SCHOOL OF MEDICINE & DENTISTRY
UNIVERSITY of ROCHESTER MEDICAL CENTER

**BST430 – Introduction to Statistical Computing**

Fall 2021
MW 11:00-12:40
Location: SRB 1402
Instructor(s): Andrew McDavid
Office Hours: TBD (and by appointment)
Course website: https://urmc-bst.github.io/bst430-fall2021-site/

Prerequisites**:** An advanced undergraduate course in Statistical Inference & some programming experience, or permission from the instructor.

Course Description
The purpose of this course is to provide a strong foundation in the computational skills needed for graduate coursework and research in Statistics and Biostatistics. We will cover reproducible and collaborative programming in R, with an emphasis on data analysis and implementing common statistical algorithms. If time permits, we will also introduce calling python and c++ code from R. Student will also learn the core ideas of programming - data structures, functions, iteration, input and output, logical design and abstraction. Students will learn how to write maintainable code, debug and test code for correctness. They will learn how to write, document, comment and organize code, how to set up and run simulations, how to fit simple statistical models to data, how to deal with large datasets.

The course will be taught via lectures and interactive sessions. The emphasis of the course will be on mastering the computational skills and techniques upon which subsequent coursework and research will build.

**Course Aims and Objectives**

- Be able to utilize and recognize common programming concepts and constructs
- Collaborate and share code using git and github
- Implement reproducible data analyses in rmarkdown
- Write documented and maintainable code for common statistical tasks
- Effectively debug and test code for correctness

**Materials and Access**

Required texts

None

Recommended texts

Wickham & Grolemund "R for Data Science" https://r4ds.had.co.nz/
Peng "R Programming for Data Science" https://leanpub.com/rprogramming
Wickham "Advanced R" https://adv-r.hadley.nz/
McBain "Git for Scientists" https://milesmcbain.github.io/git_4_sci/
Wickham "R Packages" https://r-pkgs.org/

Required software and programming languages

We will be using rstudio cloud to complete your assignments, though strictly speaking this is not a requirement. Otherwise, you must have a working install of R 4.1 / Bioconductor 3.13. You should bring a laptop to class every day, as we will periodically hold in-class labs.

**Assignments and Grading Procedures**

You will be evaluated in terms of **homework** and **labs** (60%), **in-class quizzes/participation** (10%), and a **take-home final** (30%). Participation will be evaluated holistically, and shall include completing polls and quizzes delivered during lecture, asking questions during synchronous lecture.

- There will be 8-10 homework (to be completed individually, primarily out of class) or labs (to be completed in groups, primarily in class) throughout the semester.
- Homework and labs will be posted and returned on GitHub.
    - The best practice for a version control system is to commit frequently with informative commit messages. Thus, this will be formal part of your homework grade.
    - I expect frequent commits. At the minimum, I expect you to commit after you have completed each question.
    - I expect informative messages for each commit.
    - Example good message: “tidying the college scorecard data.”
    - Example bad message: “More stuff”
    - Lack of frequent and informative commits will result in up to a 25% reduction in an assignment grade.
- Because `git` allows me to view your progress on an assignment, I will accept a late assignment if I see progress and a **consistent commit history** in that assignment. If I do not see any progress in an assignment, I will not accept a late submission.
- I permit corrections and regrades to assignments to receive up to 75% of original credit.

**Take-home final**

An open book, no-collaboration-permitted final will be assigned on November 9, and will be due 24 hours later.

**Academic Integrity and Programming Exercises**

Academic integrity is a core value of the University of Rochester. The academic integrity policy in this class seeks to maximize the pedagogical benefit of the homework, project and labs, as well as model norms of attribution in scientific writing and presentation. In short: when in doubt, cite.

1. In **homework**, and the **take home** (anything not marked as a lab exercise) you ****are not**** generally permitted to copy and paste your classmates’, my code, or the internet’s code, except where specifically indicated. For homework, may consult with your classmates and external resources on algorithmic and implementation details, but the code you submit must have been typed into your editor, with your fingers, the hard way, and you should cite any sources that you have manually transcribed. For the final, you may not consult with your classmates.

2. In labs, you **can**, and will often be encouraged, to electronically re-use code chunks provided by **your instructor, or your labmates**. Typically this will be done using github, but copy-paste is okay too. For re-use of other chunks of code you may find on the internet or otherwise, manual transcription is required, or seek instructor approval.
3. Adequate citation of all sources, including program code, figures or illustrations, and prose submitted for evaluation in homework, labs, exams and presentations is required. The citation standard depends on the format. For written work, citation (in a recognized format of your choice) and a bibliography are standard. For presentations (probably not applicable) verbal acknowledgment and a short reference to the origin are appropriate. For code, inline comments or acknowledgment in documentation and other scholarship is an appropriate way for provide attribution (copyright requirements not withstanding).
4. Students who violate the University of Rochester University Policy on Academic Honesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Since academic dishonesty harms the individual, other students, and the integrity of the University, policies on academic dishonesty are strictly enforced. For further information on the University of Rochester Policy on Academic Honesty, please see the *Jurisdiction and Responsibility for Academic and Nonacademic Misconduct* section in the **Regulations and University Polices Concerning Graduate Studies** *http://www.rochester.edu/GradBulletin/PDFbulletin/Regulations.pdf*

**Accommodations for Students with Disabilities**

I strive to conduct an inclusive course. Students needing academic adjustments or accommodations because of a documented disability should contact the Access Services Coordinator. For information regarding access services and support at SMD, please refer to our webpage: https://www.urmc.rochester.edu/education/graduate/current-students/disability-supports-services.aspx

**Tentative Course Schedule**

| | | |
|---|---|---|
| 1 | 25-Aug-21 | Syllabus, intro to github, intro to rstudio cloud |
| 2 | 30-Aug-21 | configuring github from RSC, RSC projects, rmarkdown (i), Intro to R (syntax), ggplot (i) |
| 2 | 01-Sep-21 | Styleguide, data from files, dplyr (i) |
| 3 | 06-Sep-21 | No class |
| 3 | 08-Sep-21 | Data structures in R |
| 4 | 13-Sep-21 | dplyr (ii) |
| 4 | 15-Sep-21 | Collaboration (merging, conficts, pull requests) with git/github |
| 5 | 20-Sep-21 | Factors |
| 5 | 22-Sep-21 | ggplot (ii), other graphing systems |
| 6 | 27-Sep-21 | Text manipulation, regex |

| | | |
|---|---|---|
| 6 | 29-Sep-21 | Indexing, iteration, linear algebra |
| 7 | 04-Oct-21 | Functions, scope, functional iteration |
| 7 | 06-Oct-21 | classes and generics |
| 8 | 11-Oct-21 | Python via Ipython and recticulate |
| 8 | 13-Oct-21 | Computer science data structures (arrays, linked lists, trees,hashmaps) |
| 9 | 18-Oct-21 | Algorithms and complexity (linear search, binary search, recursion) |
| 9 | 20-Oct-21 | Debugging ii (breakpoints, stack dumps, generics) |
| 10 | 25-Oct-21 | Linear models |
| 10 | 27-Oct-21 | Other models |
| 11 | 01-Nov-21 | Unit testing |
| 11 | 03-Nov-21 | Local config, folders, projects |
| 12 | 08-Nov-21 | TBD |
| 12 | 10-Nov-21 | Class cancelled, work on your final |
| 13 | 15-Nov-21 | Wrap up |

Last update: 2021-Aug24