)

- Branch: local copy of the main repository
  - allows to try new changes, e.g. adding a new feature
  - isolated from main branch
- You can have more than one branch from the main, or create branches from branches
- Changes made in branches can be merged to the main (usually a manual process)