SOCI 508: Advanced Quantitative Methods
Winter 2023 Term 2

Instructor

Laura K. Nelson
Email: laura.k.nelson@ubc.ca
Twitter: @LauraK_Nelson
Office hours: TuesWed: 12:30pm - 1:30pm, on Zoom and in AntSoc 1314
Sign-up for a slot here: https://calendly.com/laura_nelson/30min

Course Description

Meetings: Tues: 2:00pm - 5:00pm in ANSO-2107

Readings:

We will be using one main textbook:

Charles R. Severance. Python for Everybody: Exploring Data Using Python 3. This book is free and open source (open science, it's a thing!). There are multiple online options, or you can buy the book on the cheap if you prefer a hard copy.

Other readings are available online through the library or, in select cases, on Canvas. If you are having problems accessing any of the readings let me know.

Overview:

Data are everywhere. Historical documents, literature and poems, diaries, political speeches, government documents, emails, text messages, social media, images, maps, cell phones, wearable sensors, parking meters, credit card transactions, Zoom, surveillance cameras. Combined with rapidly expanding computational power and increasingly sophisticated algorithms, we have now entered the computational era.

New sources of data and computational methods have impacted every single discipline, including those in the social sciences and humanities. Those of us studying human culture and society must now contend with large amounts of digital/digitized data and computational approaches to studying culture and society. Some of us may use new data and methods in our research. All of us must have a basic understanding of these approaches.

In this course we will explore how data, digital material and digital methods are impacting the humanities and social sciences. We'll also reflect critically on how these materials and methods are being used inside and outside of academia.

We'll explore these issues from three perspectives: the technical skills necessary to access and analyze data (Python! computers!), the epistemology underlying these methods and best practices re: research design (hint: there is not just one way to do computational research), and the practical knowledge we and others can produce using digital data and methods.

Learning Goals

This course is NOT a computer science course. It is not even a course on computer programming. No prior programming experience is required or assumed. It is primarily a social science and humanities course, with an eye toward digital technologies. We will not have computers analyze data or cultural material for us. Instead, we will harness the superior ability for computers to count and extract patterns from complex data and cultural material, and use this output to enhance our own critical thinking and interpretive analyses. To implement these methods we will use the open source (and free!) programming language Python and the Jupyter platform.

Specific skills covered include collecting digitized data, structuring digitized data, data formats, and an introduction to text analysis – with a smudge of machine learning. The ultimate goal is to encourage you to think about novel and creative ways you can apply these techniques to your own area of study.

By the end of the course you will have a better understanding of the range of types of digital data available, different ways of collecting and structuring them, ways computers can help you answer questions, what kind of evidence the different techniques produce, and how this evidence can be used to help you better understand the social world.

Learning Outcomes

By the end of the course you should be able to:

1. Explain three different ways computers are being used in social science and humanities research to ask and answer questions
2. Know enough Python basics to qualify as, at a minimum, a novice programmer
3. List three different types of digital data (e.g., delimited separated files, raw text, json), be able to write Python code to input and process each type, and explain how and why you might use each data type in research
4. Write Python code to collect and structure digitized data, including from APIs, process the data, and produce two or three visualizations and/or output to explore or analyze the data
5. Explain what the output from computational methods means, and derive a few insights about the social world from the output and visualizations

6. Feel comfortable learning new techniques and new Python libraries on your own

Course Format

Course time will consist of a mix of tutorials, in-class programming exercises, substantive discussion, and workshopping. For programming skills, the course will be split into three parts: basics of Python; data analysis, collection, and visualization; and text analysis. Substantively, we will start with thinking through research design and approach, including critical perspectives on research design.

The ultimate goal is for you to be able to use these different types of data and approaches to explore research applications of your choosing. As such, you will complete three short projects showcasing your ability to analyze data using Python, with a focus for the second two projects on text analysis.

The form of these three projects will be a Jupyter notebook, also called a computational essay. A Jupyter notebook is an interactive computational environment that allows you to combine text, code, output, and visualizations into one document, and easily share the document with colleagues or publish it on the web. It can be used with a variety of programming languages, including Python. Because it is a functioning program environment that also can incorporate text and visualizations in a seamless and visually pleasing manner, it is popularly used to teach programming and computational methods, to present scientific findings, and it is starting to be widely used in industry, including data-driven journalism. As such, your completed final project will be an excellent addition to your resume or CV. Here's an example of a Jupyter notebook, and a good one to emulate for your own projects:

http://nbviewer.jupyter.org/github/brianckeegan/Bechdel/blob/master/Bechdel_test.ipynb

Course Requirements

Technology Requirements

Students must have access to a laptop and you must bring it to class every day. If you do not have a laptop contact me and we can try to work something out.

This course will be taught in the open source programming language Python and the programming environment Jupyter. Participants should install Anaconda prior to the first day of class (try your best, be we'll trouble shoot that first day if you have difficulties):

- Anaconda for Python 3.9 (https://www.anaconda.com/download/). Anaconda includes Python, a Python interpreter, the necessary Python packages, and Jupyter. I recommend using the Graphical Installer.

Grading and Assignments

15% Attendance and participation
- participation in class discussions
- completing in-class tutorials

20% Weekly reading responses
20% Weekly programming exercises
45% Three computational essays (15% x 3)

Course Structure

To facilitate learning both programming and domain knowledge, the course will consist of practical tutorials aimed at getting you processing and analyzing cultural material via Python, as well as discussions about assigned readings that explore a practical question, epistemology, or issue using computational techniques. It is important that you complete the readings before each class and come prepared to discuss the material. During these discussions there will be space to critique the material and these methods. It is important that we respect one another's thoughts, give everybody the space to talk, and address our comments at the ideas and not the person.

The course will meet once a week for 3 hours. Class time will be a combination of lecture, discussion, hands-on tutorials, and programming practice.

Class time will be structured as follows:

2:00 - 2:30: programming tutorial
2:30 - 3:00: in-class exercises
3:00 - 3:15: exercise solutions
3:15 - 3:30: break
3:30 - 4:30: substantive discussion
4:30 - 5:00: practicum – tutoring/get started on your programming exercise

Assignments

Reading Responses must be submitted by 10AM on Tuesdays. Please read everyone's reading response before class on Tuesday. The programming exercises must be submitted by 10PM on Sunday.

In lieu of a final exam, you will explore questions relevant to the humanities and/or social sciences in three research projects using three different types of digitized data or material and computational techniques: text analysis, visualizations, and machine learning. These projects are designed to encourage you to creatively combine the knowledge and skills built through the semester to explore questions about the social world. I will hand out detailed rubrics closer to the due date for each project. We will set aside some class time to view and comment on each other's projects.

Attendance and Participation

Each week we will learn skills and develop knowledge that builds on previous skills learned, so it is important to attend every class. Learning Python is like learning a foreign language. The best way to learn it is to use it all the time. To encourage the continual use of the skills you are learning we will complete short exercises during class. While I will not keep attendance, if you miss a class you will miss in-class tutorials and discussions that might impact your grade.

However, given the current situation I don't want you to attend class if you're sick, and your health should be your top priority. I will drop the lowest grades that depend on attendance (i.e. in-class tutorials). If you need to miss a class please do so, and there's no need to inform me. If you think you will need an extended absence (more than three class periods) do let me know so we can discuss options.

Reading Responses

In this course humanities and social science questions are central, and computational techniques are used to answer those questions. There will be three types of assigned readings: (1) a Python textbook to help you get familiar with computer science basics; (2) articles exploring the epistemology underlying computational analyses and research methods; and (3) articles that apply computational methods to answer questions about the social world. The weekly reading responses are designed to help you engage deeply with the material, and will help structure our discussions.

Each week you must post a reading response on Canvas consisting of the following three elements:

1. Describe the three most important aspects (concepts, issues, factual information, etc.) of the readings, justifying your choices (you should draw connections across readings for this).
2. Identify two aspects of the readings you don't understand, and briefly discuss why these confusing aspects interfered with your general understanding of the readings.
3. Pose a question to the text's authors, the answer to which should go beyond the reading content and does not reflect the areas of confusion in point 2. You may bring in other readings or other topics we have discussed in class in this element, and/or draw connections between the readings for the week.

Post your reading response with the above three elements to the appropriate Canvas discussion thread by 10am on the day of class.

I will drop the three lowest grades for the reading responses. If you miss a class or have a week where you simply can't complete the readings no issue, and no need to inform me.

Programming Exercises

As with learning any foreign language, the best way to learn Python is to keep writing code. You are required to complete (short) weekly programming exercises, designed to give you practice writing code. These will also help me gauge the speed at which we're going through the programming material. You can get up to 3 points for each programming exercise. As with the reading responses, I will drop the three lowest grades for the programming exercises. The programming exercises are due by 10pm on Sunday.

Computational Essays

The goal of the computational essays is to creatively combine the techniques you learned in the course to explore questions related to the humanities or social sciences. Note that this is an introductory course, and I will present a lot of material throughout the semester, so these projects will by necessity be only preliminary explorations of substantive questions. The goal is to provide you with enough technical skills in the course so you can explore substantive questions further on your own, through your own projects or in other classes.

Through these essays you should show that you understand (a) what types of questions are interesting or important to humanists and/or social scientists, (b) what types of questions can be best answered using computational or digital techniques, (c) what types of techniques and evidence are appropriate to best answer your question, and (d) that you can think about how to present your findings and analysis in a reproducible way and in a way that supports, and persuades others of, your conclusion.

Keeping the above goals in mind, your computational essays should include the following:

1. 1-2 cells describing the question or puzzle you are exploring, why it is interesting or important, how others have attempted to answer this question, and how you are improving on these answers. If no one has addressed this question, explain why you think this is the case. In other words, what are you doing that's different from what others have done?

2. 2-4 cells describing the data or material you are using to explore the question and how you collected the data or material. These cells should include summary statistics of the data/material. If appropriate, describe what your data or material are representative of.

3. 2-10 cells containing the analysis or steps toward an analysis. These cells should contain a description of the planned analysis process and why it is appropriate for your question and data/material, followed by code implementing either some of the techniques or at least provides some summary descriptions of your data or material, the output from the calculations or the summary descriptions of your data or material, and a description of how you understand the output.

4. 1-2 cells producing some sort of data visualization or data summary output.

5. 1-2 cells detailing your interpretation of the output, and broader conclusions about history and/or the world around you that you draw from your exploration, or that you would hope to draw if you carried the project further. Support your interpretation with evidence from your analysis. End with suggestions for further analyses and other data or material that could help us continue to explore your question.

Questions? Discussion Board, Office Hours, and Email

If you come across errors as you run code that you can not solve, post them to the discussion board on Canvas (start a new thread for new errors). You may also post questions or comments about the readings or about your research projects. I encourage everyone to answer each other's questions, as this is the best way to learn complicated material. Often many people will get the same error or will have similar questions, so check the discussion board for answers before posting your error or question. This is not the comments section on YouTube, so keep your comments respectful. Disrespect will absolutely not be tolerated.

You are also encouraged to come to my virtual office hours. Email should only be used for quick logistical questions or if you need to inform me of a planned absence. I will get back to emails within 16 working-hours, so plan ahead. My general philosophy is to work hard during the week, and to take weekends off. If you email me or post questions on a Friday afternoon or a weekend, I may not respond until the following Monday.

Consulting Resources

I encourage you to take advantage of the consulting resources at the UBC Library. They offer a wealth of consulting services, including data services, digital scholarship, and writing. Use them for this course, but their services can also help you throughout your time at UBC.

Note on Plagiarism

I encourage you to work together to help each other review the readings and to learn the coding. However, *all written and coding work must be your own*. I take academic honesty seriously, and you should too.

This class has very strict standards for borrowing code: if you borrow anything for use in your projects, you must have a citation. A good guideline is that if you take more than three lines of code from some source, you must include the information on where it came from. A URL or a notation (e.g., "Pandas help files") is fine. If it is an entire function, note it at the beginning of the code segment and include any original credit information. Provide a qualitative description of what you used, and what you changed/contributed. If you are unsure about this policy, ask the instructor. The university's academic integrity policy discusses actions regarded as violations and consequences for students.

For more information on your rights and responsibilities as a student see:
https://academicintegrity.ubc.ca/

Code of Conduct

The course code of conduct ensures our diverse community is a safe space where we can learn from each other. I recognize that people may have good intentions, but sometimes our actions affect people differently than we anticipated. The point of having procedures to address disagreements is to foster broader understanding of people different from ourselves.

We will begin with UBC's Student code of conduct. During the first week of class we will expand this code of conduct together, to ensure everyone is able to successfully learn throughout the course.

Course Schedule

Week	Date	Theme ********** Notes	Readings
1	Jan. 10	installation check, python and markdown, and course overview ********** come to class with Anaconda 3.8 installed.	the syllabus
2	Jan. 17	philosophy of computational thinking **********	*Python for Everybody*, Chapters 1, 2, and 3 Rob Kitchin (2014). "Big Data, new epistemologies and paradigm shifts." *Big Data & Society* 1 (1). Sebastian Benthall (2016). "Philosophy of Computational Social Science." *Cosmos and History: The Journal of Natural and Social Philosophy* 12 (2): 13-30. Petter Törnberg & Justus Uitermark (2021). "For a heterodox computational social science." *Big Data & Society.* 8(2).
3	Jan. 24	merging the quantitative and the qualitative **********	*Python for Everybody*, Chapters 5, 6, and 8 Monica Lee and John Levi Martin (2015). "Coding, counting and cultural cartography." *American Journal of Cultural Sociology* 3: 1–33. Laura K. Nelson (2020). "Computational Grounded Theory: A Methodological Framework." *Sociological Methods and Research* 49 (1), 3–42. Juan Pablo Pardo-Guerra and Prithviraj Pahwa (2022). "The Extended Computational Case Method: A Framework for

			Research Design." *Sociological Methods & Research*, 51(4), 1826–1867.
4	Jan. 31	research strategies and data analysis **********	*Python for Everybody*, Chapter 7 Introduction to Pandas Desi Rodriguez-Lonebear (2016). "Building a Data Revolution in Indian Country." In T. Kukutai & J. Taylor (Eds.), *Indigenous Data Sovereignty*. Canberra: Australia National University Press. Tukufu Zuberi (2008). "Deracializing Social Statistics Problems in the Quantification of Race." Pp. 127-136 in *White Logic, White Methods: Racism and Methodology*, edited by Tukufu Zuberi and Eduardo Bonilla-Silva. Rowman & Littlefield Publishers. Quincy Thomas Stewart (2008). "Swimming Upstream: Theory and Methodology in Race Research." Pp. 111-125 in *White Logic, White Methods: Racism and Methodology*, edited by Tukufu Zuberi and Eduardo Bonilla-Silva. Rowman & Littlefield Publishers. Handout: Originating, Specifying, and Central Questions Handout: Where to find data
5	Feb. 7	data visualization **********	Kieran Healy and James Moody (2014). "Data Visualization in Sociology." *American Review of Sociology*. 40: 105-28. Catherine D'Ignazio and Lauren Klein (2020). "On Rational, Scientific, Objective Viewpoints from Mythical, Imaginary, Impossible Standpoints." *Data Feminism*.
6	Feb. 14	data collection and APIs **********	*Python for Everybody*, Chapters 4, 13 Moya Bailey (2015). "#transform(ing)DH Writing and Research: An Autoethnography of Digital Humanities and Feminist Ethics." *Digital Humanities Quarterly* 9(2). Catherine D'Ignazio and Lauren Klein (2020). "The Power Chapter." *Data Feminism*.

7	Feb. 21	midterm break **********	no class
8	Feb. 28	counting words ********** **First computational essay due before class**	*Python for Everybody*, Chapters 9, 10, and 11 Laura K. Nelson, Rebekah Getman, & Syed Arefinul Haque (2022). "And the Rest is History: Measuring the Scope and Recall of Wikipedia's Coverage of Three Women's Movement Subgroups." *Sociological Methods & Research*, 51(4), 1788–1825. Christopher A. Bail, Taylor W. Brown, and Marcus Mann (2017). "Channeling Hearts and Minds: Advocacy Organizations, Cognitive-Emotional Currents, and Public Conversation." *American Sociological Review* 82(6): 1188-1213.
9	March 7	part-of-speech & dependency parsers **********	Oscar Stuhler (2022). "Who Does What to Whom? Making Text Parsers Work for Sociological Inquiry." *Sociological Methods & Research*, 51(4), 1580–1633. Carly Knight (2022). "When Corporations Are People: Agent Talk and the Development of Organizational Actorhood, 1890–1934." *Sociological Methods & Research*, 51(4), 1634–1680.
10	March 14	supervised machine learning **********	Bart Bonikowski, Yuchen Luo, & Oscar Stuhler (2022). "Politics as Usual? Measuring Populism, Nationalism, and Authoritarianism in U.S. Presidential Campaigns (1952–2020) with Neural Language Models." *Sociological Methods & Research*, 51(4), 1721–1787.
11	March 21	topic models ********** **Second computational essay due before class**	Daniel Karell & Michael Freedman (2019). "Rhetorics of Radicalism." *American Sociological Review*, *84*(4), 726–753. Rochelle Terman (2017). "Islamophobia and Media Portrayals of Muslim Women: A Computational Text Analysis of US News Coverage." *International Studies Quarterly* 61(3): 489-502.
12	March 28	word embeddings **********	Andrea Voyer, Zachary D. Kline, Madison Danton, & Tatiana Volkova (2022). "From Strange to Normal: Computational Approaches to Examining Immigrant Incorporation Through

			Shifts in the Mainstream." *Sociological Methods & Research*, *51*(4), 1540–1579. Alina Arseniev-Koehler & Jacob G. Foster (2022). "Machine Learning as a Model for Cultural Learning: Teaching an Algorithm What it Means to be Fat." *Sociological Methods & Research*, 51(4), 1484–1539. *Optional:* Austin C. Kozlowski, Matt Taddy, and James A. Evans (2019). "The Geometry of Culture: Analyzing Meaning through Word Embeddings." *American Sociological Review* 84(5).
13	April 4	looking to the future: machine vision and generative models **********	Jackelyn Hwang and Robert J. Sampson (2014). "Divergent Pathways of Gentrification: Racial Inequality and the Social Order of Renewal in Chicago Neighborhoods." *American Sociological Review* 79(4), 726–751. James Evans (2022). "From Text Signals to Simulations: A Review and Complement to Text as Data by Grimmer, Roberts & Stewart (PUP 2022)." *Sociological Methods & Research*, *51*(4), 1868–1885.
14	April 11	presentations + wrap-up **********	Catherine D'Ignazio and Lauren Klein (2020). Chapter 2, Collect, Analyze, Imagine, Teach. *Data Feminism*.
15	April 19	Final (no class) ********** **Third computational essay due by 5:30pm**	