Teaching how to make reproducible Python environments for science

Samuel Grayson*

* **Department of Computer Science, University of Illinois at Urbana-Champaign**

For ENG 598 T

Motivation

- Chemistry, astronomy, and other research disciplines increasingly use computational methods, but may not have CS background [1].
- Making software reproducible is hard [2].
- Science has to be reproducible [3].
- Therefore, we designed this course to explain how to make reproducible software reproducible to a scientific audience.
- Scientists often use Python, so we limit the scope to Python without much loss.

Course objectives

Students will be able to:

- Define terms related to reproducibility (ACM definitions)
- Specialize the terms for software experiments
- Explain why packaging software for Python is difficult
- Explain the pros and cons of Pip, Conda, Docker, and Nix
- Be able to use Conda or Nix in a practical setting

Prior work

- **Software Carpentries** [4]: targetting practicing researchers, teaching computing skills that improve efficiency, distributed by volunteers at conferences and online
- **The Turing Way** [5]: targetting data scientists, teaching reproducible and ethical research, distributed as a book
- **Barba reproducibility reading list** [6]: targetting grad students, teaching how and why to make research reproducible, distribtued as a research paper reading list

This work is most similar to software carpentries

Delivery mode

2 delivery modes:

- 60 minute, slideshow, workshop tutorial, with interactive questions, hands-on example, 1 instructor and 1 floating assistant instructor
- 20 minute read, blog post, with interactive questions questions, no instructor

This is similar to Software Carpentries as well [4].

Course outline

- Introduction
  - Objectives
  - Prerequisites
  - Expecatations
- Reproducibility terms
  - Repeatability, reproducibility, replicability
  - How to specialize metrology for software
    - What does “measurement” mean?
    - What are “operating conditions”?
  - Related terms
    - What are “package managers”?
    - From-source vs binary package managers?
    - What are “software dependencies”?
    - What is a “diamond dependency”?
    - When do we need or not need a “dependency solve”?
- Ecosystem of strategies for reproducible environments
  - How do Python packages work?
  - Why is packaging for Python hard?
  - How to specify the environment for Jupyter notebooks?
  - How does Pip work?
  - How does Conda work?
  - How does Spack work?
  - How does Nix work?
  - How does Guix work?
  - How do containers work?
  - A note on pseudo-random number generators
  - A note on data
- Our recommendations
  - How to use Nix
  - How to work with other package managers

Measurement

- Before and after examination
  - Do learners know more about the theoretical concepts?
- Usefulness assessment
  - Do learners feel this actually useful?
- Time on page (online version)
  - Do learners engage with the online text?
- Online comments (thematically coded)
  - What feedback can we get from the community?

Community-building

- Publicize in forums, especially research software engineering groups
- Accept contributions to the curriculum by pull requests on GitHub, similar to Software Carpentry [4] and The Turing Way [5].

References

[1]: Simon Hettrick. Software in Research Sruvey. 2018. Zenodo. https://doi.org/10.5281/zenodo.1183562

[2]: Samuel Grayson, Darko Marinov, Daniel S. Katz, and Reed Milewicz. 2023. Automatic Reproduction of Workflows in the Snakemake Workflow Catalog and nf-core Registries. In Proceedings of the 2023 ACM Conference on Reproducibility and Replicability https://doi.org/10.1145/3589806.3600037

[3]: Robert Merton. The sociology of science: Theoretical and empirical investigations. 1973. University of Chicago Press

[4]: Greg Wilson. Software Carpentry: lessons learned. F1000Res. 2014. https://doi.org/10.12688/f1000research.3-62.v2

[5]: Multiple authors. The Turing Way. https://the-turing-way.netlify.app/index.html

[6]: Lorena Barba. Barbagroup reproducibility syllabus. https://lorenabarba.com/blog/barbagroup-reproducibility-syllabus/

ILLINOIS