LSCI 202A: Computational Skills for Language Science Research

Connor Mayer

Fall 2021

E-mail: cjmayer@uci.edu

Lecture times: MWF 1pm-1:50pm
Lecture Zoom: https://uci.zoom.us/j/94218178989

Office Hours: M 2pm-3pm
Office Hours Zoom: https://uci.zoom.us/j/7250073257

Website: https://canvas.eee.uci.edu/courses/39480

Course Description

This course teaches fundamental computational skills for use in Language Science research. We will touch on basic data analysis in R, command line tools, management of data and code using version control systems, text processing using regular expressions, software development best practices, software for running experiments, and academic websites. It would be impossible to treat all of these topics in detail in a ten week course. Instead, the intent is to provide you with some basic skills in each of these domains, as well as knowledge of what kinds of tools and resources are available to you and where to look if you want to learn more.

The primary goal of this class is to help you be an effective researcher as a grad student and beyond. This means that I want to make this class as useful to you as possible. The course content listed on the syllabus covers a range of skills that I think will be valuable. However, if there are specific topics you want to learn more about or if you have data you're not sure how to analyze, it's likely that we can modify the schedule to accommodate your requests.

Course Format

Lectures and office hours for this class will mainly be held over Zoom. We may have occasional in-person lectures or office hours as appropriate.

Prerequisites

This course is intended for graduate students in the Department of Language Science at UCI. Some basic familiarity with programming is assumed.

Learning Outcomes

- Given a data set, students should be able to use R to perform exploratory data analysis, visualization, and fit and interpret simple linear models.
- Students should be able to use basic command line tools, including tools to connect to remote servers and command line text editors.
- Students should be able to write regular expressions to match and extract substrings from a text file.
- Students should understand the purpose of version control systems, be able to use basic `git` commands to create and manage a repository, and be able to use Github to host and share repositories.
- Students should be able to identify some basic tenets of effective software design and apply them to their own code.
- Students should understand the purpose of an academic website, as well as how to create and host one of their own.
- Students should be able to identify additional resources to learn more about the above topics.

Course Materials

Readings

You do not need to purchase any textbooks for this course. All materials (assignments, notes, readings) will be distributed through the course website on Canvas or are available from free online resources.

Readings are optional, but *strongly* recommended. They are generally fairly short, and reading them before class will substantially improve your comprehension of the material.

The abbreviations for readings used in the course schedule are as follows:

- R for Data Science (R4DS): A free, online textbook for doing data science in R.
- The Missing Semester of Your Computer Science Education (MSCSE): Free, online course material from MIT that teaches "computing ecosystem literacy," which is often left out of a traditional CS education. Many of the topics it covers are directly relevant to the content of this course.

- The Pragmatic Programmer (PP): One of the foundational texts of responsible software development (another fine book in this area is Code Complete). This book is not free, but excerpts of certain chapters are. We will read one of these excerpts when we talk about writing modular code in Week 9.
- R Markdown: The Definitive Guide (RMDG): A free, online textbook for learning R Markdown. We will use R Markdown to create academic websites in Week 10.

Software

Most of the homework exercises will involving writing or modifying small R scripts. You will need access to a laptop or desktop computer with R Studio installed. You will also need software to connect to a remote server, such as command line tools like `ssh` or GUIs like PuTTY. We will cover how to install and use these programs in class.

Requirements and grading

The grades for the course are broken down as follows:

Component	Proportion of grade
Homework exercises	100%

Homework exercises (100% of final grade)

The course grade will be calculated based on five equally-weighted homework exercises. These will include writing scripts in R to perform data analysis or visualization, writing programs that use simple regular expressions to parse text data, short tasks that demonstrate understanding of command line and version control tools, creating an online experiment, and creating and publishing an academic website.

Students are permitted (encouraged, even!) to collaborate on homework assignments. If you do collaborate, please list at the top of your assignment all of the people you've collaborated with. Jointly authored assignments are not allowed.

There are no tests or exams.

Numeric and letter grades

Letter grades are calculated from numeric grades as follows:

Numeric grade	Letter grade
$\geq 90\%$	A
$\geq 80\%$	B
$\geq 70\%$	C
$\geq 60\%$	D
$< 60\%$	F

I reserve the right to scale final grades if I think it is necessary. I will only scale grades up: that is, your final grade can only *improve* as the result of scaling.

Getting help

- The first place you should seek help is using the discussion board on Canvas. If you have a question, it's likely that someone else has the same question. Posting on the discussion board allows everyone to see the answer. I also strongly encourage you to try to answer your peers' questions on the discussion board. This gives you valuable practice engaging with the course material, utilizing online resources, and synthesizing information, all of which will serve you well down the road.
- The second place you should come for help is my office hours. Please feel free to drop by as frequently as you like, even if you don't have any specific questions and you just want to work on an exercise or chat.
- If neither the discussion board or office hours are viable, you can email me with questions or concerns. I will reply to you within 24 hours.
- In certain circumstances I may be willing to arrange a meeting with you outside of normal class times and office hours. For the sake of my schedule (and yours!), please consider this a last resort, and do your best to seek help using the resources in the previous three points.

Academic integrity

All students are expected to adhere to the UCI Academic Dishonesty Policies (for more information, please visit https://aisc.uci.edu/students/academic-integrity/index.php).

Disability

Any student requesting academic accommodations based on a disability is required to apply with Disability Service Center at UCI. For more information, please visit http://disability.uci.edu/.

Acknowledgements

Thanks to Richard Futrell for input from previous iterations of the course.

Course Schedule

Week	*Date*	*Topic*	*Readings*	*Deadlines*
0	9/24	Introduction		
1	9/27	Intro to R: Workspace setup, Tidyverse	R4DS Chs. 1, 6	
	9/29	Visualization in R: Simple plots	R4DS Chs. 2, 3	
	10/1	Visualization in R: Transformational plots	R4DS Chs. 2, 3 Wickham 2010	
2	10/4	Data wrangling in R: Workflow	R4DS Chs. 4, 5	
	10/6	Data wrangling in R: Tidy data	R4DS Chs. 9–11	
	10/8	Data wrangling in R: Tidy data	R4DS Chs. 12	
3	10/11	Data wrangling in R: Relational data	R4DS Ch. 13	**Exercise 1 due**
	10/13	Data wrangling in R: Factors	R4DS Ch. 15	
	10/15	Data analysis in R: Exploratory analysis	R4DS Ch. 7	
4	10/18	Data analysis in R: Linear models	R4DS Chs. 22, 23	
	10/20	Data analysis in R: Linear models	R4DS Ch. 24	
	10/22	Data analysis in R: Linear models	R4DS Ch. 25	
5	10/25	Command line: Setup, `ssh`		
	10/27	Command line: Basics	MSCSE Topic 1	
	10/29	Command line: `vim` and `scp`	MSCSE Topics 3, 5	
6	11/1	Text processing: Intro to regex	Regex Cheat Sheet	**Exercise 2 due**
	11/3	Text processing: Command line tools	MSCSE Topic 4 Unix for Poets	
	11/5	Text processing: R	R4DS Ch. 14	
7	11/8	Version control: Overview	Git Magic Ch. 1	
	11/10	Version control: Github	Github Cheat Sheet	
	11/12	Version control: More git	Git Magic Ch. 2 MSCSE Topic 6	
8	11/15	Experiment software: Platform overview		**Exercise 3 due**
	11/17	Experiment software: PCIbex	PCIbex Docs	
	11/19	Experiment software: PCIbex	PCIbex Docs	
9	11/22	Software development: Best practices	PP Ch. 9	
	11/24	Software development: Testing code	R Packages Ch. 12	
	11/26	*Thanksgiving (no class)*		
10	11/29	Academic websites: What, why, and how?		**Exercise 4 due**
	12/1	Academic websites: R Markdown	RMDG Ch. 2	
	12/3	Academic websites: R Markdown	RMDG Ch. 10	
11	12/6	Exam week		
	12/8	Exam week		
	12/10	Exam week		**Exercise 5 due**