Syllabus

Instructor: Susan Vanderplas
E-mail: `susan.vanderplas@unl.edu`
Office: Hardin 343D
Student Hours: Schedule here

Time: Wednesdays 1-2:50
Location: Hardin 142-East

Course Description

Introduction to programming for statistical analysis. Covers basic programming concepts necessary for statistics, good computing practice, and use of built-in functions to complete basic statistical analyses.

Course Objectives

At the end of this course, students will

1. Be comfortable using R and/or python for statistical analysis
2. Write basic programs using logic including for loops and control structures
3. Implement basic data analyses in R or python
4. Describe the sequence of steps necessary to solve a simple problem
5. Be familiar with good computing practices, such as version control and documentation

Textbook

The primary textbook for this course is one that I have developed from other resources listed below. It is available for free at https://srvanderplas.github.io/stat-computing-r-python/. It is under construction/a work in progress, so it may be hard to work more than a couple of weeks ahead in this class using the primary textbook.

In addition, you may find it useful to reference some of the following resources that I have consulted while assembling the textbook. Most are available online for free, though some require an institutional email address.

- R for Data Science
- Python for Everybody
- Python for Data Analysis - Available online for free if you register with your UNL email address.
- Python Data Science Handbook - Available online for free if you register with your UNL email address.
- Advanced R

Class Schedule & Topic Outline

This schedule is tentative and subject to change. Students are expected to read the corresponding textbook chapter (linked in Canvas) **before coming to class**. For the most part, for each week's topic, there will be one or two corresponding textbook chapters.

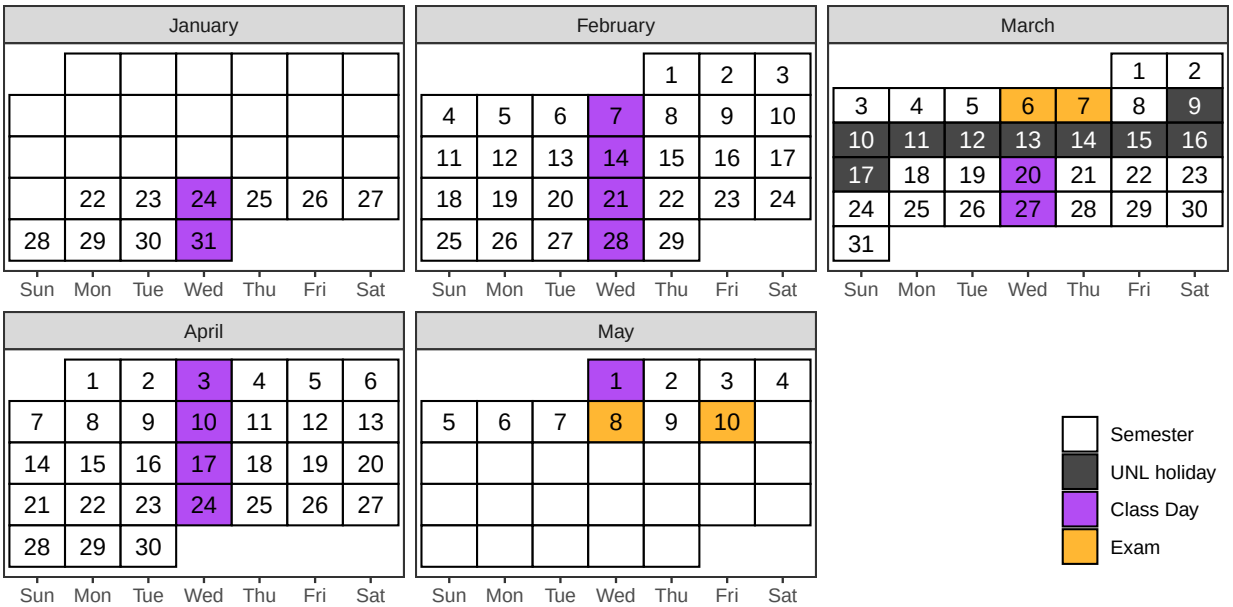

Figure 1: Course Calendar

Table 1: Tentative schedule of class topics

Date	Time	Topic
Jan 24		Getting Started
Jan 31		Scripts & Notebooks
Feb 7		Intro to Programming
Feb 14		Data Types
Feb 21		Data Structures
Feb 28		Control Structures
Mar 6	In class	Midterm Assigned
Mar 7	6pm	Midterm Due
Mar 20		Functions
Mar 27		Version Control
Apr 3		Data Visualization
Apr 10		Data Cleaning
Apr 17		Strings
Apr 24		Reshaping Data
May 1		Graphics
May 8	In class	Final Assigned
May 10	6pm	Final Due

Course Policies

Assessment/Grading

Assignments	Weight
Reading Quizzes	10%
Weekly Homework & Participation	50%
Midterm Exam	20%
Final Exam	20%

Lower bounds for grade cutoffs are shown in the following table. I will not "round up" grades at the end of the semester beyond strict mathematical rules of rounding.

Letter grade	X +	X	X -
A	97	94	90
B	87	84	80
C	77	74	70
D	67	64	61
F	<61		

Interpretation of this table: A grade of 85 will receive a B. A grade of 77 will receive a C+. A grade of 70 will receive a C-. Anything below a 61 will receive an F.

Exams

Two take-home exams will be given during the course of the semester, on dates noted on the Tentative Course Outline on the last page.

Exams will require that you write code to solve problems utilizing the course material. You are expected to take exams at the scheduled times. If this is impossible due to extreme circumstances (illness, death in the family, previously scheduled activities vital to academic program), please notify me and provide appropriate documentation. No make-up exams will be given if I am not notified at least 24h prior to the exam release date. You will be required to obtain a note from your physician or adviser explaining the nature of the conflict.

Homework

Approximately 8-12 homework assignments will be made over the course of the semester. You will typically have one week to work on each of the assignments. The only way to learn statistics is to practice working problems, and homework is therefore an essential part of the course. Homework must be submitted in the file format specified, and should run or compile as submitted.

Participation

All students are expected to attend and fully participate in class activities. Participation will be determined based on a combination of class attendance and activities.

Late Policy

Late assignments will be accepted only under extenuating circumstances, and only if you have contacted me **prior** to the assignment due date and received permission to hand the assignment in late. I reserve the right not to grade (or to assign a 0 to) any assignments received after the assignment due date.

You may submit an assignment that was not submitted in time as a resubmission, assuming it meets the conditions outlined in the next section.

Resubmission Policy

Learning how to program is a process that is ongoing. I want you to not be afraid to make mistakes. As a result, I allow you to re-submit homework over the course of the semester in order to make corrections and respond to my initial feedback.

This policy is subject to the following constraints:

- You may not resubmit an assignment for which solutions have been posted.
- Each assignment should be resubmitted only once
- Resubmits are due on the day before the next exam is released. I reserve the right to set a rolling due date (e.g. 3 weeks after the original due date) if grading resubmitted assignments becomes too overwhelming.
- Resubmitted assignments will be graded at my convenience.

In general, this policy means that I will typically not post assignment solutions, though you are welcome to visit office hours to ask questions about an assignment.

I spend the extra time grading resubmissions because it reinforces your learning; this policy is not designed to ensure that you can get a better grade in this class. This resubmission policy is subject to adjustments throughout the semester to ensure that this objective is met. I reserve the right to limit or penalize resubmissions if it becomes clear that the current policy does not promote learning of the material and/or support the course objectives.

Attendance

You are expected to attend class. Consistent, repeated failure to attend class or actively participate in the course will affect the participation portion of your grade.

If you are feeling ill, please **do not come to class**. Instead, review the material and work on the homework assignment, and then schedule an appointment with me to meet virtually. This appointment (and your email notifying me about missing class) will substitute for the participation grade for the missed class.

Inclement Weather

If in-person classes are canceled, you will be notified of the instructional continuity plan for this class by Canvas Announcement. In most circumstances where there is power in the Lincoln area, we will hold class via Zoom if the university cancels class.

Expectations

You can expect me to:

- reply to emails within 48 hours during the week (72 hours on weekends)
- be available in class to assist with assignments
- be available by appointment for additional help or discussion

I expect you to:

- Read the module material and watch the videos before coming to class
- Engage with the material and your classmates during class
- Seek help when you do not understand the material
- Communicate promptly if you anticipate that you will have trouble meeting deadlines or participating in a portion of the course.
- Do your own troubleshooting before contacting me for help (and mention things you've already tried when you do ask for help!)
- Be respectful and considerate of everyone in the class

Make Mistakes!

Programming is the process of making a series of silly or stupid mistakes, and then slowly fixing each mistake (while adding a few more). The only way to know how to fix these mistakes (and avoid them in the future) is to make them. (Sometimes, you have to make the same mistake a few dozen times before you can avoid it in the future). At some point during the class, you will find that you've spent 30 minutes staring at an error caused by a typo, a space, a parenthesis in the wrong place. You may ask for help debugging this weird error, only to have someone immediately point out the problem... **it is always easier to see these things in someone else's code**. This is part of programming, it is normal, and you shouldn't feel embarrassed or apologize (unless you put no effort into troubleshooting the problem before you asked for help)

If you manage to produce an error I haven't seen before, then congratulations! 👏 You have achieved something special, and that achievement should be celebrated. 🎉 Each new and bizarre error is an opportunity to learn a bit more about the programming language, the operating system, or the interaction between the two.

Assignment Evaluation Criteria

In every assignment, discussion, and written component of this class, you are expected to **demonstrate that you are intellectually engaging with the material**. I will evaluate you based on this engagement, which means that technically correct but low effort answers which do not demonstrate engagement or understanding will receive no credit.

When you answer questions in this class, your goal is to show that you either understand the material or are actively engaging with it. If you did not achieve this goal, then your answer is incomplete, regardless of whether or not it is technically correct. This is not to encourage you to add unnecessary complexity to your answer - simple, elegant solutions are always preferable to unwieldy, complex solutions that accomplish the same task.

While this is not an English class, grammar and spelling are important, as is your ability to communicate technical information in writing; both of these criteria will be used in addition to assignment-specific rubrics to evaluate your work.

With the proliferation of AI tools such as Chat-GPT, I reserve the right to replace exam and homework grades with grades based on an oral discussion of your submitted solutions. If you cannot explain why your solution is correct, or the logic behind your solution, then you will not receive credit. Chat GPT can be a useful tool, but this course's objectives are meant to assess your ability to program in R and Python, not your ability to use AI systems. Fundamentally, you need to understand concepts and also develop some syntactic fluency to program effectively; generative AI solutions may hamper both of these goals if you rely on them too much.

Academic Integrity and Class Conduct

You will be engaging with your classmates and me through in-person discussions and collaborative activities. It is expected that everyone will engage in these interactions civilly and in good faith. Discussion and disagreement are important parts of the learning process, but it is important that mutual respect prevail. Individuals who detract from an atmosphere of civility and respect will be removed from the conversation or the classroom.

Students are expected to adhere to guidelines concerning academic dishonesty outlined in Article III B.1 of the University's Student Code of Conduct. The Statistics Department academic integrity and grade appeal policy is available here.

You must be able to explain how the logic works for any code you turn in. This means that code you obtained from e.g. StackOverflow is fine to use if you can explain it and modify it for the purposes of this class, but if you cannot explain your code you will not get credit for the assignment. This is in line with what is generally considered acceptable behavior in programming - reuse is fine (subject to the code's license) but you must be able to fully explain and modify any code you did not write yourself.

Required University Information

See https://executivevc.unl.edu/academic-excellence/teaching-resources/course-policies.