

Crash course on reproducible computing in R

Julia Wrobel

Overview

Today, we cover:

- Course overview
- Basic principles of reproducible computing
 - Coding style
 - Using the command line
 - Project organization
 - Git and GitHub
- Lab with practice questions, if time allows

About me

Research

- Functional data analysis
- Spatial proteomics
- Application areas: cannabis impairment, neurological disorders, ovarian cancer, wearables/sensor data

Announcements

- Discussion board post due tomorrow (1/15) at 10:00AM
- Homework 1 due Wednesday, 1/29 at 10:00AM

Course objectives

My goal for this course is for you to learn practical skills you need to become a statistical methods researcher. There are two main areas we will focus on:

1. Computing

- R focused

2. Algorithms

- Look under the hood at algorithms for commonly used methods like logistic regression, LASSO, mixed effects models

My teaching philosophy

- Ask questions early and often
 - It helps you stay engaged and others are likely to benefit from your question as well
- What you learn is proportional to the effort you put in
 - Try to think about this less as a class where you try to get a grade and more as a place where you are learning the skills to be a good researcher and help you write a **great** dissertation
 - Use ChatGPT, but wisely
- Feel free to work together

Emails vs. Discussion board

Discussion board

- All questions about course content should go on the discussion board!
 - This includes homework questions
 - I will monitor the discussion board daily
 - This allows all students to benefit from questions asked about course content
 - **Posting** and **answering** questions posted on the discussion board will count towards your participation grade

Emails

- Email should only be used for schedule purposes and personal

Office hours and homework due dates

Office hours

- Tuesdays at 11:00AM
- Thursdays at 2:00PM

Homework policies

- Homeworks will be due on Wednesdays at 10am
- Late assignments will receive a maximum of half credit.

Assignments more than 3 days late will not be accepted.

- Let me know (if possible, in advance) if you have known conflicts with the due dates or a special circumstance (conference travel, family emergency)

Grading

- Homework (50%)
 - 7 assignments
- Discussion board posts (10%)
 - 4-5 posts
- Participation (10%)
 - Participation can mean asking questions in class, posting on the discussion board, or attending office hours
 - To quantify, 5+ posts on discussion board besides official posts
 - I encourage you to post computing tricks, cool algorithms, etc
- Final project (30%)
 - Related to your research, if possible

Syllabus

- Course content will be posted on Canvas
- Any other questions?

Coding style

Principles for scientific coding

In this order:

1. Code that works.
2. Code that is reproducible.
3. Code that is readable.
4. Code that is generalizable.
5. Code that is efficient.

A **minimal standard** for scientific computing is 1-3.

Advice for beginner coders

Test code before relying on it.

It's OK to **copy/paste code** from ChatGPT or Stack Overflow, but make sure you **understand how it works**.

- Run line by line and see what each does.
- Change the code and see if it behaves as expected.

Stakes for copy/paste can be high!

- Incorrect analyses.
- Expensive (inadvertent) cloud computing.

For high stakes analyses, ask a colleague for a **code review**.

Advice for more advanced coders

Getting code **correct** AND **readable** is **most important**.

- Make your code more efficient later.
- After a paper is submitted for review?

Remember: you don't get bonus points for code that "looks impressive".

Think before you code

Before you start writing code, think about what you want the code to do.

Plain English → pseudo-code → actual code

This careful thought process can ultimately lead to **more efficient** and **more robust** code development.

Don't repeat yourself

Don't repeat yourself (DRY) is a fundamental concept in programming.

- *Ruthlessly eliminate duplication*, Wilson et al.

If you write the same code more than once, it should be a function.

Command line

Basic GitHub workflow

```
git pull
```

```
git status
```

```
git add --all
```

```
git commit -m "informative commit message!"
```

```
git status
```

```
git push
```