STATISTICAL COMPUTING

POL 281 | WINTER 2021
Professor Rachel Bernhard
ribernhard@ucdavis.edu

Classes

January 5 – March 9

Lecture + Section:

Tuesdays, 12:10-3:00 pm
https://ucdavis.zoom.us/j/91876220609?pwd=UG1jU1JIV3BadTJjT1lHQVZJUmVYQT09
or Meeting ID: 918 7622 0609, Passcode: 938310

Office Hours

By appointment (please Slack me to schedule), using https://ucdavis.zoom.us/j/92120972638?pwd=K25RNEpXTEJWOUd3VitnN2FBTmlnUT09, or Meeting ID: 921 2097 2638, Passcode: 445227

Overview

This course will provide graduate students the technical skills necessary to conduct research in computational social science, introducing them to the programming skills and application knowledge in R that they will need to be successful in their own research.

The course divides into three main sections. In the first, students dive into the structure, analysis, and visualization of data. In the second, students learn how to collect web data using APIs and web scraping. In the third, students learn additional means of classifying, analyzing, and visualizing data through tools like automated text analysis and machine learning.

Objectives

- Write, execute, and debug R code for novel data collection, cleaning, analysis, and visualization
- Be familiar with the concepts and tools of a variety of computational social science / digital humanities applications
- Be familiar with the basic guidelines around reproducible research, good scientific computing practices, and ethics/privacy/legal quandaries
- Learn independently and train themselves in a variety of computational applications and tasks through online documentation

Class in the Era of COVID

As a methods course, this class is light on reading and heavy on homework, which in this class will primarily mean a combination of individual programming through DataCamp (DC is an online course provider that, most importantly, provides an online sandbox so you can “practice” and debug your code even when I am not there) and pair programming to undertake independent data collection and analysis.

The structure of class each Tuesday will be roughly as follows:

12:10-1:30 pm: lecture introducing new topic/material
1:30~1:45 pm: break
1:45~2:15 pm: Q&A on previous week's material
2:15~3:00 pm: section—pair work/homework/office hours as needed

Wow, that's a weird structure, you might be saying to yourself! Yes, yes it is. Here's the idea: three hours is too long to listen to a lecture on programming. Research suggests that we can pay close attention to something like that for about 90 minutes before needing a break. My experiences teaching online in the last year suggest that this is especially true when we are spending very large chunks of time on Zoom. So, I will lecture on a new topic (say, Data Wrangling) for about all the time we can stand, and then we will take a break. When we resume, I will answer questions and conduct review as needed on the previous week's topic (say, Data Structure), which you will by then have had additional homework/DC coursework on and doubtless have questions about as a result. When we finish Q&A, however long that takes, you will then be able to finish the homework and/or work in pairs on projects as needed, like you normally would in a "section" (albeit with me present rather than a TA).

In short, I think this class structure will allow us maximum use of the time we have together while not being oppressive in its length and intensity. That said, because it is my first time online-teaching this course, we may need to adjust the format further. Please come with generosity of spirit and a willingness to innovate, and I will do my very best to be flexible and support you during what I know is an especially challenging time for graduate students.

Finally, I strongly encourage you to be proactive in letting me know if something happens—a health or family event—that may affect your work, *even if it doesn't result in an absence*. Sometimes, events and situations may affect your work for longer or harder than you initially anticipate, and it is much easier for me to work with you to find solutions ahead of time than try to "fix" things after it has become a problem.

Other Policies

If you are a parent and your childcare falls through, you are welcome to bring your child or infant to class provided they are able to be present without disrupting class. Similarly, if you are nursing, you are welcome to breastfeed in class.

I strongly recommend you use Slack rather than email to contact me; it may be hard to imagine, but I receive roughly 120 emails a day and relatively few Slack messages, so your odds of getting a response (especially for a quick question) go up substantially if you write in Slack. If you must email, please include "POL 281" in the subject line of your emails; if you do not, your email is likely to end up in the wrong folder and may be missed. I will try to respond to emails within 48 hours during the week or 72 hours over the weekend, and I usually respond to student emails each afternoon. As graduate students, in both class and via messages, you are welcome to address me as Rachel.

If you need disability-related accommodations in this class, and/or if you have emergency medical information that you wish to share, and/or if you need special arrangements in order to participate in Zoom meetings, please inform me immediately. Please email me or see me during office hours. For disability-related accommodations, you must also obtain an accommodations letter (https://sdc.ucdavis.edu), which will be sent directly to me.

As a UC Davis student, we trust you to conduct your academic affairs ethically. Betrayal of that trust will not be tolerated. Cheating in an online course includes, but is not limited to, having someone take a quiz or complete an assignment for you, or using someone else's written work or materials without appropriate citations (plagiarism). I take violations of academic integrity seriously. If you have questions about how best to cite another's work or facts in the public domain, please write. When in doubt, cite. I recommend the Chicago Manual of Style's author-date format if you don't have a favorite. If you have questions about whether an action qualifies as misconduct, please talk to me.

TECHNICAL REQUIREMENTS

We will use Zoom for lecture and section, R and R Studio for programming, Datacamp for some of the homework assignments, and Slack to communicate about class. All class materials, including class notes, code demonstrations, sample data, and assignments, will be made available through Slack and Dropbox.

Zoom

During synchronous class time, please silence and put away all other devices: cell phones, tablets, etc. Even if you are muted on Zoom, please do not load or listen to anything, including music, that might distract the class if you were suddenly unmuted (mistakes happen): do your part to make the classroom a learning environment.

If you must miss a week of class, let me know so that we don't hold the class up waiting for you. Please feel free to consume snacks or drinks during lecture, but reserve eating full meals for the breaks.

R and RStudio

Many of you may already have this software installed. If not, see below. Please note that if you already have R and update to a newer version, you will have to reinstall any previously installed packages.

For Windows: Install R by downloading and running the .exe file from CRAN (http://cran.r-project.org/bin/windows/base/release.htm). Also, please install the RStudio IDE (http://www.rstudio.com/ide/download/desktop). All you need is RStudio Desktop.

For Mac: Install R by downloading and running the .pkg file from CRAN (http://cran.r-project.org/bin/macosx/R-latest.pkg). Also, please install the RStudio IDE (http://www.rstudio.com/ide/download/desktop). All you need is RStudio Desktop.

For Linux: You can download the binary files for your distribution from CRAN (http://cran.r-project.org/index.html). Or you can use your package manager (e.g. for Debian/Ubuntu run `sudo apt-get install r-base' and for Fedora run `sudo yum install R'). Also, please install the RStudio IDE (http://www.rstudio.com/ide/download/desktop). All you need is RStudio Desktop.

DataCamp

You can enroll in DataCamp using your UC Davis email and this link: https://www.datacamp.com/groups/shared_links/3325f52c30066c98d131976d0a8d2dc3c27e1d02723aad60df07e2f9b95a49b2. Once you do so, you will be automatically enrolled in the assignments for POL 281.

Slack

We will use Slack for class communications. You can enroll in our class channel using https://join.slack.com/t/ucdavis-i6z6148/shared_invite/zt-kka1pybn-pSIagWsKud8s7QRSRwgnxg. Note that you will need a UC Davis email address for the invite link to work.

Other

You will need Google Chrome installed for the section on HTML/CSS/Javascript.

ASSESSMENTS

As mentioned above, class is light on reading and heavy on homework. The formally assessed parts of the class are as follows:

Class Participation | 10%

I trust that in-class participation (e.g., during Q&A) is self-explanatory. We will use a Slack channel for questions and comments on the coursework that occur outside of lecture, and the Zoom chat or raising one's hand for questions and comments during lecture.

Your participation can take many forms, both verbal and virtual. However, to count for class participation, your participation must benefit others: so, for instance, coming to office hours does not count as class participation. However, that does mean that other activities outside of class—for instance, organizing a study group, or taking notes for a fellow student—do count as participation. For those activities that do not occur in my presence, simply send me an email notifying me of your (or a classmate's) work.

DataCamp Courses (aka Problem Sets) | 25%

Most of the homework will be through DataCamp (DC), which offers "courses" (~2-3 hour coding trainings with little problem sets) and "tracks" (several linked courses in a row). We will use DC for two main reasons. First, it allows me to view your course progress remotely, which can help me see where folks are getting stuck and ensures everyone gets partial credit even if they can't complete a track. Second, because DC uses a "sandbox" format—a browser-based interface that allows you to type and run code to solve the problem sets—it ensures you will be able to get immediate help debugging your code, which is much trickier to get in a timely fashion when we are not all in the same place.

DataCamp also has a feature that lets you "test out" of a given course after taking a short assessment. So, for instance, the first track starts with an "R for Beginners" refresher course. If you've been using R recently or heavily, you can take the assessment to skip this course and move right on to "Intermediate R."

Pair Projects | 65%

The primary deliverables for this course take the form of paired (you + a classmate) projects. The projects break down into five main components:

1. **Project Proposal (5%):** The final project consists of using the tools we learned in class on your own data of interest. Pre-ABD 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 prospectus; post-ABD students are encouraged to gather data that can be used for a conference paper. Students are required to write a short proposal by Tuesday, January 26 (no more than 2 paragraphs, to be posted in the relevant Dropbox folder) in order to get approval and feedback.

2. **API Project (10%):** You and your groupmate will write original code to collect, clean, and analyze data gathered through an API. This will require you to submit three things: A) your code, B) the cleaned csv file you produce, and C) an analysis or visualization of your data (e.g., a cross-tab, a scatterplot, etc.). This must all be copied into the relevant Dropbox folder by 11:59 pm on Wednesday, February 3.
3. **Scraping Project (15%):** You and your groupmate will write original code to collect, clean, and analyze data gathered by scraping web data. This will require you to submit three things: A) your code, B) the cleaned csv file you produce, and C) an analysis or visualization of your data (e.g., a cross-tab, a scatterplot, etc.). This must all be copied into the relevant Dropbox folder by 11:59 pm on Wednesday, February 17.
4. **Final Presentation (10%):** The final class we will have "lightning talks" where students present a working version of their projects in a maximum 5-minute talk, with 5 minutes for class Q&A. These will occur on the last day of class, Tuesday March 9. Presentations should be copied into the relevant Dropbox folder before the beginning of class.
5. **Final Project (25%):** Like the other projects, the final project consists of submitting fully commented code (I recommend an RMarkdown file), cleaned csvs, and at least two informative visuals of your data (e.g., cross-tabs, plots, etc.). The end product should look like outline of a paper: lacking a full write-up, lit review, etc., but with well-explained code in lieu of a methods section and well-explained results with visuals. Since there is no expectation of a formal paper, you should select a project that is completable by the end of the term. *Feasibility is key.* In other words, submitting a research design for a future paper that will use skills from the class but collects no data is not acceptable, but completing a viably small portion of a study is. Final projects should be copied into the relevant Dropbox folder and are due by 11:59 pm on Tuesday, March 16. If for any reason your data cannot be shared (e.g., proprietary or sensitive data), please let me know in advance so we can arrange an alternative.

COURSE OUTLINE

Week 1 | January 5

Lecture:	Introduction + Data Structure
Start:	DataCamp (DC) Track on R Programming

Week 2 | January 12

Lecture:	Data Wrangling
Q&A:	Data Structure
Due:	DC Track on R Programming
Due:	Slack me confirming your project partner(s) for the term!
Start:	DC Track on Data Manipulation in R

Week 3 | January 19

Lecture:	Data Visualization
Q&A:	Data Wrangling
Due:	DC Track on Data Manipulation in R
Start:	DC Track on Data Visualization in R

Week 4 | January 26

Lecture:	APIs
Q&A:	Data Visualization
Due:	DC Track on Data Visualization in R
Due:	Post project proposal (two paragraphs) to Dropbox
Start:	API Project

Week 5 | February 2

Lecture:	HTML/CSS/Javascript
Q&A:	APIs
Due (Weds.):	API Project
Start:	DC Course on Working with Web Data in R and DC Course on Web Scraping in R

Week 6 | February 9

Lecture:	Web Scraping
Q&A:	HTML/CSS/Javascript
Due:	DC Course on Working with Web Data in R and DC Course on Web Scraping in R
Start:	Scraping Project

Week 7 | February 16

Lecture:	Text Analysis I
Q&A:	Web Scraping
Due (Weds.):	Scraping Project
Start:	DC Track on Text Mining with R

Week 8 | February 23

Lecture:	Text Analysis II
Q&A:	Text Analysis I
Due:	DC Track on Text Mining with R
Start:	DC Track on Machine Learning Fundamentals in R

Week 9 | March 2

Lecture:	Machine Learning
Q&A:	Text Analysis II
Due:	DC Track on Machine Learning Fundamentals in R
Start:	Final Project Presentations

Week 10 | March 9

Lecture:	Wrap-Up
Due:	Final Project Presentations

March 16 | Final Projects Due

Syllabus references and credits are due to: Richard Ashcroft, Amber Boydstun, Julia Christensen, Justin Grimmer, Chris Hare, Ryan Hübert, Jae Yeon Kim, Brandon Stewart, Rochelle Terman, and Michelle Torres.