Everyday Reproducibilty

Simple Flexible Tools for Making Analyses more Accessible and Reproducible

Schedule

- Introductions (5 min)
- Overview: Reproducibility and its Goals (25 min)
- Code Notebooks (45 min)

  Break (10 min)
- Everyday Practices for Reproducible Programming (45 min)
- Version Control (20 min)

  Break (10 min)
- Containers (25 min)
- Putting everything together (25 min)

  Break (10 min)
- Discussion (20 min)

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

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?

Reproducibility: goals

What are we really trying to achieve?

- **Reproducible:** Entire analysis fully and exactly reproducible

Reproducibility: goals

What are we really trying to achieve?

- **Reproducible:** Entire analysis fully and exactly reproducible
- **User friendly:** Easy to access, install, run

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

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

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**

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

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 + Rmarkdown
    - Jupyter + Jupytext + markdown
- git
- dependency management
    - Docker
    - `renv`
    - python virtual environments

Some useful tools

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