Computational Thinking for Social Scientists

Jae Yeon Kim

2020-09-27

Contents

Chapter 1

PS239T

Welcome to PS239T

This course will help social science graduate students to think computationally and develop proficiency with computational tools and techniques, necessary to conduct research in computational social science. Mastering these tools and techniques not only enables students to collect, wrangle, analyze, and interpret data with less pain and more fun, but it also let students to work on research projects that would previously seem impossible.

1.1 Objectives

The course is currently divided into two main subjects (fundamentals and applications) and six main sessions.

1.1.1 Part I Fundamentals

- In the first section, students learn best practices in data and code management using Git and Bash.
- In the second, students learn how to wrangle, model, and visualize data easier and faster.
- In the third, students learn how to use functions to automate repeated things and develop their own data tools (e.g., packages).

1.1.2 Part II Applications

- In the fourth, students learn how to collect and parse semi-structured data at scale (e.g., using APIs and webscraping).

- In the fifth, students learn how to analyze high-dimensional data (e.g., text) using machine learning.
- In the final, students learn how to access, query, and manage big data using SQL.

We will learn how to do all of the above mostly in **R**, and sometimes in **bash** and **Python**.

1.2 Logistics

1.2.1 Instructor

Jae Yeon Kim: jaeyeonkim@berkeley.edu

1.2.2 Time and location

- Lecture: TBD (Zoom)
- Section: TBD (Zoom)

1.2.3 Office hours

By appointment with …

1.2.4 Slack & bCourses

- We will Slack for communication (announcements and questions) and bCourses for turning in assignments. You should ask questions about class material and assignments through the Slack channels so that everyone can benefit from the discussion. We encourage you to respond to each other's questions as well.
- All course materials will be posted on GitHub at https://github.com/jaeyk/PS239T, including class notes, code demonstrations, sample data, and assignments. Students are required to use GitHub for their final projects, which will be publicly available, unless they have special considerations (e.g. proprietary data).
- Textbook: https://jaeyk.github.io/PS239T/ (bookdown version of the course Git repository)

1.2.5 Accessibility

This class is committed to creating an environment in which everyone can participate, regardless of background, discipline, or disability. If you have a particular concern, please come to me as soon as possible so that we can make special arrangements.

1.2.6 Code for conduct

TBD

1.2.7 Course requirements and grades

This is a graded class based on the following:

- Completion of assigned homework (50%)
- Participation (25%)
- Final project (25%)

1.2.7.1 Assignments

Assignments will be assigned at the end of every session. They will be due at the start of the following class unless otherwise noted. The assignments will be frequent but each of them should be fairly short.

You are encouraged to work in groups, but the work you turn in must be your own. Group submission of homework, or turning in copies of the same code or output, is not acceptable. Remember, the only way you actually learn how to write code is to write code.

Unless otherwise specified, assignments should be turned in as pdf documents via the bCourses site.

1.2.7.2 Class participation

The class participation portion of the grade can be satisfied in one or more of the following ways:

- attending the lecture and section (note that section is non-optional)
- asking and answering questions in class
- contributing to class discussion through the bCourse site, and/or
- collaborating with the campus computing community, either by attending a D-Lab or BIDS workshop, submitting a pull request to a campus github repository (including the class repository), answering a question on StackExchange, or other involvement in the social computing / digital humanities community.

Because we will be using laptops every class, the temptation to attend to other things during slow moments will be high. While you may choose to do so, I do request that you think of your laptop screen as in the public domain for the duration of classtime. Please do not load anything that will distract your classmates or is otherwise inappropriate to a classroom setting.

1.2.7.3 Final project

The final project consists of using the tools we learned in class on your own data of interest. First- and second-year students in the political science department

are encouraged to use this as an opportunity to gather data to be used for other courses or the second-year thesis. Students are required to write a short proposal by March (no more than 2 paragraphs) in order to get approval and feedback from the instructor.

During sections in April we will have **lightning talk sessions** where students present their projects in a maximum 5 minute talk, with 5 minutes for class Q&A. Since there is no expectation of a formal paper, you should select a project that is completable by the end of the term. In other words, submitting a research design for your future dissertation that will use skills from the class but collects no data is not acceptable, but completing a viably small portion of a study or thesis is.

1.2.8 Class activities and materials

1.2.8.1 Lecture

Classes will follow a "workshop" style, combining lecture and lab formats. The class is interactive, with students programming every session. During the "skills" parts of the class, we will be learning how to program in Unix, HTML, and R by following course notes and tutorials. During the "applications" sections, we will follow a similar structure, with occasional guest speakers.

1.2.8.2 Section

The "lab" section will generally be a less formal session dedicated to helping students with materials from lecture and homework. It will be mostly student led, so come with questions. If there are no questions, the lab turns into a "hackathon" where groups can work on the assignments together. Section is required unless prior permission to miss it is obtained from both the instructor and one's groupmates. Attending office hours is not a substitute for attending section.

1.2.8.3 Computer requirements

The software needed for the course is as follows:

- Access to the UNIX command line (e.g., a Mac laptop, a Bash wrapper on Windows)
- Git
- R and RStudio (latest versions)
- Anaconda and Python 3 (latest versions)
- Pandoc and LaTeX

This requires a computer that can handle all this software. Almost any Mac will do the job. Most Windows machines are fine too if they have enough space and memory.