EVERYDAY REPRODUCIBILTY

COURSE OUTLINE

- Reproducibility and its Goals
- Code Notebooks
- Reproducible Programming
- Version Control
- Containers
- Putting everything together

JSU SCHEDULE

All session in Room 147

1. **Introduction, Notebooks, Programming**

Sunday July 28th, 1:15 PM – 3:15 PM

2. **Working session** – BRING A LAPTOP

Tuesday July 30th, 10:15 AM – 12:15 PM

3. **Version Control, Containers, All Together**

Tuesday July 30th, 1:30 AM – 3:00 PM

4. **Working session** – BRING A LAPTOP

Wednesday July 31st, 3:15 PM – 5:15 PM

OVERVIEW: REPRODUCIBILITY AND ITS GOALS

OUTLINE:

- What is “reproducibility”?
- Goals of reproducibility
- Discussion

REPRODUCIBILITY…

… and other words that start with “R”

- reproducibility
- replicability
- repeatability
- robustness
- rigor

Our focus: computational reproducibility

REPRODUCIBILITY: GOALS

What are we really trying to achieve?

- **Reproducible:** Entire analysis fully and exactly reproducible
- **User friendly:** Easy to access, install, run
- **Transparent:** Easy to inspect, understand, modify
- **Reusable:** Others may build upon the project
- **Version controlled**
- **Archived**

FULLY AND EXACTLY REPRODUCIBLE

- The “most original” data should be available
- Include all code necessary to get from the original data to the final results
- The code should directly produce the plots / tables / numbers in the paper
- All software dependencies should be specified and ideally included with the code
- Random seeds specified
- etc.

USER FRIENDLY

- Code easy to access and inspect, ideally even without downloading
- Should require minimal effort for a user to install and run
- Should cause minimal disruption to a user’s resources (e.g., not install unwanted software on their system)
- etc.

TRANSPARENT

- Code should be organized and well documented, ideally in a notebook format
- Analytical choices, such as statistical tuning parameters, should be clearly highlighted
- Interactive elements such as widgets should be used when appropriate to help users explore the impact of different analytical choices
- Results-caching should be used so that users can quickly re-run specific parts of the analysis, perhaps after making minor modifications
- Both raw data and cleaned / re-formatted data should be made available when appropriate, e.g., when the raw data is difficult to use or understand without additional processing
- etc.

REUSABLE

- Code should be portable across platforms
- Code should be modular to facilitate re-use in other project
- Depending on the project, creating a new software package may be helpful
- etc.

PERMANENTLY ARCHIVED

- In a (file) format suitable for long-term preservation
- in a (physical) format suitable for long-term preservation

VERSION CONTROLLED

- This aids transparency
- Ultimately, most valuable for **you**

GOALS

1. Exactly reproducible
2. User friendly
3. Transparent
4. Reusable
5. Archived
6. Version controlled

These are distinct goals

They pose distinct challenges

FOCUS: *EVERYDAY REPRODUCIBILITY*

- Most of our goals are readily achievable for “everyday” projects
- Hard challenges we won’t discuss:
  - restricted access datasets
  - massive datasets
  - proprietary software
  - highly computationally intensive code

DISCUSSION

- Which reproducibility goals have you attempted on a project? Which do you regularly strive for?
- What challenges have you run into?
- Any interesting / notable experiences?
- What lessons have you learned?

DISCUSSION

What tools do you find helpful?

SOME USEFUL TOOLS

- notebooks / markdown
  - R Studio, Jupyter
  - jupyter notebooks, quarto notebooks,
  - jupytext and quarto
- git
- dependency management
  - Docker
  - `venv` (python virtual environments)
  - `renv` (R virtual environments)

SOME USEFUL TOOLS

- R and python packages
- results caching
  - makefiles
  - pickle
  - etc
- output formatting
  - kable
  - xtable
- sharing
  - github
  - zenodo