Syllabus

ST441/541 Fall 2018

Oct 08 2018

Syllabus

Course Name: Probability, Computing and Simulation in Statistics
Course Number: ST441/541
Course Credits: **4** This course meets for two 80 min lectures per week and one 50 min lab per week.
Prerequisites: ST422/522 or equivalent (at least two quarters of Mathematical Statistics)

Course Content

From the catalog

> Review of probability, including univariate distributions and limit theorems. Random-number generation and simulation of statistical distributions. Bootstrap estimates of standard error. Variance reduction techniques. Emphasis on the use of computation in statistics using the S-Plus or MATLAB programming language.

A guiding goal for the class is to have you to plan, organize, implement and communicate a computationally intensive statistical investigation.

You'll start by learning about some of the computational approaches to the same problems you solved on pen and paper in your Math Stat classes, like integrating to find moments, or differentiating to find maximum likelihood estimates. Along the way you'll brush up on the basics of programming in R and start to consider what makes code "good".

As your investigations become more complicated you'll learn best practices for scientific computing that make your work robust, reproducible and easy to share. You'll also learn computational techniques that help you write code that is correct, clear and, if necessary, fast.

The computational topics you'll learn are applicable to any programming language, but you'll explore and implement them in R. You'll inevitably spend some time learning R's individual idiosyncrasies, but I consider this time well spent. Learning one language in depth makes it easier to pick up another.

Unlike prior terms, **we will not** be using Matlab. (Is this a deviation from the description? Not really, *S-Plus* was a commercial implementation of the S programming language, *R* is the open source successor to S.)

Student Learning Outcomes

After successfully completing this course, you will be able to:

- Investigate properties of random variables through simulation.
- Break down a computational task into modular components and implement them in R.
- Balance the concerns of speed and clarity to write R code that is both efficient and understandable.
- Organize a computational project in a way that facilitates reproduction and collaboration.

(Very) Tentative Topic Schedule

Week 0 starting Sep 20 **Introduction**

- Computational versus analytical methods
- Floating point representation of numbers

Week 1 starting Sep 24 **Monte Carlo Methods I**

- Simulating random variables: pseudo-random numbers, inverse method, rejection sampling
- What makes code "good"?

Week 2 starting Oct 01 **Monte Carlo Methods II**

- Using simulation for estimation and integration
- Variance Reduction
- Functions
- Iteration: for loops versus functional programming

Week 3 starting Oct 08 **Organization**

- Best practices for scientific computing
- Handling larger simulations

Week 4 starting Oct 15 **Bootstrap**

- Putting things together so far

Week 5 starting Oct 22 **Optimization**

- Maximum Likelihood
- Convergence

Week 6 starting Oct 29 **Efficiency**

- Identifying bottlenecks
- Strategies for making code run faster

Week 7 starting Nov 05 **Clarity**

- Making code easy to understand and easy to use
- Functional programming
- Object Oriented programming: S3

Week 8 starting Nov 12 **Correctness**

- Documentation: projects and functions
- Testing for correctness

Week 9 starting Nov 19 **Communication**

- Sharing computational work
- Essentials for presentations

Week 10 starting Nov 26 **Project Presentations**

Learning Resources

All lecture notes, lab materials, homework assignments and additional resources will be posted on the class website: http://stat541.cwick.co.nz

I will use canvas to send announcements and record grades. You will use canvas to submit homework when it is ready for grading.

Lectures

My goal for our time in class is that it is very hands-on. That is, you spend a lot of it writing code and thinking about writing code. I will ask you to bring along a laptop if you can. Computing activites will generally be in small groups and each group will only need one laptop, so if you are unable to bring a laptop, get friendly with someone who can.

Being hands-on in class also means I will often assign a reading or activity to be completed **before** class. It is important to complete these before class so you are prepared for the in-class activities, and being prepared is part of your participation grade.

RStudio.cloud

For in class activities and labs we'll use rstudio.cloud, this way we will all be using exactly the same version of R, and R packages.

To get started:

1. Head to https://rstudio.cloud/ and log in either by creating an account or using your account on an existing service (google or github).
2. Join the class workspace by going to: ST541 workspace (this link will only work during the first week of class).

You should see a couple of existing projects including `russia-elections`, which we'll explore on the first day of class. To access the workspace in future, log in to rstudio.cloud, and find the workspace under *Spaces* on the left hand menu.

git and github

github will be our primary mode for sharing our work (including me providing you starter code, data or documents for homework). I have set up a private organization for us for this purpose. You will get set up during the first lab.

Before then, get a github account if you don't have one. Head to https://github.com and sign up for a free account (don't pay for one!). **But first,** read this advice on choosing a username.

Although not required for the class, you may also like to request a Student Developer Pack, which among other things gives you unlimited private repositories while you are a student.

Textbook

While there is no required textbook for the class, I will post readings (or assign them as homework) from mostly open and online resources.

In addition you may find the following books useful supplements:

- *R for data science: import, tidy, transform, visualize, and model data.* Wickham, Hadley, and Garrett Grolemund. O'Reilly Media, Inc., 2016.
- *Advanced R*. Wickham, Hadley. Chapman and Hall/CRC, 2014.
- *The R Inferno*. Burns, Patrick. Lulu.com, 2012.
- *A first course in statistical programming with R.* Braun, W. John, and Duncan J. Murdoch. Cambridge University Press, 2016.
- *Simulation* Ross, Sheldon M. Elsevier Science & Technology, 2012.
- *Simulation and the Monte Carlo Method* Rubinstein, Reuven Y., and Kroese, Dirk P. 3rd ed., Wiley, 2016.
- *Handbook of Monte Carlo methods.* Kroese, Dirk P., Thomas Taimre, and Zdravko I. Botev. Vol. 706. John Wiley & Sons, 2013

Evaluation of Student Performance

Your final grade will be a weighted combination of homework (50%), a project (40%) and class participation (10%).

Homework (50%): Weekly homework will be released on the class website on Fridays and due the following Thursday at midnight. You may discuss ideas with other students, but you must write up your homework without assistance and on your own. Late homework will not be accepted without prior arrangement with the instructor. Your lowest homework score will be dropped.

Project (40%): You will undertake an **individual** project that includes a substantial component of statistical computing. The final deliverable will include a written report, a github repository and (optionally for ST5441 students) an oral presentation. Incremental deadlines will occur throughout the quarter to help you make consistent progress.

Class participation (10%): Particpation will be assessed based on three components: attendance, preparation and contribution. A perfect participation score comes from showing up to lecture on-time (or letting me know you can't make it and how you plan to catch up), having completed any required pre-lecture activities (e.g. readings or computing tasks) and actively contributing to small group activities in class.

Grading Scale

Letter grades will be assigned according to the following scheme:

Percent	Grade
95 – 100	A
88 – 94.9	A-
80 – 87.9	B+
75 – 79.9	B
70 – 74.9	B-
65 – 69.9	C+
60 – 64.9	C
55 – 59.9	C-
45 – 54.9	D
0 – 45	F

Student Conduct

Students are expected to be honest and ethical in their academic work. Please read the full text of the University Student Conduct Code at http://studentlife.oregonstate.edu/code to understand what constitutes academic dishonesty under OSU policy. Any incidents of academic dishonesty will be dealt with as outlined in the University's

Academic Regulations.

The Student Conduct Code defines academic dishonesty as:

> Any action that misrepresents a student or group's work, knowledge, or achievement, provides a potential or actual inequitable advantage, or compromises the integrity of the educational process.

Examples include, but are not limited to, the following:

- copying another student's homework assignment
- copying another student's exam
- using prohibited materials (e.g., cell phone) during an exam
- communicating with another student during an exam
- changing answers on an exam after the exam has been graded
- unattributed use of material copied from an article, textbook, or web site
- continuing to write on an exam after the instructor or TA has asked for the exams to be handed in

Statement Regarding Students with Disabilities

Accommodations for students with disabilities are determined and approved by Disability Access Services (DAS). If you, as a student, believe you are eligible for accommodations but have not obtained approval please contact DAS immediately at 541-737-4098 or at http://ds.oregonstate.edu. DAS notifies students and faculty members of approved academic accommodations and coordinates implementation of those accommodations. While not required, students and faculty members are encouraged to discuss details of the implementation of individual accommodations.