Teaching software best practices to scientists.

Chaya D. Stern

PyData NYC

Why this course?

Computers are essential to all branches of science but most scientists do not receive software training.

Topics we are covering

1. Version control
2. Data management
3. Clean code and documentation
4. Environment management and package management
5. Testing

Version control

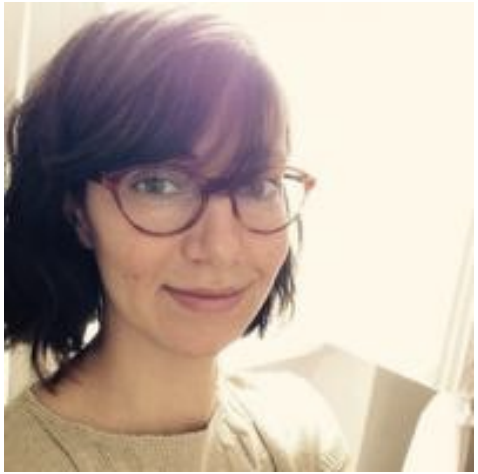
- Git and Github
- Create, fork and clone repos
- Add and commit changes
- Using branches
- Reverting commits

Reshama Shaikh

Data Management

Organizing, documenting , automation and dissemination of research

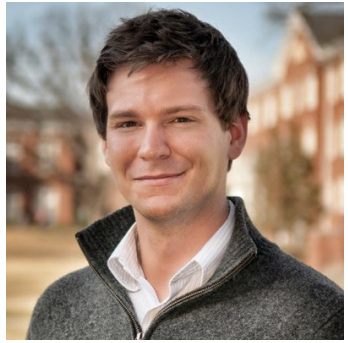
- Data collection
- Repository organization
- Configuring run environments
- Documentation
  - Specifying dependencies
  - Creating README
  - Creating data dictionaries
- Automating with master script
- Disseminating your code



April Clyburne-Sherin
Code Ocean
Director of scientific outreach

Clean code and documentation

- Writing readable software
- PEP8 style guide for Python
- Automating function and class documentation with sphinx



Daniel Smith
MolSSI Software Scientist

Environment and package management

Software packaging from three perspectives

- Users:
  - Common package managers and how to use them
  - Best practices when using them to avoid headaches
  - Virtual environments
- Maintainer
  - How to know if code is ready for packaging
  - Packaging code
  - Keeping packages up-to-date
- Backend engineer (extra)
  - How does this all work?

CJ Wright
Columbia University
Conda-forge core developer

Unit testing

- Principles of unit testing
- Writing unit tests
- How to determine what unit tests your code need
- What to do with hard-to-test code
- Limits of unit testing in the sciences

Jane Adams
Two Sigma

Questions and Feedback