Syllabus

SDS 270: Programming for Data Science in R, Spring 2026

Overview

Contact

- Instructor: Jericho Lawson
- Email: jlawson01@smith.edu
- Office Hours:
  - Mondays, 3-4pm
  - Fridays, 10am-12pm
  - McConnell 207

Location/Time

- Mondays/Wednesdays, 10:50am - 12:05pm in Bass 002

Description

Advanced programming techniques for data science using R. This course is not about data analysis—rather, students will learn the R programming language at a deep level. Topics may include data structures, control flow, regular expressions, functions, environments, functional programming, object-oriented programming, debuggging, testing, version control, documentation, literate programming, code review, and package development. The major goal for the course is to contribute to a viable, collaborative, open-source, publishable R package. Prerequisite of SDS 192 and CSC 110/111, or equivalent.

Learning Outcomes

By the end of this course, students will be able to do the following:

- **Write** R code that uses various deep-level ideas, such as regex, functions, objects, and documentation.
- **Contribute** to an open-source software project using version control.
- **Develop** and **debug** sophisticated functions in R.
- **Create** a robust, encapsulated software package in the R programming language.

Materials

Textbooks

- Wickham, H. (2014). *Advanced R (2nd ed.).* CRC Press. https://adv-r.hadley.nz/
- Wickham, H., & Bryan, J. (2023). *R Packages (2nd ed.).* O'Reilly Media. https://r-pkgs.org/

Technology

- For coding: R, RStudio
  - For downloading/installing: Instructions
- For assignment turn-ins: GitHub
- For course management: Moodle, but will be used seldomly
- For course materials: This website
- For discussions and announcements: Slack

Grading and Expectations

Breakdown

Formative		**Summative**	
In-Class Exercises	10%	Exams (2)	30%
Problem Sets	20%	Package	40%

A traditional grading system will be used here. Cumulative numerical averages of 90-100 are guaranteed at least an A-, 80-89 at least a B-, and 70-79 at least a C-. The exact ranges for letter grades will ultimately be determined at the end of the course.

Classes and In-Class Exercises (10%)

Each class will consist of some form of the following: mini-lectures, demos, exercises, discussions, and package work time. Expect an experiential environment, where you will be working with code and concepts during the classes. As such, each student must **bring their laptop to class.**

Students are expected to participate during lecture by **asking questions** and **working with other students.** In-class activities will be submitted by the end of class through a GitHub repository and are graded for completion and accuracy. To complete these in-class activities, it is **required** that you show up for class unless extenuating circumstances apply.

Weekly readings will be assigned to prepare you for content that will be talked about in class. Even though there will be no grade associated with these readings, it is **strongly recommended** to come to the following week's classes with these readings completed.

Problem Sets (20% total)

Problem sets will be assigned bi-weekly to provide practice related to the current topics. These problems will involve conceptual exploration of topics, as well as solving through extended problems that tie in multiple topics at once.

Similar to the in-class activities, the problem sets will be turned in through a GitHub repository. Questions are graded based on accuracy.

Grade drop: Your lowest problem set grade will be dropped from your problem set total.

Exams (30% total)

Two exams will be given out in-class to demonstrate your knowledge on the course content. The first exam will be a closed-note written exam, while the second exam will be an open-note, open-laptop exam. These exams may include an oral portion, where you will have to explain particular concepts in a conversation-style setting.

Expect questions to be similar in vein to the problem sets.

Package (40%)

The overall objective of this course is to develop a new R package with a team of your choice. Throughout the second half of the course, you will receive feedback on the functionality, clarity, robustness, and readability of your code and documentation through a series of checkpoints.

Despite the unique opportunity to create a package in R, it may seem daunting to do so at first. The checkpoints and mini-assignments will pace yourselves as we prepare for the final package. Beneath are the following:

- **Proposal and Outline**: due Monday, 3/30 at 11:59pm
  - Develop a one-page proposal that summarizes the package you want to create and/or modify with appropriate motivations, references, and a course of action.
  - Create an outline that mimics the potential description file you will ultimately develop.
- **Checkpoint #1**: due Monday, 4/13 at 11:59pm
  - Develop an R package that meets minimum requirements.
- **Rehearsal**: given during alloted times on Monday, 4/20
  - Prep for demonstration
- **Checkpoint #2**: due Friday, 4/24 at 11:59pm
  - Develop an R package that advances on progress from the first checkpoint, including more functions, datasets, and/or descriptions.
- **Demonstration**: given in-class on Wednesday, 4/29
  - Showcase and run through a 5-8 minute demonstration on your R package.
- **Whole package**: due Wednesday, 5/6 at 11:59pm
  - Complete the R package and make any refinements.

More details to come in later weeks.

Policies

Preparation and Attendance

As a four-credit course that meets 4.5 hours per week, Smith College expects students to dedicate at least 7.5 hours per week towards the course outside of class. The assignments, readings, and assessments are designed with this target in mind. Expect to read through lecture notes, examples, articles, and forums outside of class along with the assignments.

Extensions

You will be granted **2 late days** to use on problem sets, with a maximum of one day used on each problem set. Additionally, you will be granted **2 late days** to use for checkpoints during R package development (excluding the demonstration and whole package). No need to inform me that you intend to take these late days. Beyond this, late assignments will not be accepted without an accommodation from a class dean or from the ARC.

Note that this policy does not apply to activities, readings, exams, or final package/presentation submission.

Academic Honesty

As a student at Smith College, the college expects all students to be honest and committed to the principles of academic and intellectual integrity in preparation and submission of all course work and examinations, as outlined by the Academic Integrity Board (AIB). The AIB provides an Academic Integrity Statement, which all students are expected to abide by. Any cases of academic dishonesty or plagiarism will be reported to the Academic Honor Board. Examples of these behaviors include:

- Submitting work completed by another student as your own.
- Copying and pasting words from sources without quoting and citing the author.
- Paraphrasing material from another source without citing the author.
- Failing to cite your sources correctly.
- Falsifying or misrepresenting information in submitted work.
- Paying another student or service to complete assignments for you.
- Submitting work generated by artificially intelligent tools such as ChatGPT without permission or instruction to do so.

You are encouraged to discuss course material, including assignments, with your classmates. All work you turn in, however, must be your own. This includes both writing and code. Copying from other students, from books, or from websites (1) does nothing to help you learn how to program, (2) is easy for us to detect, and (3) has serious negative consequences.

Generative AI

As mentioned in the Academic Integrity Board, the professor for each course decides whether and how students are allowed to use generative AI in a given course. For this specific course, the use of generative AI to complete assignments or produce content for this course is on an **assignment-by-assignment basis**, but will be *strictly prohibited* on in-class assignments and any final package submissions (i.e. demonstration, final package).

As an upper-division course, you have developed skills to code through basic tasks and processes in data science, as well as to analyze particular behaviors and patterns in the results. Generative AI can be useful as a tool to refine material and gather information about particular parts of code; however, the more you rely on AI, the more you lose out on the ability to think like a data scientist. The human element will be critical in succeeding in data science, and as such, appropriate considerations need to be made when using generative AI on assignments.

Specifically, here are the following permissions and constraints regarding generative AI on assignments and tasks in this course:

- **Activities, problem sets, and readings:** Generative AI can be used for pre-writing activities and code editing. Use of generative AI for writing any length of written content, composing any length of code, or synthesizing content are prohibited. Code and prose should be written in your own style. Any permitted use of generative AI should be cited in areas in which it was used.
- **Exams:** Generative AI is strictly prohibited on all exams.
- **Package:** Generative AI for use in development of ideas, pre-writing, and structuring is permitted, particularly in the early phases of development. However, any formal writing of documentation and code is not permitted.

Any unauthorized use of generative artificial intelligence in this course will be considered a case of academic dishonesty/plagiarism and will be reported to the Academic Honor Board. Some assignments will also contain modifications to these rules, so be mindful of these rules for each assignment.

Community & Support

Code of Conduct

As the instructor for this course, I am committed to making participation in this course a harassment-free experience for everyone, regardless of level of experience, gender, gender identity and expression, sexual orientation, disability, personal appearance, body size, race, ethnicity, age, or religion. Examples of unacceptable behavior by participants in this course include the use of sexual language or imagery, derogatory comments or personal attacks, trolling, public or private harassment, insults, or other unprofessional conduct.

As the instructor I have the right and responsibility to point out and stop behavior that is not aligned to this Code of Conduct. Participants who do not follow the Code of Conduct may be reprimanded for such behavior. Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the instructor.

All students and the instructor are expected to adhere to this Code of Conduct in all settings for this course: seminars, office hours, and over Slack. This Code of Conduct is adapted from the Contributor Covenant, version 1.0.0, available here.

Principles of Community

Whether in a class, college, or neighborhood setting, achieving a warm community is essential to your well-being. In this class, I hope we can foster a collaborative and welcoming environment: one that celebrates successes, respects individual strengths and weaknesses, demonstrates compassion for each other's struggles, and affirms diverse identities.

To establish this, consider the following:

- Check-in with colleagues before starting collaborative work.
- Consider when to step up and when to step back in class discussions, creating space for others to contribute. Listening is just as important to community-building as speaking.
- Acknowledge what we do and don't know, as well as how our colleagues experience the world.
- Support colleagues that may be stepping outside of their comfort zone (i.e. presentations).
- Ask questions often in our Slack workspace. Help each other out by answering questions when you can.
- Admit mistakes. They happen, and I will certainly make mistakes in class.
- Use pronouns. This provides a foundation to a safe, respectful classroom environment that creates a sense of trust. For information on pronouns and usage, please see the Office of Equity and Inclusion link here: Pronouns

Accommodations

It is my goal for everyone to succeed in this course. If you have personal circumstances that may impact your experience of our classroom, I encourage you to contact the Accessibility Resource Center in College Hall 104 or at arc@smith.edu. The Center will generate a letter that indicates to me what kind of support you need and how I can make your classroom experience more accommodating. Once you have this letter, you are welcome to visit my office hours or email me to discuss ideas about how we can tailor the course accordingly. While you can request accommodations at any time, the sooner we start this conversation, the better. If you have concerns about the course that are not addressed through ARC, please contact me. At no point will I ask you to divulge details about your personal circumstances to me.

Student Well-Being

College life is stressful, and life outside of college can be overwhelming. It is my position that attending to your physical and mental health and well-being should be a top priority. I will remind you of this often throughout the semester. I encourage you to schedule a time to talk with me if you are struggling with this course. If you, or anyone you know, is experiencing distress, there are numerous campus resources that can provide support via the Schacht Center.

Additional resources and support offered by the college are listed below:

- Accessibility Resource Center (ARC)
- Spinelli Center: Support for students doing quantitative work. Includes tutoring and resources.
  - For this class: Sun-Thurs, 7-9pm in Sabin-Reed 301.
  - Email qlctutor@smith.edu for specific request for help.
- Crisis Resources
- Counseling Services
- Wellness Resources
- Gender Identity and Expression
- Discriminatory Harassment

Course Outline

Week	Dates	Topics	Ch.	Assignments / Notes
1	1/26-30	Introduction (R, GitHub), Data Structures	AR3	
2	2/2-6	Lists, Data Frames, Assignment	AR2, 4	
3	2/9-13	Control Flow, Functions	AR5, 6	
4	2/16-20	Functions, Iteration	AR5, 6, 9	
5	2/23-2/27	Functionals, Review	AR9	Exam #1: Wed. 2/25
6	3/2-3/6	Environments, Packages	AR7, RP1-3	
7	3/9-13	Conditions, OOP	AR8, 12	
8	3/16-20	Spring Break!		*No class: Mon. 3/16; Wed. 3/18*
9	3/23-27	S3, Package Foundation	AR12, RP3	
10	3/30-4/3	Description, `usethis`	RP9-12	Exam #2: Mon, 3/30 Proposal: due Mon, 3/30
11	4/6-10	Code, Data, Manuals, Testing	RP6, 7, 16	
12	4/13-17	Testing, Vignettes	RP13, 17	Checkpoint #1: due Mon, 4/13
13	4/20-24	Project Time		Checkpoint #2: due Fri, 4/24
14	4/27-5/1	Project Time, Demonstrations	NA	
Finals	5/2-8	Finals	NA	Package: due Wed, 5/6

- Note that the assignments and topics covered are tentatively scheduled and may be altered slightly as the quarter progresses. The instructor has the right to modify the syllabus if needed. If this occurs, students will be notified before the change occurs.