Software Best-Practices for Reproducible Science

Course description:
This course will provide scientists with the right tools to implement software best-practices in their work. During the sessions, you will learn about concepts such as version control, testing, documentation, packaging and sharing code. The goal is not only to introduce these concepts, but also provide enough practical training for attendees to start using it in their work.

Who is this course for?
MSK scientists (grad students to faculty) who have some coding familiarity; you must know about for and while loops and might have written some scripts to analyze data.

Course director:
Chaya Stern

Date and Time:
Lecture 1: Github and version control Thursday November 8, 9:30-12:30
Lecture 2: Data Management Monday November 12, 9:30-12:30
Lecture 3: Clean Code and Documentation Thursday November 15, 9:30-12:30
Lecture 4: Packaging/ Environment Monday November 19, 9:30-12:30
Lecture 5: Testing Tuesday November 27, 2:00-5:00

You must commit to attending all 5 lectures. You should have your own laptop with Python 3 and Git installed before the course begins.
Follow instructions here to install Git:
https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
Installing Python – You should install Python 3:
https://wiki.python.org/moin/BeginnersGuide/Download

Session descriptions:

Session 1: Version Control:
Reshama Shaikh
This session will cover Git, the open source version control system for storing you programs
The following topics will be covered:

- Introduction to Git and GitHub; explain the difference between the two
- Introduce GitHub website; review setting and options on GitHub Account
- Create a repo on GitHub website
- Set up a repo on GItHub and invite collaborators
- Fork and clone user or organization repo
- Mark changes and track them using git
- Use branches on GItHub (*time permitting)
- Undo changes – revert commits (*time permitting)

Session 2: Data Management
April Clyburne-Sherin
This session will develop your skills in organization, documentation, automation and dissemination of your research. The following topics will be covered:

- Data collection
- Repository organization (separate code and data)
- Configuring run environments
- Documentation:
    - Specifying dependencies
    - Create a README
    - Creating data dictionaries
- Automation:
    - Creating a master script

    - Creating relative paths
- Dissemination
    - Specifying a license
    - Publishing your code

Session 3: Clean Code and Documentation
Daniel Smith
This session will cover code documentation and best practices in code style.

The following skills will be covered:
- The PEP8 style guide for Python.
- The YAPF formatter for automatic PEP8 implementation.
- Beyond style guides, tips for writing readable software
- Getting started with Sphinx documentation.
- Automatic function and class documentation
- Sphinx shortcuts and usage ideas.

Session 4: Package management and Environments
Christopher J. Wright
Software packages are the most common way to distribute and install software in the scientific/data/developer fields. Gone are the days of shipping CDs and floppy disks of software around the globe or downloading source code and hoping that everything is compatible. Package managers can have new software on your machine in seconds, and automatically keep everything up to date and compatible with very little human intervention. However, using packages and making packaging can be a bit of an art.

In this session I will discuss software packaging from three perspectives: users, maintainers, and backend engineers.
The user's perspective will focus on:
- What are some of the common package managers and what do they do
- How do we use package managers to get software and keep it up to date
- What are some best practices when using a package manager to avoid headaches

The maintainer's perspective will focus on:
- How do I know if my code is ready to be packaged
- How do I package my code
- How do I keep my packages up to date

The backend engineer's perspective (time permitting) will focus on:
- How this all works under the hood
- What are some of the fronters for packaging and how do they impact user and maintainer experience

Finally we'll make packages of our own (if anyone has code ready to be packaged).

Session 5: Testing you code
Jane Adams
Scientists are always hearing that they should be testing their code, but rarely do they hear what that would actually look like. In this session, we will introduce the principles of unit testing, and outline the major assumptions and consequences of these principles. You will learn how to write unit tests, as well as how to determine what unit tests your code needs and what to do with hard-to-test code. You will get hands-on experience writing unit tests using Python's unittest library, and learn about additional tools and best practices that you can adopt to efficiently incorporate unit testing into your everyday coding workflow. We will also discuss the limits of unit testing in the sciences specifically and discuss alternative testing approaches and libraries that can handle unique scenarios like stochasticity.

Course Instructors bio

1) *Reshama Shaikh*

Reshama Shaikh is a freelance data scientist/statistician in New York City. She worked for over 10 years as a biostatistician in the pharmaceutical industry. She is an organizer of the meetup groups NYC Women in Machine Learning & Data Science and PyLadies. She received her M.S. in statistics from Rutgers University and her M.B.A. from NYU Stern School of Business. Twitter: @reshamas
LinkedIn: https://www.linkedin.com/in/reshamas/
Blog: https://reshamas.github.io
GitHub: https://github.com/reshamas

2) *April Clyburne-Sherin*, Code Ocean

April is an epidemiologist, methodologist and expert in open science tools, methods, training and community stewardship. She holds an MS in Population Medicine (Epidemiology). Since 2014, she has focused on creating curriculum and running workshops for scientists in open and reproducible research methods and is co-author of FOSTER's Open Science Training Handbook. She is currently the Director of Scientific Outreach for the reproducibility platform Code Ocean.

3) *Daniel Smith*, The Molecular Science Software Institute, Virginia Tech

Daniel is a Software Scientist at the Molecular Sciences Software Institute, a nexus for science, education, and cooperation serving the worldwide community of computational molecular scientists. He traditionally writes software for open-source, high-performance quantum chemistry, but has recently begun to develop community-scale quantum chemistry databases for machine learning, biomolecular forcefield fitting, and novel method accuracy assessment.

4) *Christopher J. Wright*, Columbia University

Christopher grew up in Rockville Centre, New York. He attended Brown University where he earned a BS with honors in Chemical Physics. He worked with Prof. Shouheng Sun on electrochemical CO2 reduction and the structural dynamics of nanoparticles. After working at Brookhaven National Laboratory as a summer intern, he attended the University of South Carolina, working with Prof. Xiao-Dong Zhou on the atomic structure of solid oxide fuel cell components and earned a masters in Chemical Engineering. He joined the Billinge Group in 2016 and is currently working on analysis pipelines, data processing techniques and simulation software for PDF. Christopher is also a core developer for the conda-forge packaging ecosystem, president of Columbia qSTEM and sits on the Columbia Foundations for Research Computing advisory board. In his free time Christopher develops for open source projects, plays woodwind instruments, enjoys good scotch and works out by playing squash.

5) *Jane Adams*, Two Sigma

Jane Adams is a Data Scientist at Two Sigma Investments where she leads data quality strategy for Data Engineering, and she is the co -creator and -maintainer of marbles, an open-source Python testing library that gives your tests beautiful failure messages. She holds an undergraduate degree from NYU in Complex Systems, and a masters degree from NYU in Urban Data Science. Throughout her career, she has worked on both data science and engineering teams, and is always looking for new ways to share skills (or steal ideas) and improve communication across disciplines. She has traveled around the world to talk about topics ranging from how ants find your picnic basket to how to not accidentally hurt people with data. You can watch her talk about unit testing data at PyData Amsterdam here.