Course Name: Statistical Computing

Course Subject/Number: BST430-1

Credit Hours: 4

Term and Year Offered: Fall 2025

Day and Time: Wednesday1:30-3:10 PM & Friday 9:05-10:45 AM.

Dates: Aug 27 - Dec 5, 2025.

Class Location: SRB 1.406.

Course Director: Seong-Hwan Jun

Instructor(s): Seong-Hwan Jun, Hongyue Wang, Shan Gao, Donald Harrington

Contact information:

- seonghwan_jun@urmc.rochester.edu;
- Hongyue_Wang@urmc.rochester.edu;
- Shan_Gao@urmc.rochester.edu;
- Donald_Harrington@urmc.rochester.edu.

Office Hours: Wednesday 3:30-4:30 PM.

Course website: https://junseonghwan.github.io/teaching/bst430_fall25/

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

Course Description: Basic/intermediate R programming; statistical analysis in R; visualization in R; reproducible research and collaborative coding; command line tools and BlueHive; introduction to SAS programming; statistical analysis in SAS. Topics in statistical analysis provide working examples.

Course Aims and Objectives:

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. 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

Assignments and Grading Procedures: *(A-E)*

Part I: 75% of the grade. You will be evaluated in terms of 4 homework assignments (40%), 10 labs (20%), in-class quizzes/participation (10%), and a take-home final (30%).

Participation will be evaluated holistically and shall include completing quizzes and in-class activities delivered during lectures, and participating in discussions.

There will be 4 homework assignments to be completed within one week's time. These homework assignments are to be completed individually and out of class. There will be 10 lab activities. The labs can be completed in groups and are done primarily in class. Homework and labs will be posted and returned on GitHub.

An open-book, no-collaboration permitted final will be assigned on November 5th and will be due 24 hours later.

Part II: 25% of the grade. Class will consist of seven lectures, covering SAS programming. 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.

93-100%	*A*
90-92%	*A-*
87-89%	*B+*
83-86%	*B*
80-82%	*B-*
77-79%	*C+*
73-76%	*C*
70-72%	*C-*
60-69	*D*
<60%	*E*

Course Policies and Expectations:

Students in this course are expected to conduct themselves in an honest and ethical manner, as well as to respect the intellectual work of others.

1. In homework, and the take home final (anything not marked as a lab exercise) you **are not** generally permitted to copy and paste your classmates' code, except where specifically indicated. You may use Google search but must cite the source if you are transcribing verbatim. AI generated solutions are not permitted for this course. For homework, you may consult with your classmates and external resources on algorithmic and implementation details, but any code and writup you submit must have been typed into your editor. For the final exam, 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 acknowledgement in documentation and other scholarship is an appropriate way for attributing others' work (copyright requriements notwithstanding).

Grading homework and lab assignments:

- 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 tracks 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.

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 visit https://www.rochester.edu/graduate-education/academic-resources/regulations/

Accommodations and Access Services for Students with Disabilities:

Students needing academic adjustments or accommodations because of a documented disability must register with the Office of Disability Resources (link below)
https://www.rochester.edu/college/disability/current/register.html

Religious Holidays:

SMD policy and practice are that classes not be dismissed on religious holidays. It is expected, however, that students will not be penalized in any way for observing religious holidays. Instructors are asked not to schedule examinations on the same day as holidays. New York Education Law provides that students who choose not to attend classes or take exams on certain days because of their religious beliefs will be given an equivalent opportunity to make up the work requirements or exams they miss without penalties.

For a list of the major religious holidays, see the Interfaith Chapel's holidays page.

Bibliographic and Other Resources

We will use the following freely available books for Part I of the course:

- R for Data Science (2e): https://r4ds.hadley.nz/
- Advanced R: https://adv-r.hadley.nz/
- Modern Statistics for Modern Biology: https://www.huber.embl.de/msmb/
- R Packages (2e): https://r-pkgs.org/

The additional resources are provided on Git/Github, Coding style, and Genomic data analysis:

- HGWR: https://happygitwithr.com/
- OHGA: https://pmc.ncbi.nlm.nih.gov/articles/PMC4509590/
- OSCA: https://bioconductor.org/books/release/OSCA/
- Git for scientists: https://milesmcbain.github.io/git_4_sci/
- Tidyverse style guide: https://style.tidyverse.org/

Course Overview:

Week	Date	Lecture	Supplemental Readings	HW/Lab
1	Aug 27	Introduction to R and RStudio. Reproducibility: git and quarto.	R4DS Ch. 2,6 R4DS Ch. 28. HGWR Ch. 4-14.	LB00
1	Aug 29	R language fundamentals.	ADVR Ch. 2-5.	LB01
2	Sep 3	Tour of Tidyverse. Reading in data.	R4DS Ch. 7.	HW1 Git exercise
2	Sep 5	Data visualization. Base R graphics. Grammar of graphics.	R4DS Ch. 1,9-11.	LB02
3	Sep 10	Data wrangling I.	R4DS Ch. 3,5,12,13.	HW1 due LB03
3	Sep 12	Class cancelled: A conference in honor of David Oakes		
4	Sep 17	Data wrangling II.	R4DS Ch. 19.	LB04

4	Sep 19	Factors, and date times.	R4DS Ch. 16-17.	HW2
5	Sep 24	Text processing I.	R4DS Ch. 14-15.	LB05
5	Sep 26	Text processing II.		HW2 due LB06
6	Oct 1	Functions and iterations I.	R4DS Ch. 25-26. ADVR Ch. 6.	
6	Oct 3	Functions and iterations II. Styles. Debugging.	ADVR Ch. 9-11. Tidyverse style guide. ADVR Ch. 22.	HW3 LB07
7	Oct 8	Modeling I: linear models.		LB07
7	Oct 10	Modeling II: generalized linear models.		HW3 due LB08
8	Oct 15	Generative models for discrete data. Statistical modeling.	MSMB Ch. 1-2.	
8	Oct 17	Simulation (sampling) and testing.	MSMB Ch. 4,6.	HW4 LB09
9	Oct 22	Bioinformatics with R: Bioconductor. Clustering. Modeling counts data.	OHGA and OSCA. MSMB Ch 5. MSMB Ch. 8.	
9	Oct 24	Trees and graphs.	MSMB Ch. 10.	HW4 due LB10
10	Oct 29	R packages I: Fundamentals.	RPKG Ch. 1-9.	
10	Oct 31	R packages II: Testing Documentation Profiling & benchmarking.	RPKG Ch. 13-15. RPKG Ch. 16. ADVR Ch. 23-24.	R package development exercise
11	Nov 5	Class cancelled: Take home final exam.		
11	Nov 7	SAS programming and analysis.		
12	Nov 12	SAS programming and analysis.		
12	Nov 14	SAS programming and analysis.		
13	Nov 19	SAS programming and analysis.		
13	Nov 21	SAS programming and analysis.		
14	Nov 26	Class cancelled: Thanksgiving.		
14	Nov 28	Class cancelled: Thanksgiving.		
15	Dec 3	SAS programming and analysis.		
15	Dec 5	SAS programming and analysis.		