Gov355M: AI & Programming Foundations for Computational Social Science

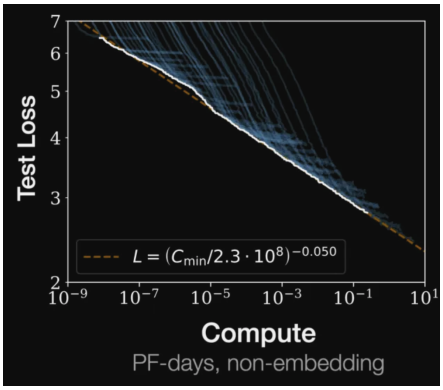
Spring 2026 ∗ Tuesdays & Thursdays, 3:30–5:00 pm ∗ Location: GAR 3.116

Connor T. Jerzak
Assistant Professor of Government
The University of Texas at Austin
Course: GOV 355M (Unique #38794)
Syllabus URL: https://ConnorJerzak.com/AIPCSS-syllabus
Feedback: https://admonymous.co/cjerzak
Piazza Signup: https://piazza.com/utexas/spring2026/gov355m
Gradescope Signup: https://www.gradescope.com/courses/1228869
Gradescope Entry Code: YBBPZ5

In an era where data and AI increasingly drive decision-making in policy, government, and the social sciences, this course introduces undergraduate students to the computational foundations of modern empirical research. The class combines three pillars:

- **AI foundations**: how large language models (LLMs) and other AI tools work at a high level, how to use them responsibly, and how to integrate them into research workflows without outsourcing critical thinking.
- **Programming foundations**: practical skills in R, Python, and SQL for data wrangling, visualization, and reproducible research using notebooks and version control.
- **Data & analysis pipelines for computational social science**: how to find, collect, and clean information from open data portals, APIs, and the web; how to merge and manage large datasets; and how to analyze them using descriptive statistics, networks, text-as-data, and basic machine learning.

Throughout, we emphasize trustworthy programming and analysis in the AI age: documenting assumptions, communicating uncertainty, and recognizing the limitations and potential of AI-based tools. A recurring theme of the course is that while today you can accomplish a great deal with one-shot prompting in general-purpose AI tools, deeper technical understanding of programming, data, and workflows unlocks qualitatively new kinds of questions you can ask and answer at scale. Course assignments: students will build end-to-end pipelines that use AI and code to obtain, document, and analyze real data about politics, policy, economic or social behavior. The course is aimed at juniors and seniors in Government (and related majors) who have some prior quantitative or coding experience and want to deepen it for research, policy, work, or graduate study.



1 Contact Information

Instructor

NAME: Connor T. Jerzak
PREFERRED FORM OF ADDRESS: Please call me "Connor" (no titles needed!).
EMAIL: connor.jerzak@austin.utexas.edu
URL: https://ConnorJerzak.com
OFFICE: BAT 3.108
OFFICE HOURS: Tuesdays, 2:00–3:00 pm (no appointment needed), and by appointment
—*Note: OH often held on 2nd floor porch in BAT when weather permits.*
—Zoom link (for remote OH as needed): https://ConnorJerzak.com/zoom
—By default, all office hours will be communal.
—Indeed, there are often synergies across questions, and from student to student.
—If needed, one-on-one meetings can be arranged.

Planned guest speakers include:

- Adel Daoud (Linköping University, Sweden),
- Isaac Corley (UT San Antonio / TorchGeo),
- Gauthier Fally (UT Austin),
- Huimin Li (Princeton),
- SayedMorteza Malaekeh (UT Austin),
- Fucheng Warren Zhu (Harvard),
- and others (subject to availability).

2 Prerequisites

The most important prerequisite is curiosity and a willingness to be temporarily uncomfortable while learning new tools.

Formally, the course assumes:

- at least one prior course in statistics, research methods, or data analysis; and
- some exposure to coding (in R, Python, or another language), even if you still feel like a beginner.

If you are unsure whether your background is sufficient, please contact the instructor. The course is designed for juniors and seniors in Government and related disciplines who want to deepen their computational skills for research, industrial, or policy work.

3 Instructional Tools

We will use the following two tools to run the course:

- **Piazza**: We will use an online discussion platform for discussions about coding, readings, and projects. You should feel free, at any point during the semester, to share resources or links you think your peers might find useful.
- **Gradescope**: Gradescope is the platform you will use to submit and receive feedback on exercises and the final exam. You can access Gradescope via Piazza.

4 Learning Outcomes

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

- explain at a high level how modern AI and foundation models work, and articulate their limitations for social-science applications;
- move from one-shot prompting to reusable, documented workflows that combine AI tools with R, Python, and SQL;
- design and implement simple *agentic* AI workflows — for example, multi-step pipelines where an AI system retrieves information, calls tools or APIs, and checks its own work — to support computational social science;
- build, version, and share an end-to-end computational social science pipeline that answers a concrete question in government, policy, or social behavior.

5 Course Framework

The course has several main components: in-class discussions, live-coding demos, short exercises done in pairs, and a collaborative final exam.

Class meetings. Each class will typically have three parts:

1. a discussion on the day's conceptual theme where students are expected to be willing to engage in debate, weight the pros and cons of disparate ideas or tools, and discuss ways of connecting these concepts to questions they personally care about in politics, policy, or social life.
2. a live-coding session where we build something together in R or Python (often with an AI assistant in the loop); and

3. a brief closing activity or student-led discussion of an applied paper, data source, company, or government initiative related to course themes.

Wherever possible, in-person coding exercises will be framed as explorations or building blocks for your future progression, so that the skills you practice in class accumulate.

We ask that students observe the following guidelines:

1. *Be on time.* Late entries can be disruptive. If you have a recurring scheduling issue, talk to the instructor so we can find a plan that minimizes disruption.

2. *Keep computer use focused.* Laptops are essential for coding, but non-class use (email, social media) is distracting for you and those around you. During discussions, you may be asked to close laptops.

Exercises. Exercises will be done in rotating pairs, and will usually involve:

- a short technical task (e.g., scraping a small website, calling an API, cleaning a dataset); and
- a brief interpretive question tying the code back to a social science question.

Pairs will be randomly assigned and re-shuffled, to help you get to know multiple classmates and practice collaborative coding.

AI Policy: The "Co-Pilot" Rule. Since this is a course about AI and computational methods, we will not ban the tools we are studying. You are encouraged to use LLMs (ChatGPT, Claude, Copilot, etc.) to debug code, explain error messages, brainstorm research questions, and draft boilerplate syntax.

Indeed, today, you can already accomplish a surprising amount with a single well-crafted prompt. A key goal of this course is to help you see *how much more* becomes possible when you combine prompting with code, data structures, and basic software engineering. Throughout the semester we will highlight cases where a scripted or agentic workflow (e.g., looping over thousands of documents, joining to other data, logging and versioning results) gives you scale, transparency, and reliability that one-off prompts cannot.

However, our policy follows a standard of personal ownership over all outputs:

- **You are responsible for the output.** "The AI hallucinated" or "ChatGPT wrote it that way" is never a valid excuse. If your code breaks, or if your paper cites a study that doesn't exist, the error is yours alone. Treat the AI like a confident but occasionally incompetent intern: you must verify everything it produces before putting your name on it.

- **The "Explain It" Test.** You may use AI to help write code, but you must understand every line you submit. If asked in class or office hours to explain a specific function or loop in your assignment, and you cannot explain how it works or why it is there, you may not receive credit for that work. Most important, you should think at all times how to check the assumptions of the output you are producing—what sanity checks can I run to ensure I am on the right path.

- **Think first, prompt second.** AI is excellent at executing tasks but terrible at original insight. Do not outsource your critical thinking, but use AI tools as an accelerant to your curiosity.

6 Course Requirements

Grading philosophy (0-indexed). Grading in this course is *0-indexed*: everyone effectively starts at zero, and points are earned through completed, thoughtful work. If nothing is turned in or no activities are completed, the resulting score is simply 0—there is no automatic "100%" from which points are subtracted. Instead, you unlock points by submitting exercises, contributing in class and online, and developing your final exam. This mirrors many environments outside academia, where you start with zero output or visibility (for example, zero followers on TikTok or zero stars on GitHub) and build up over time through consistent, substantive contributions. The goal is to frame grades as accumulated evidence of learning and effort, not as penalties for mistakes.

Grading components. The course grade is based on the following components:

- **Class participation (20%)**:

  Participation includes attending class, engaging in discussions, contributing to Piazza, and being a constructive partner in group work. Being generous with questions (and patient with yourself and others when things are confusing) is part of this. Participation also will include brief in-class quizzes.

- **Class presentations (30%)**:

  Over the semester, each student (or small group) will lead a short in-class presentation. Options include:

  - presenting a social science paper or working paper whose main contribution is a new dataset or data pipeline (especially AI- or web-enabled); or
  - presenting a public data portal, API, or open-source tool and demonstrating how it could be used in computational social science.

  Presentations will typically be 10–12 minutes, followed by 5–8 minutes of discussion. Emphasis should be on:

  - the substantive question and why it matters;
  - how the data were obtained or constructed;
  - strengths and weaknesses of the design/data; and
  - what we can learn for our own projects.

- **Exercises (20%)**:

  Short exercises (roughly bi-weekly) will reinforce key concepts and provide guided practice. They will often be started in class and completed by the end of the week.

  - *Collaboration policy.* You are encouraged to discuss exercises with your assigned partner, other classmates, and instructional staff. However, each pair must write up and submit its own code and answers. Copying solutions defeats the purpose; if you are stuck, it's much better to reach out early.
  - *Submission.* Each exercise will come with a template (RMarkdown, Google Docs, or Jupyter, depending on context). Submit a compiled PDF (and, when requested, your source notebook/package) via Gradescope by 5 pm Friday. Late submissions are only accepted with prior approval from the instructor. Late submissions without prior approval will, by the 0-indexing grading philosophy, receive no credit.

Each exercise will also include a brief question nudging towards your future progression (research, work, startup, policy, etc., etc.).

- **Final exam (30%)**:

  The final exam is the capstone of the course. Details TBD, but it may include a real-life scenario involving:

  - building a new dataset (or substantially extending an existing one) relevant to a political or social-science question;
  - constructing an end-to-end, documented pipeline that uses code (and optionally AI tools) to obtain, clean, and analyze data; or
  - developing a small open-source tool (e.g., a wrapper around an API, a data-cleaning function, or a reproducible workflow template) useful for computational social science.

7 Course Materials

There is no single required textbook. Instead, we will draw from a mix of books, articles, documentation, and videos. A non-exhaustive list of helpful references:

- Borgman, C. L. 2015. *Big Data, Little Data, No Data: Scholarship in the Networked World.* MIT Press.
  https://mitpress.mit.edu/9780262529914/big-data-little-data-no-data/
- Salganik, M. J. 2017. *Bit by Bit: Social Research in the Digital Age.* Princeton University Press. Online resources and full text: https://www.bitbybitbook.com/
- Wickham, H., M. Cetinkaya-Rundel, and C. Grolemund, 2023. *R for Data Science.* https://r4ds.hadley.nz/
- James, G., D. Witten, T. Hastie, and R. Tibshirani. 2021. *An Introduction to Statistical Learning with Applications in R.*
  Free PDF: https://www.statlearning.com/
- Barocas, S., M. Hardt, and A. Narayanan. *Fairness and Machine Learning.* Book available at https://fairmlbook.org/

All required readings and links will be posted on Piazza. Whenever we assign a reading, we will include a URL in the syllabus and on Piazza.

8 Course Plan

This course is organized into four main modules:

1. **Module 1: AI & Programming Foundations**
2. **Module 2: Data Gathering for Computational Science**
3. **Module 3: Data Analysis for Computational Social Science**
4. **Module 4: Future Directions**

We move from AI and programming foundations (Module 1) to acquiring and organizing data (Module 2), and then to analyzing and interpreting data (Module 3). Module 4 focuses the future of programming and computational social science in the AI age, with special attention to agentic AI workflows and emerging tools that go beyond one-off prompting.

Note: Spring Break (no class) is the week of March 16–20, 2026. There are no meetings on Tues. March 17 or Thurs. March 19.

Module 1. AI Foundations & Programming Basics

C1 (Tu. Jan. 13) Overview of the course; what is computational social science? AI in government and policy.

Reading
- Lazer, D., A. Pentland, L. Adamic, et al. 2009. "Computational Social Science." *Science* 323(5915): 721–723. https://www.science.org/doi/10.1126/science.1167742
- Salganik, M. J. 2017. *Bit by Bit: Social Research in the Digital Age*. Read Chapter 1. https://www.bitbybitbook.com/

C2 (Th. Jan. 15) Programming refresher: data structures, data frames, and workflows in R and Python; working in RStudio and Jupyter/VS Code.

Reading
- Wickham, H. et al. *R for Data Science*, 2e. Chapters 1–3. https://r4ds.hadley.nz/
- Pandas documentation. "Getting started." https://pandas.pydata.org/docs/getting_started/index.html

C3 (Tu. Jan. 20) Reproducible research: RMarkdown/Quarto, Jupyter notebooks, and version control with Git and GitHub.

Reading
- Peng, R. D., S. Kross, and B. Anderson. *Mastering Software Development in R*. Read Sections 1.8–1.10 (on projects and packages). https://bookdown.org/rdpeng/RProgDA/
- Blischak, J. D., E. R. Davenport, and G. Wilson. 2016. "A Quick Introduction to Version Control with Git and GitHub." *PLoS Computational Biology* 12(1): e1004668. https://doi.org/10.1371/journal.pcbi.1004668
- (Optional) Software Carpentry. "Version Control with Git." https://swcarpentry.github.io/git-novice/

C4 (Th. Jan. 22) AI and foundation models: what are LLMs, what can they (not) do, and how might they assist social scientists?

Reading
- Bommasani, R., et al. 2021. "On the Opportunities and Risks of Foundation Models." https://arxiv.org/abs/2108.07258
- Matarazzo, A. and R. Torlone. 2025. "A Survey on Large Language Models with Some Insights on their Capabilities and Limitations." https://arxiv.org/abs/2501.04040
- (Video) MIT 6.S191: "Introduction to Deep Learning" Lecture 1 (YouTube). https://www.youtube.com/watch?v=njKP3FqW3Sk

C5 (Tu. Jan. 27) AI as a coding and research assistant: prompt design, chain-of-thought, and AI "agents" for search and data collection. Contrasting one-shot prompting with multi-step, code-based workflows and simple agentic scaffolding. How to verify and debug AI outputs.

Reading
- Schmidgall, S., et al. 2025. "Agent Laboratory: Using LLM Agents as Research Assistants." arXiv preprint. https://arxiv.org/abs/2501.04227
- (Skim) OpenAI usage and safety policies (for reference on responsible use). https://platform.openai.com/docs/usage-policies

C6 (Th. Jan. 29) Ethics, bias, and fairness in AI and data-driven governance.

Reading
- Barocas, S., M. Hardt, and A. Narayanan. *Fairness and Machine Learning.* Read Chapter 1. https://fairmlbook.org/

Module 2. Coding for Data Gathering

C7 (Tu. Feb. 3) The web as data: HTML, CSS, and the Document Object Model. Reading pages, inspecting elements, and thinking about what can (and cannot) be scraped.

Reading
- Pittard, Steve. 2022. Web Scraping with R. (First Few Chapters). https://steviep42.github.io/webscraping/book/
- W3Schools. "HTML Tutorial" (skim tags, attributes, and basic structure). https://www.w3schools.com/html/

C8 (Th. Feb. 5) Web scraping I: static pages and pagination with `rvest` (R) and `requests`/`BeautifulSoup` (Python). Legal/ethical considerations and `robots.txt`.

Reading
- Library Carpentry. "Introduction to Web Scraping." https://carpentries-incubator.github.io/lc-webscraping/
- rvest. "Web Scraping in R: rvest Tutorial." https://rvest.tidyverse.org/articles/rvest.html
- Real Python. "A Practical Introduction to Web Scraping in Python." https://realpython.com/python-web-scraping-practical-introduction/

C9 (Tu. Feb. 10) Web scraping II: dynamic content, simple browser automation (e.g., Selenium), and strategies for rate limits, failures, and logging.

Reading
- Munzert et al. *Automated Data Collection with R.* Read Chapter 3 (dynamic web pages). https://www.wiley.com/en-us/Automated+Data+Collection+with+R-p-9781118834817
- (Optional) Selenium documentation (Python bindings). https://www.selenium.dev/documentation/webdriver/

C10 (Th. Feb. 12) Working with APIs: REST, JSON, and authentication. Examples from the U.S. Census, OpenStreetMap/Overpass, and the City of Austin open data portal.

Reading
- Fergusson, A. and C. J. Wild. 2021. "On Traversing the Data Landscape: Introducing APIs to Data-Science Students." *Teaching Statistics* 43(S1): S71–S83. https://onlinelibrary.wiley.com/doi/10.1111/test.12266

- U.S. Census Bureau. "API User Guide." https://www.census.gov/data/developers/guidance/api-user-guide.html
- City of Austin Open Data Portal (browse datasets). https://data.austintexas.gov/

C11 (Tu. Feb. 17) Spatial data and remote sensing: shapefiles, rasters, and basic mapping. Using open geospatial data and (briefly) platforms like Google Earth Engine.

READING
- Data Carpentry. "Introduction to Geospatial Raster and Vector Data with R." https://datacarpentry.org/r-raster-vector-geospatial/
- Gorelick, N., M. Hancher, M. Dixon, et al. 2017. "Google Earth Engine: Planetary-Scale Geospatial Analysis for Everyone." *Remote Sensing of Environment* 202: 18–27. https://doi.org/10.1016/j.rse.2017.06.031

C12 (Th. Feb. 19) Crowdsourcing, online experiments, and human-coded data: platforms (Qualtrics, Prolific, MTurk), data quality, and ethics.

READING
- Grootswagers, T. 2020. "A Primer on Running Human Behavioural Experiments Online." *Behavior Research Methods* 52: 2283–2286. https://doi.org/10.3758/s13428-020-01395-3
- Chandler, J., C. Rosenzweig, A. J. Moss, J. Robinson, and L. Litman. 2019. "Online Panels in Social Science Research: Expanding Sampling Methods beyond Mechanical Turk." *Behavior Research Methods* 51(5): 2022–2038. https://doi.org/10.3758/s13428-019-01273-7

C13 (Tu. Feb. 24) Merging data and entity resolution: keys, joins, string matching, and probabilistic linkage.

READING
- Enamorado, T., B. Fifield, and K. Imai. 2019. "Using a Probabilistic Model to Assist Merging of Large-Scale Administrative Records." *American Political Science Review* 113(2): 353–371.
- (Optional) Comber, A. and W. Zeng. 2019. "Spatial Interpolation Using Areal Features: A Review of Methods and Opportunities Using New Forms of Data with Coded Illustrations." *Geography Compass* 13(10): e12465. https://doi.org/10.1111/gec3.12465

C14 (Th. Feb. 26) Databases and SQL: relational data models, joins, and queries. Connecting R and Python to SQLite/PostgreSQL.

READING
- Codecademy. "Learn SQL" (Modules on Manipulation and Queries). https://www.codecademy.com/learn/learn-sql
- W3Schools. "SQL Tutorial" (skim basic SELECT, WHERE, JOIN). https://www.w3schools.com/sql/

C15 (Tu. Mar. 3) Data cleaning and wrangling pipelines: tidy data, `dplyr`, and pandas. Building reusable workflows for social science datasets.

READING
- Wickham, H. et al. *R for Data Science*, 2e. Chapters on data transformation and tidying. https://r4ds.hadley.nz/

- Datacamp. "Python pandas Tutorial: The Ultimate Guide for Beginners." https://www.datacamp.com/tutorial/pandas

Module 3. Analysis for Computational Social Science

C16 (Th. Mar. 5) Exploratory data analysis and visualization: summary statistics, plots, and communicating uncertainty.

Reading
- Wickham, H. et al. *R for Data Science*, 2e. Chapters on data visualization (e.g., ggplot2 basics). https://r4ds.hadley.nz/
- (Video) StatQuest with Josh Starmer. "Linear Regression and Linear Models" playlist (focus on first 2–3 videos). https://www.youtube.com/playlist?list=PLblh5JKOoLUIzaEkCLIUxQFjPIlapw8nU

C17 (Tu. Mar. 10) Regression for social science: linear models, interpretation, and connecting to causal questions.

Reading
- James, G. et al. *An Introduction to Statistical Learning*, 2e. Read chapter on linear regression. https://www.statlearning.com/
- (Video) StatQuest. "Multiple Regression, Clearly Explained!!!" (YouTube). https://www.youtube.com/watch?v=EkAQAi3a4js

C18 (Th. Mar. 12) Classification and prediction: logistic regression, evaluation metrics (accuracy, ROC, AUC), and fairness metrics.

Reading
- James, G. et al. *An Introduction to Statistical Learning*, 2e. Read Chapter 4 (Classification). https://www.statlearning.com/
- (Video) StatQuest. "Logistic Regression" playlist. https://www.youtube.com/playlist?list=PLblh5JKOoLUKxzEP5HA2d-Li7IJkHfXSe

C19 (Tu. Mar. 24) Machine learning workflows: decision trees and random forests; train/validation/test splits; cross-validation.

Reading
- James, G. et al. *An Introduction to Statistical Learning*, 2e. Read Chapter 8 (Tree-Based Methods). https://www.statlearning.com/
- (Video) StatQuest. "Random Forests Part 1: Building, Using and Evaluating." https://www.youtube.com/watch?v=J4Wdy0Wc_xQ

C20 (Th. Mar. 26) Graphs and networks: basic network concepts (nodes, edges, degree, centrality) and examples from politics and policy.

Reading
- Barabási, A.-L. *Network Science*. Read Chapter 1 (What is a network?). https://networksciencebook.com/
- (Optional) Barabási, A.-L. *Network Science*. Read portions of Chapters 2–3 if interested. https://networksciencebook.com/chapter/2

C21 (Tu. Mar. 31) Text as data I: bag-of-words, document-term matrices, dictionary methods, and basic sentiment.

Reading
- Grimmer, J. and B. M. Stewart. 2013. "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts." *Political Analysis* 21(3): 267–297.
https://doi.org/10.1093/pan/mps028
- (Optional) Gentzkow, M., B. T. Kelly, and M. Taddy. 2019. "Text as Data."
https://web.stanford.edu/~gentzkow/research/text-as-data.pdf

C22 (Th. Apr. 2) Text as data II: LLMs as classifiers. Prompting LLMs to label text and designing validation checks.

Reading
- (Skim) Grimmer, J., M. E. Roberts, and B. M. Stewart. 2022. *Text as Data*. (Look at Chapter 1.)
https://press.princeton.edu/books/hardcover/9780691207544/text-as-data
- Gilardi, F., M. Alizadeh, and M. Kubli. 2023. "ChatGPT Outperforms Crowd Workers for Text-Annotation Tasks." *Proceedings of the National Academy of Sciences* 120(30): e2305016120.
https://doi.org/10.1073/pnas.2305016120

C23 (Tu. Apr. 7) Synthetic data and "digital twins": LLM-based survey respondents and simulations. Opportunities and pitfalls.

Reading
- Cho, S., J. Kim, and J. H. Kim. 2024. "LLM-Based Doppelgänger Models: Leveraging Synthetic Data for Human-like Responses in Survey Simulations." *IEEE Access*.
https://ieeexplore.ieee.org/document/10758652
- Bisbee, J. et al. 2024. "Synthetic Replacements for Human Survey Data? The Perils of Large Language Models." *Political Analysis*.
https://doi.org/10.1017/pan.2024.5

Module 4. Capstone & Future Directions

C24 (Th. Apr. 9) Measurement error, proxies, and uncertainty in computational social science data (including geocoded, scraped, and AI-labeled data).

Reading
- le Cessie, S., J. Debeij, F. R. Rosendaal, S. C. Cannegieter, and J. P. Vandenbroucke. 2012. "Quantification of Bias in Direct Effects Estimates Due to Different Types of Measurement Error in the Mediator." *Epidemiology* 23(4): 551–560.
https://doi.org/10.1097/EDE.0b013e318254f5de
- Cvetojevic, S., L. Juhasz, and H. Hochmair. 2016. "Positional Accuracy of Twitter and Instagram Images in Urban Environments." *GI_Forum* 4(1): 191–203.
https://doi.org/10.1553/giscience2016_01_s191

C25 (Tu. Apr. 14) Reproducible pipelines at scale: packaging code, containerization basics, and data sharing (GitHub, Dataverse, HuggingFace).

Reading
- Peng, R. D. et al. *Mastering Software Development in R*. Read Chapter 3 (R packages).
https://bookdown.org/rdpeng/RProgDA/building-r-packages.html

- (Video) Ben Hamner. "Reproducible Data Science with Docker Containers." (YouTube). https://www.youtube.com/watch?v=cYavmPa3aFQ
- Harvard Dataverse. "Getting Started." https://support.dataverse.harvard.edu/getting-started

C26 (Th. Apr. 16) From prompts to agents: agentic AI workflows and future applications. How can we programmatically architect AI environments (tools, memory, multi-step planning) to support computational social science? Short demos and idea clinic.

Reading
- Google Research. 2025. "Towards an AI co-scientist." https://arxiv.org/abs/2502.18864
- Anthropic. 2025. "Disrupting the first reported AI-orchestrated cyber espionage campaign." Anthropic blog. https://www.anthropic.com/news/disrupting-AI-espionage

C27 (Tu. Apr. 21) Guest speaker session (e.g., remote sensing, agentic AI workflows, or large-scale text pipelines). (Reading TBD based on speaker.)

Reading
- Short readings or videos assigned closer to the date, tailored to the speaker's expertise (posted on Piazza with URLs).

C28 (Th. Apr. 23) Student lightning talks with peer feedback, in preparation for final exam. Focus on communication of technical results.

Reading
- Winston, P. 2018. "How to Speak." (MIT lecture on presenting.) https://www.youtube.com/watch?v=Unzc731iCUY

Reading
- Chen, M., S. Mao, Y. Zhang, and V. Leung. 2014. *Big Data: Related Technologies, Challenges and Future Prospects.* (Skim concluding chapter.) https://link.springer.com/book/10.1007/978-3-319-06245-7

Final Exam Deliverables: **Final written exam and code deposit due by May 7, 2026 (tentative).**

9 Q&A

- *Q: Why are AI foundations and programming placed at the very beginning of the course?*

  A: Your final projects will require both computational skills and conceptual understanding of what AI tools can and cannot do. The sooner you acquire these skills, the sooner you can apply them to build your project pipelines. We therefore front-load core programming and AI foundations.

- *Q: Will there be guest speakers?*

  A: Yes, schedule and availability permitting. Planned guests include practitioners and researchers working on remote sensing (e.g., TorchGeo), AI agents for scientific workflows, and applications of LLMs in social science.

10 A Note on the Readings

When doing the readings:

1. Consider skimming first to get the big picture: glance at figures, section headings, and conclusions to decide where to focus.
2. Read introductions and conclusions carefully; then dive into technical details as needed for your project or interests.

Do not be discouraged if some of the math or technicalities feel heavy. The goal is to extract core ideas and connect them to code and applications. Office hours, peers, online resources, and implementing the methods yourself are all part of the learning process.

11 Other Resources

Additional resources that may be helpful:

On giving presentations:

- Winston, P. "How to Speak." (MIT lecture.)
  https://www.youtube.com/watch?v=Unzc731iCUY

On writing:

- Pinker, S. *The Sense of Style: The Thinking Person's Guide to Writing in the 21st Century*. Penguin Books.
- Williams, J. and J. Bizup. *Style: Lessons in Clarity and Grace*. Pearson.

12 Arrangements

Students with accommodations approved by UT Disability & Access Services should share their accommodation letters with the instructor as early as possible so that we can set up a supportive environment and appropriate arrangements.

13 Integrity

"The core values of The University of Texas at Austin are learning, discovery, freedom, leadership, individual opportunity, and responsibility. Each member of the University is expected to uphold these values through integrity, honesty, trust, fairness, and respect toward peers and community."

— UT Austin Code of Conduct

"As a student of The University of Texas at Austin, I shall abide by the core values of the University and uphold academic integrity."

— UT Austin Student Honor Code

14 Final Reflections

The instructor reserves the right to modify aspects of the course (including topics, readings, and deadlines) in response to student feedback, pacing, or new developments in AI and computational social science. Any changes will be communicated via Piazza and in class.