DRAFT - PS 811: Introduction to Statistical Computing

Marcy Shieh

Fall 2020

E-mail: mshieh2@wisc.edu
Web: marcyshieh.github.io
Office Hours: TBD
Class Hours: TBD
Office: polisci811-fall2020.slack.com
Class Room: TBD

Course Description

Much of modern social science takes place on a computer. As you mature in the discipline, you will find yourself desperate for efficiency and organization. This class will guide both quantatiative and qualitative scholars on how to better organize data, conduct analyses, compile literature reviews, create bibliographies, and complete a paper using popular computational tools and reproducible workflows. While many research projects are inundated with decisions that require theoretical justification, this class will at least help you make your analytical work transparent to the casual interloper.

Political Science 811 is an introductory class of computing resources for political and social sciences. While not all political scientists use these tools (yet), most of the top journals in the discipline (e.g., *Journal of Politics, American Political Science Review, American Journal of Political Science*, and other journals that you may want to publish in) now require authors to submit replication files. Instead of having to retrace all your steps, these tools will allow you to organize as you go along and keep you honest as a scholar. In addition, I can attest that these skills transfer to technical and research-oriented jobs outside of academia. Even if your job description is not strictly technical in nature, you may end up having to interact with software developers and data analysts who operate within these computing frameworks.

The goal of this course is to familiarize you with popular computing tools and where the discipline is headed so you can implement them in your own work. Other method courses focus on the foundational aspects of quantitative and qualitative research, which are important and great, but they leave a gap between "doing the research" and "writing the paper." This class is meant to fill in that gap, which can often be complicated, confusing, and ambiguous. The hope is that this class will introduce you to tools that will ameliorate that daunting chasm, or at least make it more managaeable.[1]

Learning Outcomes

1. Prepare, model, and visualize data using the statistical programming language **R**.

[1] Special thanks to my PS 811 predecessors, Mike DeCrescenzo and Devin Judge-Lord, for sharing their materials with me.

2. Write a research paper using **R Markdown**.
3. Manage your research projects using **Git**.

Course Credit Information

This class is a 1-credit pass/fail course. This means that the class meets for one (1) hour each week over the fall semester and carries the expectation that students will work on course materials for about two (2) hours out of the classroom for every class period.

Our course will meet for two (2) hours every week. Students *should* attend the first hour synchronously (TBD) or asynchronously to gain all the necessary materials for the course. This hour will include lectures, code demos, and some applied exercises. While there may be some code snippets on the lecture slides, I will upload the full code scripts to Canvas so you can reference them later.

Student then have the option to attend the second hour, where I will go over some questions on the problem set and answer any burning questions you may have about the course materials. In essence, you may interpret the second hour as additional office hours.

Required Software

You will need to download the following software tools on to your computer. Fortunately, they are all free.

- **R**, a free software environment to conduct statistical analyses and create data visualizations. For those uninitiated to the world of statistical programming, R is basically a super powerful version of Microsoft Excel, but way more efficient. Download R at https://www.r-project.org/.
- **R Studio**, a user-friendly environment for programming in R. One way to think about this is that R is the engine, and R Studio is the car. You can use the engine without the car, but you cannot drive the car without the engine. Or something like that. Downloda RStudio Desktop (the free open source version) at https://rstudio.com/products/rstudio/download/.
- **LaTeX**, a software system for preparing documents. I don't envision us using LaTeX in its natural state, but you will need to download the software in order to compile R Markdown documents. Download LaTeX at https://www.latex-project.org/get/.
- **Git**, a version control system. You know about track changes in Microsoft Word? Git is track changes, but for all your files. It tells you who made changes to the file, what changes they made, and when they made them. Thus, it is ideal for collaboration and working solo. Instead of naming your files `document_2020-08-25.Rmd` then over-writing it as `document_2020-08-26.Rmd` when you make some changes, Git does all of this for you without you possibly confusing dates, saving the file without saving *as* the file, and all the other problems you may run into when you attempt to version control manually. You may already have Git in your computer. Check whether you have Git on your computer at https://git-scm.com/book/en/v2/Getting-Started-Installing-Git. **Do not install from source.**

I am not dogmatic about how you choose to code in R, as long as you *know* how to code and *understand* what you are doing. Half the battle with learning R is fostering confidence and curiosity. Only then, would you feel comfortable running different commands and Googling your R questions.

This class is an introduction on how to use R, and aims to give you the ability to self-learn R and be literate in all kinds of R code. Thus, I will try to cover both base R and the Tidyverse language, which I will talk about more during the course, but it is ultimately up to you to determine what works for you.

Deliverables & Grading

The assignments in this class are meant to give you extra practice on the materials we learn in class and contribute to your final project. Late assignments will not be accepted.

1. **Weekly assignments: 50%** These assignments will help you apply the skills learned in class and will focus on data tasks in R, then integrating them into the Git workflow.
2. **Midterm project: 20%** You and a partner will work on a project together where you demonstrate your ability to use Git for collaboration purposes. You will present your project live or submit a recording. Your colleagues can provide feedback on Slack.
3. **Final project: 30%** You will be responsible for submitting a project repository that includes the R Markdown paper, the R code for the analysis, graphics, and bibliography, etc. and is version-controlled with Git and pushed to Github. In essence, I should be able to reproduce your entire project with the files in your project respository. You will present your project live or submit a recording. Your colleagues can provide feedback on Slack.

Class Logistics

Attendance

Attendance is not required but strongly recommended. If you cannot attend class synchronously, you should watch the video recordings of the lectures in your own time. The advantage of attending class synchronously is that you will be *encouraged* to ask questions as you learn the concepts. However, if it is not possible at all for you to attend lecture synchronously (e.g., life circumstances, time zone differences), please let me know and we can discuss accommodation options.

Academic Integrity

Academic integrity is important to the University of Wisconsin–Madison. As such, please do not plagiarize, use unauthorized materials or fabricated data, tamper with another student's work, or assist any students in these acts. If you collaborated with your colleagues (i.e., worked on the assignment together), do not copy and paste their code or pass them as your own. To keep yourself honest, you may also link to the website(s) or forum post(s) that helped you find the answer. Type out the code and credit them for their help. Explain the code in ways that you understand and can reference in the future. If you fail to follow best practices, you can face disciplinary action, which includes failing the assignment, failing the course, or facing other disciplinary action

by the department or the university. I will report substantial or repeated cases of misconduct to the Office of Student Conduct & Community Standards for further review. Refer to https://conduct.students.wisc.edu/misconduct/academic-integrity/ for more information.

Diversity & Inclusion

Diversity is a source of strength, creativity, and innovation for UW–Madison. We value the contributions of each person and respect the profound ways their identity, culture, background, experience, status, abilities, and opinion enrich the university community. We commit ourselves to the pursuit of excellence in teaching, research, outreach, and diversity as inextricably linked goals. The University of Wisconsin–Madison fulfills its public mission by creating a welcoming and inclusive community for people from every background — people who as students, faculty, and staff serve Wisconsin and the world. Visit diversity.wisc.edu for more information.

Accommodations

The University of Wisconsin-Madison supports the right of all enrolled students to a full and equal educational opportunity.

If you observe any religious holidays, please let me know and we can discuss accommodations.

The Americans with Disabilities Act (ADA), Wisconsin State Statute (36.12), and UW-Madison policy (Faculty Document 1071) require that students with disabilities be reasonably accommodated in instruction and campus life. Reasonable accommodations for students with disabilities is a shared faculty and student responsibility. Students are expected to inform faculty [me] of their need for instructional accommodations by the end of the third week of the semester, or as soon as possible after a disability has been incurred or recognized. I will work either directly with the student you or in coordination with the McBurney Center to identify and provide reasonable instructional accommodations. Disability information, including instructional accommodations as part of a student's educational record, is confidential and protected under FERPA. Visit https://mcburney.wisc.edu/instructor/ for more information.

Sudden life circumstances can certainly impact class performance. There are various resources on campus to help you navigate difficult times, such as the Dean of Students Office and the [Division of Student Life](https://students.wisc.edu/].

Please let me know as soon as possible if you require any accommodations at all, including anything that I did not mention. I want to ensure that this course contributes to your knowledge base rather than be a hindrance to your success.

Helpful Resources

At some point, you might get stuck, as we all do. But have no fear! Not only will this class allow you to make mistakes, you will have me and your fellow colleagues as resources. And, since computing tools are so ubiquitious in academic and professional circles, it is very likely that someone also has the same question as you do and asked it on an Internet forum. With any luck, someone may have even answered the question!

1. **Class Attendance** You should attend both first and second hours of class. The first hour introduces you to concept. The second hour allows you to ask questions about what you just learned, making it the best way to ask questions and learn.

2. **Class Materials** I strongly recommend that you reference the class materials first. Rarely would I include a question in an assignment that you will be unable to solve by applying concepts that we went over in lecture. The answer keys will reference the slide where the concept is discussed in case you want to go back to the exact slide and review the concept.

3. **Collaboration** As we continue to navigate our virtual higher education landscape, we are confronted with hurdles when it comes to ease of collaboration. Even in normal times, however, we often have to work with collaborators who we cannot meet in-person. As such, I highly encourage you to collaborate with each other on weekly assignments. For effective collaboration, I recommend doing the exercises on your own, identifying the questions you need help on, then showing up to a collaborative space with what you know and what you don't know. Collaboration *does* mean seeking help among each other, but it *does not* mean copying and pasting text and code written by someone else and passing it off as your own. If you use your colleague's code, as one often do when they learn coding, type out the code yourself and make sure to *credit* your colleague for their work. If you have some experience with coding or feel comfortable with the material, please consider assisting your colleagues. Sharing knowledge is as rewarding as gaining knowledge, but hoarding knowledge is not rewarding at all.

4. **Slack** My hope to facilitate collaboration is why I've set up Slack for this course. Slack is a communication platform that came to fame among software developers, but is now used by project teams of all stripes. Posting on Slack is not a requirement, but it is certainly a way for you to reach out to help from me and/or your colleagues and get an answer as soon as possible. If you believe your question will be beneficial to not only you, but to your colleagues as well, you should *definitely* post it on the Slack. As a graduate student, I'm in front of my computer quite a bit, but I will give you all a chance to help each other with questions from the assignment before I jump in. Some examples of Slack-friendly items include: questions about the assignments, resources that you have found useful on and off the web (e.g., books, websites, blogs, Twitter posts, videos), mnemonics, or cross-references with your other courses.

5. **Office Hours** While I plan to be available on Slack on most weekdays (and maybe even weekends), I will also hold regular office hours. To schedule office hours with me, simply email me during my regular office hours and I will jump on a video call with you.

6. **Email** If your question is of a personal nature and is specific to you and only you, please email me. Expect me to respond within 24 hours and if I do not, please feel free to email me again. You will *not* be annoying me.

7. **Google (or your preferred search engine)** There are *tons* of resources online about R and Git. You will start to notice that your search results will often yield either posts from Stack Overflow and Stack Exchange. Use them. Much of Goggling in R to achieve required results is learning how to refer to certain commands. You learn as you go. Sometimes the Stack Overflow and Stack Exchange communities can get a bit abrasive (in my experience). I recommend the R Studio Community if you do actually want to ask a question because you can't find an answer to it anywhere. I love Googling and I have spent many hours going down on various R-related Googling rabbit holes late into the night. I don't recommend

this. While it has helped me learn *how* to Google, there comes a point (it's the 5-hour mark for me, but it might be the 2-hour mark for you) where you should ask for help.

Class Schedule

The schedule includes topics, resources, and due dates for the midterm and final projects. I will try to keep resources at a minimum. You will likely be overwhelmed by the R you will learn in PS 812 and the R you will learn in here, so I want to minimize the confusion and maximize the practicality.

I will try to provide some instructions for setting up everything you need prior to each lecture, but feel free to ask questions in class, on the Slack, or via email. I would rather give you time to set up everything in your own time and ask for help when convenient than feel left behind in class and freak out, which is often the story of my life.

This syllabus might evolve over time and accommodate any feedback I receive over the course of the semester. Think of this syllabus as a road map. There are certain destinations we'd like to hit, but it's possible for us to take a different route to get there.

Week 01, 08/15 - 08/19: Set up all the tools you will need for the class, including R, R Studio, R Markdown (knit as PDF and HTML), Git, and Github.

- Connect RStudio to Git and Github
- Markdown Syntax

Week 02, 08/22 - 08/26: Do you what you already do but with cool computing tools: create a project, create a document, write some math in your document, and organize a bibliography.

- Project Management with R Studio
- Mathematics in R Markdown
- Bibliographies and citations

Week 03, 08/29 - 09/02: Base R vs. Tidyverse, and how to be comfortable with both.

- Syntax equivalents: base R vs Tidyverse

Week 04, 09/05 - 09/09: Some useful functions for your own work and PS 812.

- Iterations and Many Models
- Purrr - tips and tricks

Week 05, 09/12 - 09/16: Manipulate data.

- Data Manipulation

Week 06, 09/19 - 09/23: Create graphics.

- Graphics

Week 07, 09/26 - 09/30: Regression and model outputs.

- Analysis
- Regression Analysis in R

Week 08, 10/03 - 10/07: Midterm project presentations.

Week 09, 10/10 - 10/14: Dealing with data.

Week 10, 10/17 - 10/21: Popular analysis tools: text analysis, mturkr.

Week 11, 10/24 - 10/28: Some more git.

Week 12, 10/31 - 11/04: Create your own Github website.

- How to Host a Website on Github

Week 13, 11/07 - 11/11: Final project presentations.