Creating a package with Python and R

Raoul Schram and Jelle Treep

February 16, 2024

Utrecht University

Packaging: why?

- Share your code with other researchers more easily
- Distribute through CRAN (R) or PyPi (Python)
- Reuse your own code
  - Project A uses functionality of project B
  - Copy-paste the functionality?
  - Package project B and simple import!
- Do not have to worry about where your code is
- Metadata added to your code

Utrecht University

Packaging: when?

- Your choice!
- If my code becomes significant ($>$ 100 lines?)
- If reuse is expected
- Library kind of code
  - Reuse as part of your code
  - Functionality that is independent of a particular dataset
- Application type of code
  - Reuse as whole of your code
  - Command Line Iterface

Utrecht University

Packaging: Python

"There should be one – and preferably only one – obvious way to do it." - Zen of Python

- Setuptools
- Poetry
- Hatch
- PDM
- Flit
- uv

Utrecht University

Packaging: Python

"There should be one – and preferably only one – obvious way to do it." - Zen of Python

- Setuptools
- Poetry
- Hatch
- PDM
- Flit
- uv

Utrecht University

Packaging: Python

"There should be one – and preferably only one – obvious way to do it." - Zen of Python

- Setuptools
- Poetry
- Hatch
- PDM
- Flit
- uv

Utrecht University

Packaging: Python

"There should be one – and preferably only one – obvious way to do it." - Zen of Python

- Setuptools
- Poetry
- Hatch
- PDM
- Flit
- uv

Utrecht University

Packaging: Python

"There should be one – and preferably only one – obvious way to do it." - Zen of Python

- Setuptools
- Poetry
- Hatch
- PDM
- Flit
- uv

Utrecht University

Packaging: Python

"There should be one – and preferably only one – obvious way to do it." - Zen of Python

- Setuptools
- Poetry
- Hatch
- PDM
- Flit
- uv

Utrecht University

Packaging: Python

"There should be one – and preferably only one – obvious way to do it." - Zen of Python

- Setuptools
- Poetry
- Hatch
- PDM
- Flit
- uv

Utrecht University

Setuptools versus Poetry

Setuptools

- Adheres to PEP621 and PEP631 (pyproject.toml)
- Utilities when working with C/C++ code
- Lean

Poetry

- Lock files for reproducibility
- Elaborate CLI (e.g. adding new dependencies)
- Good documentation
- Automatic virtual environments

Utrecht University

The (standard) pyproject.toml file

- TOML: Tom's Obvious Minimal Language
  - Easy to read by humans and computers
  - Expressive
- Contains metadata for your package
  - Name of the package
  - Version
  - Authors
  - How to build it
  - Where the package is

```toml
[build-system]
requires = ["setuptools>=45", "setuptools-scm[toml]>=6.2"]
build-backend = "setuptools.build_meta"

[project]
name = "ibridges"
authors = [
    { name = "Christine Staiger", email = "c.staiger@uu.nl" },
]
description = "Package for accessing data and metadata on iRods servers."
readme = "README.md"
requires-python = ">=3.8"
```

Python packaging: extra tips

- Package modules in directory named after the package
- Packages can have subpackages
- Each package and subpackage needs an `__init__.py`
- Take care of non-python files (`MANIFEST.in`)
- Type hinting needs a `py.typed` in package directory

Utrecht University

R packaging: Jelle Treep

Templates

Poetry:

`https://github.com/UtrechtUniversity/re-python-package-poetry`

Setuptools:

`https://github.com/UtrechtUniversity/re-python-package`

Utrecht University