Reproducible Research in R (Biol 4800)

Location:

Time:

Instructor: Dr. Tad Dallas (tadallas@lsu.edu)

Office: A343 Life Sciences

Office hours: T and Th from 2:30 - 4:20pm

Course Overview:

Scientific knowledge builds upon existing scientific knowledge. This knowledge is generated through observation and experimentation, which results in scientific publications, but also in data and computer code. While the published record may persist, much of the computer code and data are no longer able to be confirmed due to a degradation of the analytical tools used (e.g., fortran punchcards) or unreadable data formats (e.g., some Microsoft Works data formats).

Only recently have scientists made their data and code available, allowing other scientists to examine the code and reproduce the results of the original paper. This ability to reproduce analyses is at the core of this course.

With the goal of doing reproducible research, we will learn many basic data science skills, including the unix command line, version control, makefiles, and the R programming language. Together, the student will gain the ability to create an analytical pipeline that can successfully run on any computer with sufficient memory.

Course Goals:

Over the course, it is expected that students gain

- a knowledge of the issues surrounding reproducibility and openness of scientific research
- an ability to design and implement reproducible scientific workflows
- experience with algorithmic thinking
- experience using the unix command line, R, and other computational tools
- an appreciation and/or deep hatred for reproducible research

Syllabus Subject to Change:

Changes to the syllabus may be made during the semester. The most up-to-date and current syllabus will always be available on the course website (on Github). Syllabus and grades will be available on Moodle (as standard).

Grading

There will be a total of 500 points, consisting of 5 assignments, some attendance/participation points, and a final project.

The final project will be performed in small groups for undergraduate students (4800), and independently for graduate students (7800).

Break down:

Item	Points	Total
Assignments	5 x 50	250
Attendance/Participation	50	50
Project	150	150
Final exam	50	50

Assignments:

There will be 5 assignments throughout the semester. *Each assignment will be worth 50 points.* The assignments will be submitted through Github, a version control platform. **Assignments are due before class on Thursday**

Attendance:

Much of the material presented will not be available if you aren't in class to hear it. Given the pace of the course, and the computational and programming hurdles you will almost certainly encounter, you should come to class. If you do not, you will likely fail. Attendance and participation will be worth 50 points.

Project:

The project will be a group exercise to demonstrate your learning and allow you to use the new tools you have acquired in the course.

The final project will be an R markdown document that reads in and analyzes a data source. I will provide groups with data, or they can find them from another source.

Your group will work collaboratively 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. **I must be able to reproduce your analyses**.

Additionally, your projects will have to incorporate at least 2 of the following elements:

- continuous integration (travisCI is your friend)
- a makefile (that will compile and clean the project directory)
- a license file
- visualizations (reproducible plots within the R markdown file)

The project is worth 150 points

Final exam:

The final exam will go over high-level concepts learned in the class, and be worth 50 points.

Late Assignments:

All assignments are expected to be electronically submitted by the due date. I will not accept any assignment that is submitted after the deadline and the assignment will receive the grade of 0. I will, however, grade whatever the last committed (partial submission to the class assignment repository) that you have made prior to the due date. This grade cannot be appealed at a later date, but is better than a 0. This course is a senior level class and you are expected to work at the appropriate level and be responsible for your work. All deadlines are posted with the syllabus or any changes will be announced ahead of time.

Academic honesty

Louisiana State University adopted the Commitment to Community in 1995 to set forth guidelines for student behavior both inside and outside of the classroom. The Commitment to Community charges students to maintain high standards of academic and personal integrity. All students are expected to read and be familiar with the LSU Code of Student Conduct and Commitment to Community, found online at www.lsu.edu/saa. It is your responsibility as a student at LSU to know and understand the academic standards for our community.

Students who are suspected of violating the Code of Conduct will be referred to the office of Student Advocacy & Accountability. For undergraduate students, a first academic violation could result in a zero grade on the assignment or failing the class and disciplinary probation until graduation. For a second academic violation, the result could be suspension from LSU. For graduate students, suspension is the appropriate outcome for the first offense.

Further information is provided on the LSU website

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 LSU Disability Services staff.

More information on registering a disability is available at LSU Disability Services, located at 124 Johnston Hall. Contact the Center by telephone at 225-578-5919 or via email at `disability@lsu.edu`.

Schedule

Week	Topic	Assignment
1	Background	–
2	Intro to tools (git, Github, unix command line)	
3	R basics (including Rmd)	`rIntro`
4	Analytical pipeline structure (make, Travis)	–
5	R data manipulation	`rDataManip`
6	Unit testing and line profiling	`rTesting`
7	R, APIs, and data visualization	`rAndAPI`
8	Dealing with spatial data	`rSpatial`
9	Dependency hell and containerization (or OA)	–
10	Parallel computing	–
11	Final projects 1	–
12	Final projects 2	–

For graduate students

This course presents you with an opportunity to learn essential skills for the analysis of data. In addition to the assignments and requirements of the undergraduates, you will complete a reproducible analysis of data that you bring in to the course (if you don't have any data to bring, I can provide you with some).

Undergraduates will perform a similar activity as groups of students. You will perform this as individuals. Bring your dissertation data, or an idea for a side project. I expect the final output to be in the form of a manuscript, formatted into the following sections.

Introduction

What question are you addressing with the data?

What have other scientists done before?

Where is the knowledge gap (i.e., what are you contributing)?

Methods

Tell me about the data.

Tell me all the detail to where I can recreate the analysis you performed.

Results

What did you find? Reference relevant figures and statistical analyses.

Discussion

What have we learned that we didn't know before?

How does this change the current state of knowledge?

What are caveats/exceptions/next steps?

The best part is that the entire analysis and manuscript need to be end-to-end reproducible. 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.

Your projects will have to incorporate all of the following elements:

- continuous integration (travisCI is your friend)
- a makefile (that will compile and clean the project directory)
- a license file

- a clear project directory (with subfolders and README files)
- visualizations (reproducible plots within the R markdown file)
- version control (everything on Github, with clear commit history)
- R and R markdown
- no proprietary software or format (if you turn in a .docx file, I will be quite sad)