

Policies

Instructor | Tad Dallas

Location | Jones 101

Times | T & TH 2:50pm - 4:05pm

Drop-in hours | Th from 1:00pm - 2:40pm

Prerequisites

You should be generally numerically literate, and some prior programming experience will be helpful. Students new to programming might find Hands on Programming with R to be helpful. Students with significant experience in programming and statistical analysis should find themselves well prepared but should find plenty still to learn in each lesson.

Instructional Methods

The majority of the course will be taught as live-coding demonstrations, where students are encouraged to follow along on their own machines. There will be a number of practice problems and group activities throughout each class period, where we will use what we've learned to gain insight into some data. While students are working on exercises the instructor will actively engage with students to help them understand material they find confusing, explain misunderstandings and help identify mistakes that are preventing students from completing the exercises, and discuss novel applications and alternative approaches to the data analysis challenges students are attempting to solve.

Grading

Grades will be assigned using the following weighted components:

Undergraduate students

component	weight
HW Exercises	30%
In class problems	20%
Participation	10%
Group project	40%

Graduate students

component	weight
HW Exercises	30%
In class problems	20%

component	weight
Participation	10%
Project	40%

Homework assignments

There will be a total of 4 homework assignments, corresponding to the number of modules in the course. Homework exercises are set up to be the same for both undergraduate and graduate students. This may change depending on how much folks struggle. The point is not to overwhelm, but to get everyone as much experience and excitement as possible without overburdening.

All assignments are due by immediately **before** the start of class on the day indicated.

Assignments should be submitted as instructed (through GitHub).

Details of grading criteria will not usually be announced in advance. It is expected that students in this course will have a wide range of prior experience and ability, and grading will aim to reflect learning and effort in the course. Every commit you make in your homework will run your code through a set of checks. These checks are just as much for me as for you. They can tell you if a test has failed, and you have access to see which tests your response has failed to meet.

In-class problems

Every week we will have a coding problem that students will do in class. This will be something fairly low stakes, and will try to reinforce the material from that week (e.g., if the lecture is on loops, expect to be writing a loop in the in-class problem). These will start on week 2 (or maybe 3) of the course, and run until week 14 (so between 10 and 11 in-class problems total). These must be completed during class time, and I will do my best to allocate 10-15 minutes at the end of Thursday lecture for this. If we can't fit it in, the problem will be due by midnight on Thursday.

Set up a private repository on GitHub called `codePractice`. You will upload your answers to this repo (you can just use the **Add file** drop-down from the right panel at the top of the repository). Name the file `weekX.Rmd`, where X corresponds to the number of that week (e.g., `week2.Rmd`).

Project details

The project will be a demonstration of your learning and allow you to use the new tools you have acquired in the course. This will be done in small groups (2-3) for undergraduates and will be a solo project for graduate researchers. The grading criteria will be slightly different between undergraduate teams and graduate researchers, as my expectations are different. However, all projects should hit the following points.

The final project will be an R markdown document that reads in and analyzes a data source. I can provide data, if necessary, but would prefer if data came from the student (either from their research or on something they are passionate about).

You will actively develop your work on Github, structuring your directory as we discussed in class. The final product will be an R markdown file that is entirely reproducible. This means I will clone your directory, and run your files on my local machine (so don't forget to `install.packages()`).

The final project will consist of 3 parts. First, you will develop a project proposal. Second, you will do the proposed project, and the final product will be a versioned and end-to-end reproducible analytical workflow that could potentially lead to a peer-reviewed publication. Lastly, you will briefly present your work to the class (15 minute scientific presentation).

This means that I will expect not only a well-written and clear manuscript, but also documentation to easily reproduce all analyses, figures, and compile manuscript text.

Final project proposal Please prepare a short proposal on your final project idea by the end of the first course module. The proposal should include:

- Title & description of the project
- Team members names
- A description of the data required, and how it will be obtained (e.g. URL/DOI to data source)
- 3 questions / analysis tasks you will perform on the data; in the spirit of the assignments we have been doing.

Please create your proposal in a markdown file called `proposal.md` in the root directory of the final project repo.

Project development You will develop the project throughout the semester, using tools and approaches that you learn during the course. Do this early and consistently. I will be able to track your progress on GitHub, meaning that I expect to see continued development on the project and not a last minute flurry of commits.

Graduate students: this is an opportunity to push forward your research. Ideally, the resulting project manuscript is something that could be submitted to an academic journal.

Final product You will present your final project as a 15 minute presentation at the end of the semester. It should provide context and information about the ‘what, why, and how’ of your research question, and go through your findings, preferably demonstrating your ability to visualize data.

Graduate students: think of this as something you would feel comfortable presenting at an academic conference.

Important dates:

Sept 21: first homework due Sept 28: final project proposal due October 12: second homework due November 9: third homework due December 7: fourth homework due

Last day of classes is December 8, reading day is Dec 9.

Present final project during final exam period: > Thursday, December 14 - 4:00 p.m.

Make-up policy

I will use the latest commit in the GitHub repository for a given assignment as the submission. Commits made after this point will not be considered. In cases of emergency, or with prior approval, I am happy to consider late assignments, but the overall grade will be docked by 20% and will be accepted within 3 days of the original due date. Try to plan ahead to get the assignments done. Some of the assignments may seem fairly straightforward, but may actually be more challenging than they seem.

Attendance Policy

The lab-based, hands on course design really depends on students being in class, for every session. I expect students to make every effort attend every class. I cannot accommodate scheduling conflicts that would cause a student to regularly miss part of class. However, I recognize that now and again an occasional absence will be unavoidable. Please notify the instructor beforehand if possible. I will not require any note or explanation and trust you to make the right decisions for your own education, but advanced notice may help ameliorate the disruption. Keep up with the reading assignments while you are away and we will all work with you to get you back up to speed on what you miss.

Academic Honesty

Cooperation has a limit. You should not copy your code or answers directly with other students. Feel free to discuss the problems with others, but write your own solutions. Penalties for cheating are severe – they range from a zero grade for the assignment or exam up to dismissal from the University, for a second offense.

Rather than copying someone else's work, ask for help. You are not alone in this course! If you invest the time to learn the material and complete the projects, you won't need to copy any answers.

Disability services

My goal is to help you learn. Students who have any difficulty (either permanent or temporary) that might affect their ability to perform in class can reach out to the University of South Carolina Disability Services staff.

More information on registering a disability is available at U of SC Disability Services.

Support

You are not alone in this course; your student colleagues and the course instructor are here to support you as you learn the material. It's expected that some aspects of the course will take time to master, and the best way to master challenging material is to ask questions. Time will be set aside in each class to ask questions and discuss the material. You are encouraged to bring up related questions that arise in your research as well.

Code of Conduct

Our course is committed to providing a respectful and welcoming environment to all participants. Please review the Contributor Covenant Code of Conduct guidelines for respectful and harassment-free conduct.

To report an incident or request more information, contact the Office of Equal Opportunity.

Course Technology

Students are required to provide their own machines and to install free and open source software on those machines. Support will be provided by the instructor in the installation of required software. If you don't have access to a computer please contact the instructor and they will do their best to provide you with one.

Materials & Resources

All reading material required for this course will be made available through this website and links to related resources.