UNIVERSITY of WASHINGTON

DATA 515A

Course introduction & Syllabus

Bernease Herman

January 5, 2021

Agenda

1. Introductions
2. Course and syllabus review
3. Project and homework discussion
4. ***[Other slides]*** Introduction to Software Engineering

Who are we?

Bernease Herman, instructor.

Mark Friedman, instructor.

Priyanka Saraf, teaching assistant.

Yash Raichura, teaching assistant.

Introduction to class and virtual technology

What is class about?

Objectives for the class:

- Teach how to create and collaborate on data- and computation-intensive research projects
- Provide practical software skills for data analysis in research and industry
- Elevate coding in science to the level of technical writing

Programming vs Software Engineering

Analogy: What is the difference between the following kinds of writing?

1. Note to yourself
2. An article in the NY Times

~~**Why data science?**~~

Why software engineering practices in data science?

Writing clean readable code helps understanding of your code (including yourself), preventing bugs and issues, collaborating with others, and becoming a better professional data scientist or technologist.

What technical skills are taught?

- Program in python using the Python scientific stack, including `numpy`, `pandas`, and `matplotlib`.
- Search, evaluate, and integrate into a project externally developed Python packages; create your own Python packages.
- Develop unit tests that validate important aspects of the project implementation.
- Develop software that it can be used by others including: shared code on github, documentation, installing packages, setup, and running computational studies.
- Create technical specifications for what a program should do and how this is accomplished.

Course structure

- Software Engineering Skills (~4 weeks)
  - *Version control, OOP, software design, debugging, testing, exceptions*
- Python Packaging & Tools (~2 weeks)
  - *Imports, packages, PyPI, style & documentation*
- Data science enrichment (~2 weeks)
  - *Cloud, Kubernetes, probabilistic software*
- Project work & Presentations (~2 weeks)
  - *Technology reviews, standups, package review*

Class structure

- Discussion (30 min - 1 hour)
  - *Open Q&A, Homework discussion, [Standups]*
- Shorter break (5 - 10 min)
- Lecture (1 - 1.5 hours)
  - *Slide deck, instructors*
- Longer break (10 - 20 min)
- Lab (1 - 1.5 hours)
  - *Jupyter notebook or demonstration, TAs*

Technology we'll use

- Course information
  - Found on Canvas
  - Contains recordings (Panopto) from lecture
- Zoom video conferencing software
  - Chat (does not persist)
  - Breakout rooms
- Ed
  - Discussion tool (does persist)
- Gradescope
  - Submit homework
- Github
  - Project work

Homework Assignments

- Six full assignments, two partial (reading only)
- Programming assignments due Tuesday at noon, Discussion questions due Tuesday at 5pm
- Assignments include:
  - Data analysis in Pandas
  - Writing / Testing k-Nearest Neighbors algorithm
  - Setting up Python package & continuous integ.
  - Object-oriented programming
  - Programming style & documentation
  - Docker container and virtualization

Group Project

- Groups of 3 - 5 students
- Must make use of at least one complex dataset
- Must include at least one library API call ***or*** a complex visualization and analysis

More discussion on project ideas and proposals next week. See Canvas for more information.