

'4th Online GEO-Inquire FAIR Training: reproducible code'

Stefano Rapisarda (RDM, Utrecht University Library)

2024-11-26

Towards FAIR software: Replicability and Reproducibility

During the workshop we emphasized that reproducibility is a core principle of science. In this context, providing readable, clean, and robust code is essential. Simply having code that runs doesn't guarantee good scientific software.

Example of Project Structure

Recommended project structures for Python and R:

Python Project Structure

my_python_package/

```
.gitignore          # Specifies intentionally untracked files to ignore
LICENSE             # Contains the project's licensing information
README.md           # Project documentation and overview

data/               # Data storage directory
  processed_data/   # Processed and cleaned data files
  raw_data/         # Original, unmodified data files

docs/               # Additional documentation files

notebooks/          # Jupyter notebooks or analysis scripts

requirements.txt    # List of Python package dependencies
```

```

src/                # Source code directory
  my_python_package/ # Package source code
    __init__.py     # Package initialization file
    main.py         # Main package script

tests/              # Unit tests and test scripts

```

R Project Structure

```

my_r_package/

.Rproj              # R Project configuration file
.gitignore          # Specifies intentionally untracked files to ignore
DESCRIPTION         # Package metadata and dependencies

R/                  # R source code directory
  main.R            # Main script
  my_functions.R   # Custom function definitions

README.md           # Project documentation and overview

data/               # Data storage directory
  processed/       # Processed and cleaned data files
  raw/             # Original, unmodified data files

notebooks/         # R Markdown or analysis notebooks

tests/              # Testing directory
  testthat/        # Test cases
  testthat.R       # Test runner script

```

Key recommendations for Licensing

- Reflect on potential users
- Provide license information in the repository
- Use resources like choosealicense.com
- Consult your Department's Research Support Office

README File Best Practices

A README should:

- Be the first thing users see
- Explain the project's purpose and motivation
- Provide getting started instructions
- Offer support contact information
- Indicate project maintainers

Writing FAIR and Clean Code

Code quality guidelines:

- Use strategic whitespaces and newlines to enhance readability.
- Choose descriptive names for functions and variables to convey intent.
- Maintain a consistent coding style across the codebase.
- Assign single, well-defined purposes to functions and classes.
- Implement unit testing to validate functionality and catch bugs early.
- Encourage peer review to identify issues and share knowledge.
- Document key functions and modules to improve maintainability.
- Keep dependencies minimal and well-organized.

Software Archiving and Citation

Recommended citation hierarchy:

1. Cite the publication
2. If no publication, cite the DOI (from Zenodo or Figshare)
3. If no DOI, cite the GitHub repository
4. Ideally, use all three methods

Git tagging process:

- Use Semantic Versioning (vX.Y.Z)
 - Major (X): Breaking changes
 - Minor (Y): Backwards-compatible features
 - Patch (Z): Bug fixes
- Use suffixes like `-beta` or `-rc` for pre-releases

Publishing on Zenodo

1. Create a public GitHub repository
2. Create a software release
3. Log in to Zenodo with GitHub credentials
4. Select repository to archive
5. Obtain a persistent identifier (DOI)

Web Resources

FAIR and Research Software

- [FAIR Principles for Research Software](#)

Packaging Guides

- [Python Guide to Packages](#)
- [R Guide to Packages](#)

Coding style guides

- [Python coding style guide](#)
- [R coding style guide](#)

Writing a good README file

- [Readme files GitHub guidelines](#)
- Example of ASReview README file, scroll down the [ASReview GitHub repository](#)

Licensing

- [Open Source License Catalogue](#)
- [Choose a License Tool](#)

Repository Management and Publication

- [Assigning a DOI to Your Repository](#)
- [GitHub README Files Guidelines](#)

Additional Resources

- [The Turing Way](#)
- [Utrecht University Python Project Template](#)
- [Utrecht University R Project Template](#)