1 KEY INFORMATION

- Tuesday and Thursday 2:00-3:30 pm
- Buchanan, D322
- Instructor: Daniel Saunders (call me Daniel)
- Email: dsaunder@mail.ubc.ca
- Office hours: by appointment, on zoom or in-person.
- Teaching Assistant: Natalie Hanna, email: nataliehanna101@hotmail.com

2 DESCRIPTION

The aim of this course is to introduce and practice core skills in the computational modeling of behavior. The early historical figures in cognitive science all shared a basic methodological move - if you want to understand human behavior, you need to construct a model capable of reproducing key features of our behavior. If you want to know how people play chess, build a computer program that plays chess and study the similarities and differences between human and machine play styles. If there is a close match, that is a clue that humans solve chess problems in a similar way to the model. If you want to understand memory, build a computer program that remembers and forgets things. See if you can make the program forget things in the same way people do. Once you do, that's a clue humans use a similar process in memorization.

The skills involved in computational modeling are usually spread throughout the university - math, programming, psychology, statistics. This can make it hard to really see how all the skills fit together. The skills are also hard. Each one of those subjects can take years of practice before people really know how to gain important insights. This class fills a crucial gap by bringing all the skills together in one place and giving you the time and space to practice those skills.

3 ASSIGNMENTS

Weekly assignments – 5%. In the beginning of the course, it is important to practice a number of core skills in Python and practice frequently. So, there are three short assignments that jointly total 5% of your final grade. These assignments have clearly correct/incorrect answers.

Unit assignments – 20% each. The course is divided into a number of units where we will explore one case study in great detail. This will involve constructing a theoretical model and performing statistics on the theoretical model. The unit assignments will ask you to explore further questions connected to these case studies: what happens if the model was a little bit different? What happens if the data was a bit different? They'll have a structured series of questions to guide you through the exploration. However, the focus on is typically on rigorous exploration, rather than arriving at a particular answer.

Final project – 35%. The final project will ask you to find some new paper and scrutinize it. You'll try to provide some alternative explanation for the papers result, implement it in a model and show that the model can adequately explain the original results. The final project will be turned in shortly after the date of our final. I technically have to hold a final, but I don't want to. So, we'll use the final exam as a day to chat about your project. 5% is allocated to showing up to the final and chatting with me about your project. The remaining 30% is the core of your grade on this project.

4 CALENDAR

Day	Content	Readings	Items Due
January 9	Intro to course		
	Unit 1: The Modeler's Toolkit		
January 11	Logic		
January 16	Python	Downey 1	
January 18	The lady tasting tea	Downey 2	Weekly 1 (due 19th Jan)
January 23	Hypothesis testing	Downey 3	
January 25	Robustness analysis		Weekly 2 (due 26th Jan)
January 30	-Work time-		
February 1	Bayesian inference	3blue1brown	Weekly 3 (due 2nd Feb)
February 6	The grid approximation algorithm		
February 8	Priors		
	Unit 2: Modeling experiments		
February 13	Conformity	Asch	
February 15	Models of mixtures		Unit 1 assignment
February 20	*Canceled*	*Reading week*	
February 22	*Canceled*	*Reading week*	
February 27	The Monte Carlo Markov Chain algorithm		

February 29	Sample sizes		
March 5	Generalizability	Henrich, Heine & Norenzayan	
March 7	-Work Day-		
	Unit 3: Causal reasoning		
March 12	Priming	Bargh	
March 14	Normal distributions		Unit 2 assignment
March 19	Measurement error		
March 21	Mediation		
March 26	Confounding	TBD	
March 28	-Work time-		
April 2	Causal graphs		
April 4	Research final project		Unit 3 assignment
April 9	Research final project		
April 11	Research final project		

5 READINGS

The expectation is that you will read the assigned readings before class. Readings for this class break down into two categories.

(1) Early in the term, you'll look at some introductions to technical materials (Downey, 3blue1brown). The motivation here is this – technical topics sometimes take multiple attempts to digest. So try to take one honest attempt at digesting it before we discuss it together.

(2) Later in the term, the readings will form the basis of case studies that we will explore in great detail over several weeks. I'll want to get your thoughts and criticisms, so it's helpful if you've thought about the paper ahead of our discussion. If you haven't read the paper, you are liable to get lost as the unit continues.

There is no textbook to purchase for this course. Every reading is available through Canvas

6 SOFTWARE

This class makes extensive use of programming to construct models and perform statistical analyses. We'll use the Python language. Python is often described as being easy to learn and there are loads of teaching materials freely available online. Python is an especially useful language to have some experience with because it's the most popular language for data science in the private sector.

For working with Python, we'll use Google Colab. It's a web-based programming environment. This means you do not have to have Python installed on your computer to write and run code. It is sent off to a cloud server to execute. It can be found here:

https://colab.research.google.com/notebooks/intro.ipynb\#scrollTo=-gE-Ez1qtyIA

It works just like Google Docs – you can share your document with anyone and save it on Google Drive. One draw back with Colab is that it requires an internet connection to run. If find that you often work without an internet connection, you should download the anacondas distribution.

https://www.anaconda.com/products/distribution

Think of this as the desktop version of Colab. It gives you access to an environment (Jupyter notebooks) which looks and works like Colab. It gives you a copy of Python plus all the packages we'll use throughout the term.

6.1 HOW HARD IS THE PROGRAMMING?

Most students have had at least some programming by the time they take 303. And some exposure to programming is all you need. In the beginning, we stick to fairly simple programming techniques: lists, loops and defining functions. I introduce all these concepts from the ground up in the January 16 lecture. I can promise no tree algorithms and no graph algorithms. If you have taken or are taking CPSC 110, 103 or 107, I think you should be good.

In the past, students who have never had any programming experience found this course difficult but not impossible. If you are in this category should anticipate putting in extra time outside of class and assignments to keep up. The Downey reading is taken from a bigger textbook on data science in Python written for total beginners. It is freely available online for extra practice.

6.2 IS PRIOR KNOWLEDGE OF STATISTICS REQUIRED?

No. Given the way the course requirements work out, most of the students in the class will have taken a statistic class once before. But the way I teach stats, this class could be a good first or second class in stats. We take a pretty different approach than standard stats textbooks. First, we focus almost exclusively on something called "Bayesian statistics" which is a rival approach to the common "frequentist" approach. This means we have to build things up from the foundations. It also means what you learn won't be redundant with your psych stats courses, you'll just know two techniques for solving similar problems.

7 POLICIES

7.1 ABSENCES

Lecture notes will be posted on Canvas that cover the material from class that day. If you need to miss class for any reason, just review these notes and you should be up to speed. They will typically be available the same day but occasionally it will take me an extra day to finish them. If I'm later than a full day, please email me because I forgot.

7.2 LATE WORK

If you know you'll need an extension, email me in advance of the deadline to request it. In your request, please propose a new deadline you think you can achieve. You do not need to describe your reasons for needing the deadline. These requests are always granted so long as you propose a reasonable deadline.

If an emergency comes up and you can't submit on time or request an extension, email me as soon as possible after the deadline to request a new deadline. The goal is to keep you on track, not to penalize you for unexpected events. These requests have to be decided on a case-by-case basis.

7.3 COLLABORATION

Professional scientists these days usually work in large teams. So why shouldn't you? Students can submit joint work on every assignment. You can have up to one collaborator. Just list your collaborators at the top of the assignment and turn in one copy for all of you. You'll all receive the same grade if you choose to collaborate. This is totally optional. You don't need a group if you feel good completing the assignment on your own.

7.4 GENERATIVE AI

You are permitted to use artificial intelligence tools, including generative AI. Google Colab now has experimental generative AI built into the interface. However, you are ultimately accountable for the work you submit. AI tools can be very useful for simple queries about Python and its packages. For example, you might ask "How do I filter a dataset in Pandas?" or "what is a function that computes the probability density under a beta distribution?"

There are some strong limitations on what generative AI can do for you. First, it won't be able to complete the assignments. I've tested it out. It fails to follow instructions, produces non-functional code, or ignores principles taught during lectures. Generative AI also has a limit on its "context" – the number of ideas, constraints, prompts and documents it can keep track of at the same time. This makes it a poor tool for more complex assignments that build on top of earlier ones. While you might have good luck

relying on during the first few assignments, I suspect this will hurt you later on when you coding skills are underdeveloped.

Second, generative AI is not particularly good at attributing ideas. Using it to write the final project creates a substantial risk of accidental plagiarism. I will treat accidental plagiarism the same as real plagiarism – you will receive a 0 for the assignment.