Computational Methods for Area Studies

Version dated 26 April 2026

Instructor: thomas.hegghammer@all-souls.ox.ac.uk

Date: Mondays 3.30–5.30pm, Trinity term 2026

Place: Middle East Centre Boardroom

Summary

This course introduces graduate students in the Humanities and Social Sciences to computational methods for research. It focuses on how recent advances in artificial intelligence can support the collection, extraction, organisation, and analysis of historical and cultural materials, especially text, images, and audio. Participants will learn a range of practical techniques, including data visualisation, map-making, optical character recognition, audio transcription, text mining, and image analysis. They will also learn how to harness large language models for academic research through agentic coding frameworks, programmatic inference, and model fine-tuning. We will work mainly in R and Python, and the course includes a from-scratch introduction to both. However, it moves quickly to applied use, so students new to programming should expect substantial self-study. The course is cross-disciplinary in scope, mixing approaches from digital humanities and computational social science and focusing more on data generation than statistical modelling. We will use real historical materials from the Middle East and other regions, with particular attention to the challenges of working computationally with non-Western languages such as Arabic. The course is especially suited to students in Middle East and other Area Studies, but it should also be useful to students in History, Anthropology, Sociology, and Political Science.

Learning aims

- Working proficiency in R and Python.
- Command of agentic coding tools for LLM use.
- Ability to apply a range of computational techniques to historical documents and data, including data visualisation, map-making, optical character recognition, audio transcription, as well as analysis of text, images, audio, and video.
- Understanding of how to fine-tune computer vision models and large language models for domain-specific tasks.
- Awareness of the main systems for storage and retrieval of large-scale research data.
- Facility with remote computing, including SSH, APIs, and cloud-based virtual machines.
- Familiarity with plain-text workflows and typesetting with markup languages.
- Grasp of best practices for version control and reproducibility of programmatic work.
- Knowledge of current scholarly debates around the ethics and utility of artificial intelligence in humanities and social science research.

Topic overview

- Week 1: Working in plain text.
- Week 2: Thinking computationally.
- Week 3: Agentic AI.
- Week 4: Visualisation and map-making.
- Week 5: OCR and audio transcription.
- Week 6: Data management and retrieval.
- Week 7: Text and image analysis.
- Week 8: Advanced model use and AI ethics.

Prerequisites

- None. But students who are new to coding must be prepared to do extensive self-study on the side.

Requirements

- Do the weekly readings listed below.
- Complete the weekly assignments by noon on the Sunday before class. The link to each week's assignment will be shared via Canvas Announcements on the Tuesday prior. See the "Communication" section below for how to submit.

Assessment

- M.Phil. students will be examined by a 3-hour examination at the end of Trinity term of their second year.
- M.Sc. students will be examined by a 5,000-word essay to be submitted by the standard deadline for M.Sc. assessment papers. The questions will be made available in 6th week.

Policies

Attendance and deadlines

- Attendance, preparation, and timely submission are considered minimum expectations. Exemptions and extensions will only be given in exceptional circumstances.
- If you are unable to solve an assignment problem, you should still submit a written best attempt along with a brief explanation of what you struggled with.
- Some of the readings in the "Tinker" section are programming vignettes. It's fine if you don't succeed in implementing them locally on your computer, so long as you make an attempt.

Communication

- Submit assignments via Git commits to an assignment repository on GitLab. Create a private repository for assignments and add me (`@Hegghammer`) as a collaborator; that way I can see and give feedback on the submission.
- Please use the Canvas Messaging feature for course-related correspondence.
- For questions or messages of a more general nature, the answer to which may be useful for others – feel free to use the Discussions section on the Canvas course page.

Class policies

- You may consume snacks or drinks in class, but please exercise caution so as not to spill on the computers on the table.
- If you are a parent and your childcare falls through, you can bring your child to class provided they can be present without disrupting class.
- If you need disability-related accommodations in this class, have emergency medical information that you wish to share, or need special arrangements to participate, please inform me as soon as possible via email or after the first class. For formal disability-related accommodations, you must also obtain a Student Support Plan, which will be sent by the Disability Advisory Service directly to me.

LLM use

- Feel free to use LLMs in any way you like as a learning tool. The University's guidelines for AI use provides useful tips and pointers.
- Never present LLM-generated content as your own. Doing so is a breach of the University's rules on plagiarism, the definition of which includes "the use of material generated wholly or in part through use of artificial intelligence".
- Include an LLM usage statement in longform submitted work such as the essay or the exam. Suggestions on how to formulate it can be found here and here.

Hardware and software needs

- A PC with any operating system.
  - Windows users must have WSL (Windows Subsystem for Linux) activated.
  - MacOS users must have Homebrew installed.
  - NB: You do not need a new or powerful machine; heavy processing will be done in the cloud.
- A text editor or integrated development environment (IDE). The recommended option is VSCodium (telemetry-free VSCode).
- An agentic coding tool. The recommended option is OpenCode (open-source equivalent to Claude Code and Codex).

If you do not have administrator rights on your PC and cannot easily install external programs, let me know in advance so I can help you set up a browser-based alternative via GitHub Codespaces.

You will also need a user account with each of the following services:

- ChatGPT Edu for agentic coding.
- GitLab for version control.
- Hugging Face for AI model files.
- OpenRouter for LLMs via API.
- Verda for cloud computing.

All these services will be available at no cost, as ChatGPT Edu is provided by the university, GitLab and Hugging Face have free tiers, and OpenRouter has free API endpoints. Verda has kindly provided the class with free credits through their academic program. Each student will get £30 worth of credits, which represents around 200 hours on Tesla V100 (16GB VRAM) or about 70 hours on RTX 6000 (48GB VRAM), storage included. Just remember to shut down the instance after use so the meter does not run unnecessarily.

Learning R and Python

This is not a programming course in the traditional sense, and you will be expected to do most of the basic R and Python learning on your own. This

may sound daunting if you are a complete beginner, but it is eminently feasible with sustained effort. The hard part is getting started and wrapping one's head around some core concepts such as iteration and function building, but the session in Week 2 is designed to help with that. Once past that threshold, learning to code is essentially a matter of practicing on a regular basis. If you have studied a difficult foreign language such as Arabic or Russian, you will find that it is much easier to gain reading fluency in R and Python than in any of those other languages.

Here are five general tips and suggestions for your coding journey:

1. *Use an LLM as your personal tutor*. LLMs excel at coding tasks and can provide responses tailored to your needs. I recommend sitting down with an LLM at the start of the course and asking it to design an entire learning schedule for you, complete with exercises and example data from a domain that interests you. With an agentic tool (which we will learn more about in Week 3), you can even create a dedicated tutor agent who teaches in a style you like, remembers what your weak points are, and holds you accountable if you skip sessions!

2. *Do a little every day*. The practical process of coding is so different from other types of academic work that the mental transition into "coding mode" can be taxing unless it is fresh in your mind. It is a bit like exercise; leave too many days in between and *every* session becomes painful.

3. *Tinker*. Do not feel constrained by what is in the vignette or instructions you are following. Try out variants, test on different data, or explore a technique that interests you even if it is not on the syllabus. The worst that can happen is that you get an error message.

4. *Use your own data*. After you are done with a vignette on example data, try the same code on material that you know, for example some data from your thesis project. This will make the exercise much more interesting and relevant to you.

5. *Take notes*. Use a note taking tool such as Obsidian, Notion, or Joplin to create a library of key concepts and code snippets. The process of distilling readings into notes and articulating ideas in your own words makes you remember things better. Taking systematic notes from the start also leaves you with a searchable archive at the end that helps with revision. I recommend taking notes in Markdown and making use of Wikilinks to connect them.

There is an ocean of learning material online in the form of YouTube videos, free courses, and coding vignettes. You should feel free to explore and use whatever you find helpful, but my recommendation is to rely primarily on introductory textbooks, many of which are freely available online. Below is a curated selection, in rough order of difficulty.

R

- Daniel Dauber, "R Basics: The Very Fundamentals," in *R for Non-Programmers: A Guide for Social Scientists* (Bookdown.org, 2023).
- Pedro J. Aphalo, *Learn R: As a Language*, 2nd ed. (Chapman and Hall, 2024).
- Xijin Ge, Jianli Qi, and Rong Fan, *Learn R through Examples* (gexijin.github.io, 2022).
- Roger D. Peng, *R Programming for Data Science* (LeanPub, 2015).
- Hadley Wickham, *The Tidyverse Style Guide* (Bookdown, 2022).
- Felix Lennert, *An(Other) Introduction to R* (Bookdown, 2022).
- Alex Douglas et al., *An Introduction to R* (Bookdown, 2026).
- Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Grolemund, *R for Data Science* (Sebastopol, CA: O'Reilly, 2023).
- James D Long and Paul Teetor, *R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics*, 2nd ed. (O'Reilly Media, 2019).

Python

- Leodanis Pozo Ramos, "How to Use Python: Your First Steps," *realpython.com*, Jan. 25, 2021.
- Allen B. Downey, *Think Python: How to Think Like a Computer Scientist* (Green Tea Press, 2023).
- Melanie Walsh, *Introduction to Cultural Analytics & Python* (Jupyter Book, 2021).
- *The Python Tutorial* (python.org, 2001).

- Folgert Karsdorp, Mike Kestemont, and Allen Riddell, *Humanities Data Analysis: Case Studies with Python* (Princeton, NJ: Princeton University Press, 2021).
- Jake VanderPlas, *Python Data Science Handbook* (O'Reilly, 2016).

Data sources

For your assignments and general tinkering you might want to explore new datasets and primary source collections. Here is a small selection of potentially useful resources for those interested in the Middle East and the Islamic world:

- Internet Archive
- HathiTrust
- Arabic Language Rare Materials Collection at The Library of Congress
- Middle Eastern Collection at Harvard Library
- Arabic Collections Online
- المكتبة الشاملة
- أرشيف الشارخ للمجلات الأدبية والثقافية العربية
- Afghan Data
- awesome-public-datasets
- Kaggle datasets

Week 1: Working in plain text

Topics

- Computer fundamentals
- The shell
- The filesystem
- Writing tools
- Markup languages
- Typesetting with Quarto
- Version control

Reading and preparation

Do

- Set up a GitLab account.
- Install VSCodium (or another IDE) on your PC.
- Complete the assignment.

Watch

- What is a transistor? (YouTube, 2 min).
- Characters, Symbols and the Unicode Miracle (YouTube, 10 min).

Read

- Matthew Justice, *How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine* (San Francisco, CA: No Starch Press, 2020), chapter 4.
- Andrew Hunt and David Thomas, *The Pragmatic Programmer: From Journeyman to Master*, 20th anniversary (Boston, MA: Pearson Education, 2020), topics 16-19.
- Ian Milligan and James Baker, "Introduction to the Bash Command Line," *Programming Historian*, Sept. 20, 2014.

- Kieran Healy, “The Plain Person’s Guide to Plain Text Social Science,” kieranhealy.org, 2020, chapters 1-3.
- Robert A. Beezer and David Farmer, *Git for Authors* (pretextbook.org, 2016), chapter 2.

Tinker

- “Command Line Crash Course,” *mozilla.org*, Oct. 9, 2025.
- Dennis Tenen and Grant Wythoff, “Sustainable Authorship in Plain Text Using Pandoc and Markdown,” Programming Historian, Mar. 2014.
- “Authoring,” *Quarto.org*, undated.

Further reading/watching

- Anil Dash, “How Markdown Took over the World - Anil Dash,” *Anil Dash*, Jan. 9, 2026.
- Garret Christensen, Jeremy Freese, and Edward Miguel, *Transparent and Reproducible Social Science Research: How to Do Open Science* (University of California Press, 2019).
- Andreas Hallberg, “Unicode for Arabists,” *Uppercase Alif*, Oct. 30, 2022.
- Chris Miller, *Chip War: The Fight for the World’s Most Critical Technology* (New York, NY: Simon and Schuster, 2022).
- AT&T Archives: The UNIX Operating System (YouTube) (YouTube, 27 min).
- “The Missing Semester of Your CS Education” (series of accessible MIT lectures).

Week 2: Thinking computationally

Topics

- Algorithmic thinking
- R grammar (objects, data structures, functions, control structures, and iteration)
- Web scraping with R
- Text manipulation in R
- Regex
- Python basics

Reading and preparation

Do

- Make sure R and Python are installed on your system.
- Complete the assignment.

Watch

- Computational thinking (YouTube, 14 min).

Read

- Donald E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, 3rd edition (Reading, Massachusetts: Addison-Wesley, 1997), pages 1-11.
- Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship* (Boston, MA: Pearson Education, 2008), chapters 2-4.
- Alex Douglas et al., *An Introduction to R* (Bookdown, 2026), Chapter 3 (Data in R).
- Felix Lennert, *An(Other) Introduction to R* (Bookdown, 2022), Chapter 11 (Functional programming and iterations).
- Andy Halterman, "Crash Course in Python for R Users," *andrewhalterman.com*, Nov. 6, 2023.

- “Regular Expressions Quick Start,” June 19, 2025.

Tinker

- “Working with Files and Folders in R,” *R-bloggers.com*, 2021.
- Gaston Sanchez, “Basic Manipulations with ”Stringr” Functions,” in *Handling Strings With R* (Leanpub, 2021).
- William J. Turkel and Adam Crymble, “Manipulating Strings in Python,” *Programming Historian*, July 17, 2012.

Further reading/watching

- Python: The Documentary (YouTube, 1h 24 min).
- Doug Knox, “Understanding Regular Expressions,” *Programming Historian*, June 22, 2013.

Week 3: Agentic AI

Topics

- Neural nets and LLMs
- Agentic coding assistants
- Model and provider options
- Privacy and security

Reading and preparation

Do

- Install OpenCode.
- Make sure your ChatGPT Edu account is activated.
- Set up an OpenRouter account and obtain an API key.
- Complete the assignment.

Watch

- "Agentic AI - how bots came for our workflows and drudgery" (Financial Times/YouTube, 12 min).

Read

- Alexander Kustov, "Academics Need to Wake Up on AI," *Popular by Design*, Mar. 2, 2026.
- James Evans, Benjamin Bratton, and Blaise Agüera y Arcas. "Agentic AI and the next Intelligence Explosion." *Science* 391, no. 6791 (Mar. 19, 2026).
- Petar Radanliev. "Artificial Intelligence: Reflecting on the Past and Looking towards the next Paradigm Shift." *Journal of Experimental & Theoretical Artificial Intelligence* 37, no. 7 (Oct. 3, 2025): 1045–1062.
- Martin Monperrus, "Bootstrapping Coding Agents: The Specification Is the Program," *arXiv.org*, Mar. 18, 2026.
- Aidan Li, "An Opinionated Guide to Agentic Coding," *aidanli.dev*, Mar. 16, 2026.

- Sebastian Porsdam Mann et al. “Guidelines for Ethical Use and Acknowledgement of Large Language Models in Academic Writing.” *Nature Machine Intelligence* 6, no. 11 (Nov. 2024): 1272–1274.

Tinker

- “Setting Env Variables in Windows, Linux & MacOS: Beginner’s Guide - Configu,” *Configu.com*, Mar. 13, 2025.

Further reading/watching

- Jennifer King et al., “User Privacy and Large Language Models: An Analysis of Frontier Developers’ Privacy Policies,” *arXiv.org*, Sept. 5, 2025.
- claude-code-best-practice (resource hub).
- everything-claude-code (advanced implementation).

Week 4: Visualisation and map-making

Topics

- Good visualisation
- R's `ggplot` system
- Map making
- Tables and diagrams
- Data wrangling with `dplyr`

Reading and preparation

Do

- Complete the assignment.

Watch

- The Art of Data Visualization (YouTube, 8 min).

Read

- Edward R Tufte, *The Visual Display of Quantitative Information* (Cheshire, CT: Graphics, 1983), chapters 6, 8 and 9.
- Felix Lennert, *An(Other) Introduction to R* (Bookdown, 2022), chapter 5.
- Kieran Healy, *Data Visualization: Practical Introduction*, 2nd ed. (Princeton, NJ: Princeton University Press, 2026), chapters 3 and 7.

Tinker

- Hao Zhu, "Create Awesome HTML Table with Knitr::Kable and kableExtra," *haozhu233.github.io*, Jan. 18, 2024.
- "Diagrams," *Quarto.org*.

Further reading/watching

- Mel Moreno and Mathieu Basille, "Drawing Beautiful Maps Programmatically with R, Sf and Ggplot2," *r-spatial.org*, Oct. 25, 2018.
- Claus O Wilke, *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures* (Beijing: O'Reilly Media, 2019).
- Robert Kabacoff, *Modern Data Visualization with R* (Chapman and Hall/CRC, 2024).
- Winston Chang, *R Graphics Cookbook: Practical Recipes for Visualizing Data* (O'Reilly Media, 2018).
- Paul Murrell, *R Graphics*, 3rd (Chapman and Hall, 2019).
- Oscar Perpinan Lamigueiro, *Displaying Time Series, Spatial, and Space-Time Data with R*, 2nd ed. (Chapman and Hall/CRC, 2018).
- Robin Lovelace, Jakub Nowosad, and Jannes Muenchow, *Geocomputation with R* (CRC Press, 2019).
- Edzer Pebesma and Roger Bivand, *Spatial Data Science: With Applications in R* (r-spatial.org, 2025).

Week 5: OCR and audio transcription

Topics

- Optical character recognition
- Audio transcription
- Using cloud GPUs
- Using Hugging Face models
- Validation

Reading and preparation

Do

- Set up a Verda account.
- Set up a Hugging Face account.
- Complete the assignment.

Watch

- Parliamentary Archives: The Digitisation Process (YouTube, 4 min).

Read

- N. Sandhya, R. Krishnan, and D.R. Ramesh Babu. "A Language Independent Characterization of Document Image Noise in Historical Scripts." *International Journal of Computer Applications* (New York, USA) 50, no. 9 (July 2012): 11–18.
- Nikos Kontonasios, Yannis Tzitzikas, and Pavlos Fafalios, "Automating the Extraction of Structured Data from Large Newspaper Corpora Using Layout Analysis, OCR and Generative AI," *Journal on Computing and Cultural Heritage*, 2026.
- Yuliang Liu et al. "OCRBench: On the Hidden Mystery of OCR in Large Multimodal Models." *Science China Information Sciences* 67, no. 12 (2024).
- Christina Bunea, "Best Open Source Speech-to-Text (STT) Model in 2026," *Northflank*, Jan. 7, 2026.

- Fatma Zahra Besdouri, Inès Zribi, and Lamia Hadrich Belguith. “Arabic Automatic Speech Recognition: Challenges and Progress.” *Speech Communication* 163 (Sept. 1, 2024).
- Ahmed Fawzi Gad, “Measuring Text Similarity Using the Levenshtein Distance,” *paperspace.com*, 2020.

Tinker

- “Introduction to the Tesseract Package,” *ropensci.org*, Nov. 20, 2023.
- Hegghammer, Thomas, “daiR Quickstart,” *dair.info*, Apr. 14, 2021.

Further reading/watching

- Thomas Hegghammer. “OCR with Tesseract, Amazon Textract, and Google Document AI: A Benchmarking Experiment.” *Journal of Computational Social Science* 5, no. 1 (2022): 861–882.
- Benjamin Kiessling et al. “Advances and Limitations in Open Source Arabic-Script OCR: A Case Study.” *Digital Studies/Le champ numérique* 11, no. 1 (2021): 1–30.
- Yuki Shizuya, “Whisper Variants Comparison: What Are Their Features And How To Implement Them?,” *towardsai.net*, Nov. 14, 2024.
- Quinn Dombrowski, Tassie Gniady, and David Kloster, “Introduction to Jupyter Notebooks,” *Programming Historian*, Dec. 8, 2019.

Week 6: Data management and retrieval

Topics

- Databases
- Word embeddings
- Semantic search and RAG
- Semantic image search
- Named entity recognition
- Programmatic LLM use
- Docker containers

Reading and preparation

Do

- Complete the assignment.

Watch

- Palantir AIP (YouTube, 8 min).

Read

- Ian Foster et al., *Big Data and Social Science* (New York: Chapman and Hall/CRC, 2020), chapter 4.
- Jay Alammar, "The Illustrated Word2vec," *jalammar.github.io*, Mar. 27, 2019.
- Chris Haffenden and Justyna Sikora, "CLIP-Topic: Identifying Themes in Large Image Collections," *kb-labb.github.io*, Dec. 1, 2025.
- Shibingfeng Zhang and Giovanni Colavizza, "Named Entity Recognition of Historical Text via Large Language Model," *arXiv.org*, 2025.
- Rada Varga and Stefan Bornhofen. "Graph Based Modelling of Prosopographical Datasets. Case Study: Romans 1by1." *DHQwords* 18, no. 2 (2024).

Tinker

- “Structured Data,” *ellmer.tidyverse.org*, 2026.
- “OpenAI Python API Library,” *openai.com*, 2026.
- Yurij Michalevich, “Rclip: An AI-Powered Command-Line Photo Search Tool,” *mikhalevi.ch*, Aug. 25, 2021.

Further reading/watching

- Ruben Tak, “Talk to Your Files in a Local RAG Application Using Mistral 7B, LangChain and Chroma DB,” *medium.com*, Oct. 24, 2023.
- Giselle Gonzalez Garcia and Christian Weilbach, “If the Sources Could Talk: Evaluating Large Language Models for Research Assistance in History,” *arXiv.org*, 2023.
- Avery Blankenship, Sarah Connell, and Quinn Dombrowski, “Understanding and Creating Word Embeddings,” *Programming Historian*, Jan. 31, 2024.
- Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif, “Recent Advances in Named Entity Recognition: A Comprehensive Survey and Comparative Study,” *arXiv.org*, Dec. 20, 2024.
- T. Smits and M. Kestemont, “Towards Multimodal Computational Humanities. Using CLIP to Analyze Late-Nineteenth Century Magic Lantern Slides,” Workshop on Computational Humanities Research (2021).
- Kenneth Benoit and Akitaka Matsuo, “A Guide to Using Spacyr,” *spacyr.quanteda.io*, Mar. 4, 2020.
- Jonathan Reades and Jennie Williams, “Clustering and Visualising Documents Using Word Embeddings,” *Programming Historian*, Aug. 9, 2023.
- Sydney Anuyah, Victor Bolade, and Oluwatosin Agbaakin, “Understanding Graph Databases: A Comprehensive Tutorial and Survey,” *arXiv.org*, 2024.

Week 7: Text and image analysis

Topics

- Classification
- Sentiment analysis
- Topic modelling
- Computer vision tasks

Reading and preparation

Do

- Complete the assignment.

Watch

- Why Computer Vision Is a Hard Problem for AI (YouTube, 9 min).

Read

- Wouter van Atteveldt and Kasper Welbers, "Supervised Sentiment Analysis in R," *github.com*, Jan. 2022.
- Valerie Hase, "Tutorial 13: Topic Modeling," *bookdown.org*, Dec. 21, 2021.
- Melkamu Abay Mersha, Mesay Gemeda yigezu, and Jugal Kalita. "Semantic-Driven Topic Modeling Using Transformer-Based Embeddings and Clustering Algorithms." *Procedia Computer Science*, 6th International Conference on AI in Computational Linguistics, 244 (Jan. 1, 2024): 121–132.
- Sowmya Vajjala and Shwetali Shimangaud, "Text Classification in the LLM Era – Where Do We Stand?," *arXiv.org*, Feb. 17, 2025.
- "Computer Vision Tasks," *GeeksforGeeks*, July 23, 2025.

Tinker

- "Camel Tools," *github.com*.
- Abdul Majed Raja, "Object Detection in Just 3 Lines of R Code Using Tiny YOLO," *heartbeat.comet.ml*, Aug. 22, 2018.

- "BERTopic," *maartengr.github.io*, 2026.

Further reading/watching

- Justin Grimmer, Margaret E Roberts, and Brandon M Stewart, "Social Science Research and Text Analysis," in *Text as Data: A New Framework for Machine Learning and the Social Sciences* (Princeton University Press, 2022), 13–32.
- Ala' Alrababa'h and Lisa Blaydes. "Authoritarian Media and Diversionary Threats: Lessons from 30 Years of Syrian State Discourse." *Political Science Research and Methods* 9, no. 4 (2021): 693–708.
- Cornelius Puschmann and Mario Haim, "Automated Content Analysis with R," *content-analysis-with-r.com*, Sept. 2019.
- Julia Silge and David Robinson, *Text Mining with R: A Tidy Approach* (Sebastopol, CA: O'Reilly, 2017).
- Taylor Arnold and Lauren Tilton, *Humanities Data in R: Exploring Networks, Geospatial Data, Images, and Text* (Springer, 2015).
- Daniel van Strien et al., "Computer Vision for the Humanities: An Introduction to Deep Learning for Image Classification (Part 1)," *Programming Historian*, Aug. 17, 2022.
- Michael Clark, *An Introduction to Text Processing and Analysis with R* (Bookdown, 2018).
- Felix Lennert, *Text Mining for Social Scientists* (Bookdown, 2023).
- Emil Hvitfeldt and Julia Silge, *Supervised Machine Learning for Text Analysis in R* (CRC Press, 2021).
- Dominic Nyhuis et al., "Automated Video Analysis for Social Science Research 1," in *Handbook of Computational Social Science, Volume 2* (Routledge, 2021), 386–398.
- Michelle Torres and Francisco Cantú. "Learning to See: Convolutional Neural Networks for the Analysis of Social Science Data." *Political Analysis* 30, no. 1 (2022): 113–131.
- Yilang Peng and Yingdan Lu, "Computational Visual Analysis in Political Communication," in *Research Handbook on Visual Politics* (Edward

Elgar Publishing, 2023), 42–54.

- Jungseock Joo and Zachary C Steinert-Threlkeld. “Image as Data: Automated Content Analysis for Visual Presentations of Political Actors and Events.” *Computational Communication Research* 4, no. 1 (2022).
- Simon Hentschel, Konstantin Kobs, and Andreas Hotho. “CLIP Knows Image Aesthetics.” *Frontiers in Artificial Intelligence* 5 (2022).

Week 8: Advanced model use and AI ethics

Topics

- Prompt techniques
- Model training and fine-tuning
- Evaluation metrics
- Using local LLMs
- Research in the age of AI

Reading and preparation

Do

- Complete the assignment.

Watch

- Recall me maybe (Financial Times/YouTube, 14 min).

Read

- Lee Boonstra, "Prompt Engineering," *kaggle.com*, Feb. 2025.
- "Classification: Accuracy, Recall, Precision, and Related Metrics," *developers.google.com*, 2026.
- Emily M. Bender et al., "On the Dangers of Stochastic Parrots," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ACM Conferences (Mar. 3, 2021), 610–623.
- Dan Williams, "How AI Will Reshape Public Opinion," *conspicuouscognition.com*, Oct. 26, 2025.

Tinker

- Jeremy Howard, "Is It a Bird? Creating a Model from Your Own Data," *kaggle.com*, 2022.
- Khalid Sabban, "Fine-Tuning Large Language Models with LoRA," *kaggle.com*, June 2025.

Further reading/watching

- "Data Collection and Annotation Strategies for Computer Vision," *ultralytics.com*, 2026.
- "Model Training with Ultralytics YOLO," *ultralytics.com*, 2026.
- Akaash Kolluri et al., "Finetuning LLMs for Human Behavior Prediction in Social Science Experiments," *arXiv.org*, Nov. 5, 2025.
- Eivind Kjosbakken, "Fine-Tuning VLMs for Document Understanding," *Towards Data Science*, May 5, 2025.
- Transformers, the tech behind LLMs (YouTube, 27 min).
- Jack Lindsey et al., "On the Biology of a Large Language Model," *transformer-circuits.pub*, Mar. 27, 2025.
- Michael E. W. Varnum et al. "Large Language Models Based on Historical Text Could Offer Informative Tools for Behavioral Science." *Proceedings of the National Academy of Sciences* 121, no. 42 (2024).
- Wenyue Hua et al., "War and Peace (WarAgent): Large Language Model-based Multi-Agent Simulation of World Wars," *arXiv.org*, Jan. 30, 2024.
- Ian Milligan, *The Transformation of Historical Research in the Digital Age* (Cambridge University Press, 2022).