ST 495 Advanced computing for statistical reasoning; Spring 2024

Lecture: Tuesdays/Thursdays, 15:00–16:15, 2235 SAS Hall

Instructor: Dr. Jonathan P Williams
Email: jwilli27@ncsu.edu
Course website: https://jonathanpw.github.io/ST495
Office location: 5218 SAS Hall
Office hours: 16:30–18:00 Thursdays, 16:00–18:00 Fridays, or by appointment
Office phone: 919.513.0191

Teaching Assistant: Sahil Patel
Email: sspate27@ncsu.edu

Course Description: This is a capstone course designed to survey topics and tools needed for an undergraduate statistics major to begin to develop a broad and thorough working knowledge of modern computational techniques for statistical inferences. Statistical methods and the algorithms used to facilitate their computations are motivated by building logical foundations for statistical reasoning. Algorithms surveyed can broadly be categorized as either optimization-based or sampling-based. Rather than focusing on learning standard software packages for implementing common statistical routines, all codes will be written from scratch using the `R` programming language (or any other high-level language of the students' choosing). Emphasis is placed on developing a practical understanding of how and why existing methods work, and when to apply a particular method. Some programming proficiency is assumed.

Student Learning Outcomes:

1. Working knowledge of basic matrix arithmetic and data structure tools necessary for standard data analysis and statistical inference techniques such as linear/nonlinear regression and bootstrapping.
2. A practical understanding of the utility of likelihood functions and limit theorems insofar as they guide the intuition for approaches to statistical inference.
3. A practical understanding of gradient-based optimization algorithms and Monte Carlo sampling algorithms.
4. Proficiency in writing code in base `R`.
5. Demonstrated competency in generating synthetic data and the proper use of random number generator seeds.
6. The ability to design and implement a simulation study, to evaluate the performance of statistical estimation procedures on synthetic data.
7. Appreciation for the importance of reproducibility and replicability in the context of data analysis, and the principles of open-source software.
8. Exposure to varied types and consequences of missing data.
9. Exposure to high performance computing cluster environments and shell syntaxes.

Prerequisites: Introductory statistics (ST 311 and ST 312), Calculus 3 (MA 242), Introductory linear algebra (MA 305 or MA 405), and Introduction to Stat Programming – `R` (ST 308)

Optional Texts:
G. Givens and J. Hoeting (2013). *Computational statistics*, 2nd edition. John Wiley & Sons.
A. Hunt and D. Thomas (2000). *The pragmatic programmer: from journeyman to master*. Addison Wesley Longman, Inc.
J. Kloke and J. McKean (2015). *Nonparametric Statistical Methods Using R*. CRC Press.
W. Shotts, Jr. (2013). *The Linux Command Line*, 2nd internet edition.
L. Trefethen and D. Bau (1997). *Numerical linear algebra*. SIAM.

Grade Distribution:

Assignments	40%
Midterm project: simulation study part 1, evaluating estimators	25%
Final project: simulation study part 2, evaluating predictors	25%
Final presentation	10%

Letter Grade Distribution:

≥ 93.00	A	73.00 - 76.99	C
90.00 - 92.99	A-	70.00 - 72.99	C-
87.00 - 89.99	B+	67.00 - 69.99	D+
83.00 - 86.99	B	63.00 - 66.99	D
80.00 - 82.99	B-	60.00 - 62.99	D-
77.00 - 79.99	C+	≤ 59.99	F

For students taking the course as credit-only, S is equivalent to C- or better; otherwise U.
No requirements, procedures, or expectations apply to students choosing to audit the course.

Final exam period: 15:30–18:00 on Thursday, 25 April 2024 in 2235 SAS Hall

Personal note to students: Please do not feel intimidated about interacting with the me. Regardless of how busy or stressed I may appear to you, teaching your class is a part of my job, and I take that very seriously. I care deeply about the quality of your learning. Please always reach out to me if you have questions, concerns, or need help. I understand that it can be difficult and can even feel embarrassing to ask for help. However, I was once in your position, and I promise to always treat you with respect, empathy, and kindness. Nobody that ever did anything meaningful did so without first failing over and over again.

Course policies and commentary:

- **Assignments**
  - **Homework will be assigned each Tuesday evening, and will be due the following Tuesday at the beginning of class. Completed assignments must be turned in as script files on Moodle. The grader *will* run your code to assess your solutions**.
  - Each homework assignment will receive the same weight in the calculation of the final course grade (i.e., longer (shorter) assignments do not count for a larger (smaller) portion

of the overall assignment course grade). For each assignment, each exercise has the following point distribution:

  * 2 points – solution is correct
  * 1 point – solution is mostly correct
  * 0 points – solution is not relevant to the question, or the script file does not run.

- No late assignments will be accepted. Reach out to the instructor if you begin to fall behind! The lowest 2 assignment grades will be dropped from each students' course grade at the end of the semester.
- Take responsibility for understanding solutions to all assignments. For example, if you find a solution on StackExchange, then convince yourself that the solution is actually correct.
- **Learn to distinguish between the things you *do* know and the things you *do not* know** (this is one of the most important results of all education). To understand, to *a* particular degree, that a given statement is true means that you can explain why the statement is true, to *the* particular degree.

- **Projects**
  - Students work individually on the midterm and final course projects.
  - A detailed outline of the project requirements is provided on the course website.

Tentative Course Outline:

Week	Content
Week 1	• Install the `R` programming language • Review data structures, functions, control structures, matrix arithmetic, random numbers, plotting, and logic • Good coding practices from Hunt and Thomas
Week 2	• Review data structures, functions, control structures, matrix arithmetic, random numbers, plotting, and logic, continued • Lectures 1-2 of Trefethen and Bau
Week 3	• Linear algebra and the singular value decomposition • Lectures 4-5, 17, 20, 24-25 Trefethen and Bau
Week 4	• Least squares problems and projection matrices • Lectures 6 and 11 of Trefethen and Bau
Week 5	• Gram-Schmidt orthonormalization, QR factorization, and linear models • Lectures 7-8 of Trefethen and Bau, Section 4.4 of Kloke and McKean
Week 6	• Likelihood inference and maximum likelihood estimator (MLE) • MLE for linear regression parameters • Chapter 1 of Givens and Hoeting
Week 7	• Introduction to logistic regression • MLE for logistic regression parameters
Week 8	• Gradient descent algorithm and momentum, stochastic, and batch variants • Receiver operating characteristic (ROC) curve for binary classification • Chapter 2 of Givens and Hoeting
Week 9	• Confidence and prediction intervals for linear regression • **Midterm project due**
Week 10	• Spring break
Week 11	• Hypothesis testing and permutation tests
Week 12	• Bootstrapping • Chapter 9 of Givens and Hoeting
Week 13	• Bootstrapping, continued • Chapter 9 of Givens and Hoeting
Week 14	• Simulation and Monte Carlo integration • Chapter 6 of Givens and Hoeting
Week 15	• Missing data and the expectation–maximization (EM) algorithm • Chapter 4 of Givens and Hoeting
Week 16	• TBD • **Final presentation and project due**

NCSU Polices, Regulations, and Rules: Students are responsible for reviewing the NC State University Policies, Rules, and Regulations (PRRs) which pertain to their course rights and responsibilities, including those referenced both below and above in this syllabus:

- Equal Opportunity and Non-Discrimination Policy Statement https://policies.ncsu.edu/policy/pol-04-25-05 with additional references at https://oied.ncsu.edu/divweb/policies/
- Code of Student Conduct https://policies.ncsu.edu/policy/pol-11-35-01
- Grades and Grade Point Average https://policies.ncsu.edu/regulation/reg-02-50-03
- Credit-Only Courses https://policies.ncsu.edu/regulation/reg-02-20-15
- Audits https://policies.ncsu.edu/regulation/reg-02-20-04

Policy on Academic Integrity: Cheating, plagiarism and other forms of academic dishonesty will not be tolerated. Violations of academic integrity will be handled in accordance with the Student Discipline Procedures (NCSU REG 11.35.02).

Disability Services for Students: Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with the Disability Resource Office at Holmes Hall, Suite 304, 2751 Cates Avenue, Campus Box 7509, 919-515-7653. For more information on NC State's policy on working with students with disabilities, please see the Academic Accommodations for Students with Disabilities Regulation (NCSU REG 02.20.01).

Privacy: Students may be required to disclose personally identifiable information to other students in the course, via digital tools, such as email or web-postings, where relevant to the course. Examples include online discussions of class topics, and posting of student coursework. All students are expected to respect the privacy of each other by not sharing or using such information outside the course.