SCHOOL OF MEDICINE & DENTISTRY
UNIVERSITY of ROCHESTER MEDICAL CENTER

BST430: Introduction to Statistical Computing
Fall 2023

Room: SRB 1.404
Time: MW 9:10-10:50am
Office Hours: by appointment
Prerequisites: An advanced undergraduate course in Statistical Inference and some programming experience, or permission from the instructor

Instructors:

Matthew N. McCall, Ph.D.
Phone: (585) 273-3177
Email: matthew_mccall@urmc.rochester.edu

Donald Harrington, M.S.
E-mail: Donald_Harrington@urmc.rochester.edu

Hongyue Cookie Wang, Ph.D.
E-mail: Hongyue_Wang@urmc.rochester.edu

Shan Gao, M.S.
E-mail: Shan_Gao@urmc.rochester.edu

Andrea Baran, M.S.
Email: andrea_baran@urmc.rochester.edu

Course Description

Part I:

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. Students 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 research and analysis will build.

Part II:

This class is an introduction to the use of the SAS programming language for analysis of biomedical data. After an introduction to the SAS environment on a PC, SAS will be used to write programs for reading and processing data, and for performing descriptive and basic statistical analysis

Course Aims and Objectives

Part I:

- 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

Part II:

- Create and run SAS programs in a PC environment or SAS Studio
- Read raw data files in various formats and create SAS data sets
- Create new variables in the data step
- Use SAS procedures to describe data numerically and graphically
- Annotate SAS output with informative titles, labels, and formats
- Work with SAS data sets: sort, subset, merge, and re-format
- Use commonly used SAS procedures for statistical inference and modelling
- Export SAS data and output to other computers and software

Materials and Access

Part I:

Wickham & Grolemund "R for Data Science" https://r4ds.had.co.nz/

Part II:

Textbook: The Little SAS Book, 5th edition by Delwiche and Slaughter

Assignments and Grading Procedures

Part I: (75% of grade)

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 poll and quizzes delivered during lecture and asking questions during synchronous lectures. 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.
- An open book, no-collaboration-permitted final will be assigned on November 9 and will be due 24 hours later.

Part II: (25% of grade)
Class will consist of seven 1.5-hour lectures, covering SAS programming contained in the class notes. Computer assignments will be done by running SAS on a personal computer. Assignments are expected to be completed and will be graded. There will be no exam and the grade will be based on the assignments only.

Academic Integrity

Academic integrity is a core value of the University of Rochester. 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

The Biostatistics and Computational Biology department policy on appropriate student collaboration can be found using the following link:
https://www.urmc.rochester.edu/biostat/courses/studentcollaboration.aspx

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 exam, you **are not** generally permitted to copy and paste code, except where specifically indicated. For homework, you 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.
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. Use of AI-based tools (e.g. GitHub Copilot, ChatGPT, etc.) **are not** permitted to be used in this course.

Accommodations for Students with Disabilities

Students needing academic adjustments or accommodations because of a documented disability must 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

Week	Date	Planned Material
1	30-Aug-23	Syllabus, intro to github, intro to rstudio configuring github from rstudio, projects/repositories, rmarkdown, Intro to R (syntax)
2	4-Sept-23	NO CLASS
2	6-Sept-23	Styleguide, data from files, dplyr (i), ggplot (i)
3	11-Sept-23	Data structures in R
3	13-Sept-23	dplyr (ii)
4	18-Sept-23	Collaboration (merging, conficts, pull requests) with git/github
4	20-Sept-23	Factors
5	25-Sept-23	ggplot (ii), other graphing systems
5	27-Sept-23	Text manipulation, regex
6	2-Oct-23	Indexing, iteration, linear algebra
6	4-Oct-23	Functions, scope, functional iteration
7	9-Oct-23	classes and generics
7	11-Oct-23	Python via Ipython and recticulate
8	16-Oct-23	Computer science data structures (arrays, linked lists, trees, hashmaps)
8	18-Oct-23	Algorithms and complexity (linear search, binary search, recursion)
9	23-Oct-23	Debugging ii (breakpoints, stack dumps, generics) (Harrington)
9	25-Oct-23	Linear models
10	30-Oct-23	Other models
10	1-Nov-23	Unit testing
11	6-Nov-23	Local config, folders, projects
11	8-Nov-23	R package development
12	13-Nov-23	Matlab (Baran)
12	15-Nov-23	SAS environment, rules, variables, data step, input statements (Wang)
13	20-Nov-23	Matlab (Baran)
13	22-Nov-23	NO CLASS
14	27-Nov-23	Check data before data cleaning: PROC UNIVARIATE / MEANS / FREQ, Work within data step: Options, formats, labels and title/footnotes, dates, numeric and character functions, missing data (Gao)
14	29-Nov-23	Work within data step(continued): ARRAY, DO-LOOP, Restructure datasets: subsetting and merging dataset, PROC TRANSPOSE, PROC EXPAND, PROC DATASETS, PROC SQL for large-scale datasets (Gao)
15	4-Dec-23	Present and summarize data: PROC TABULATE / REPORT, PROC SGPLOT / SGPANEL (Gao)
15	6-Dec-23	Control output using ODS, SAS Macros (Gao)
16	11-Dec-23	Statistical Testing/Modeling 1: TTEST, ANOVA, GLM, NPAR1WAY (Wang)
16	13-Dec-23	Statistical Testing/Modeling 2: PROC LOGISTIC, PROC GENMOD, PROC MIXED (Wang)

Last updated: August 29, 2023