ECON/DCS 368: Big Data and Economics/Data Science for Economists

Professor Kyle Coombs (he/him/his)
Winter 2024

E-mail: `kcoombs@bates.edu`
Web: kylecoombs.com GitHub: @kgcsport

Course Website: https://github.com/big-data-and-economics
OH Link: https://calendar.app.google/XF36Ujpg9NcJbSD58

*You can get in touch with me via email, but please always write **[ECON 368]** in the subject. I am most quick and responding to course matters raised as GitHub Issues and Discussions posts.*

Note

This syllabus contains a rough outline of the course and may change in the future. If you have any questions, you should check with me. I reserve the right to make changes to this plan at any time during the semester.

Course Description

Economics is at the forefront of developing statistical methods for analyzing data collected from uncontrolled sources. Since econometrics addresses challenges in estimation such as sample selection bias and treatment effects identification, the discipline is well-suited to analyze large or unstructured datasets. This course introduces practical tools and econometric techniques used to conduct empirical analysis on topics like equality of opportunity, education, racial disparities, and more. These skills include data acquisition, project management, version control, data visualization, efficient programming, and tools for big data analysis. The course also explores how econometrics and statistical learning methods cross-fertilize and can be used to advance knowledge on topics like inequality, education, racial disparities, health care, and more where large volumes of data are rapidly accumulating. We will also cover the ethics of data collection and analysis.

Course Objectives

After this course is done, you should know how to: 1. Organize empirical projects that are replicable, reproducible, and collaborative using Clean Code principles 2. Acquire, clean, and wrangle many data formats including CSV, JSON, XML, DTA, Shapefiles, and more using APIs 3. Use data to generate key insights about economic opportunity, inequality, and racial discrimination 4. Understand the differences between prediction, causality, and description, and when to apply each 5. Demonstrate facility with R, RStudio, GitHub, and computer file management

---

Course Materials

Course notes, assignments, extra readings, recordings, and all other materials are available on the GitHub class materials repository. *The notes are adapted from Grant McDermott's course at the University of Oregon, Tyler Ransom's course at the University of Oklahoma, Raj Chetty's course at Harvard University, Nick Huntington-Klein's Econometrics course, and others listed in this syllabus.*

Software requirements

All the software requirements for this course are open-source and/or free. Please aim to have ***R*** and ***Rstudio*** installed by the start of our first lecture. Other installation will be a part of Problem Set 0. I will be available for installation troubleshooting during the first week of the semester. If you want a detailed tutorial on how to achieve a perfect working setup, I can think of no finer guide than Jenny Bryan et al.'s http://happygitwithr.com/ (see esp. sections 4 – 15).

R and RStudio We will mainly be using the statistical programming language ***R***. Please make sure that you also install the ***RStudio*** integrated development environment (IDE). You can download both from the links below. If you already have R and RStudio installed, please make sure that you have the latest versions. You can check this by running `R.Version()` in the R console. If you have an older version, please update.

- **R:** Download
- **RStudio IDE:** Download

Git/GitHub Desktop

- **Git:** Installation instructions
- **GitHub:** Create an account and register for an education discount here

We'll use GitHub to store and share code and data. You'll need to create an account and a private repository for your final project. I also advise you to register for an education discount, which gives you access to major services including 180 core-hours of free access to GitHub Codespace servers.

GitHub is a fantastic tool for collaboration and tracking changes to your work over time, but it can be a bit intimidating at first. If you're new to GitHub, I'd recommend starting with the GitHub Desktop app. It's a bit easier to use than the command line.

During the semester, I will ask you to maintain your code and writing in your GitHub repository. This will allow me to provide feedback directly on the code you develop and in your written work. When you finish, you will also have a single repository dedicated to this final project that you can share widely with potential employers, graduate schools, and others.

Also, if you are oscillating between working on your personal computer and the lab computers, GitHub can allow you to easily sync your work between the two.

I plan to work with GitHub Classroom to distribute assignments and provide feedback. This will allow me to provide feedback directly on the code you develop and in your written work. When you finish, you will also have a single repository dedicated to this final project that you can share widely with potential employers, graduate schools, and others.[1]

[1] Caveat: Certain features of GitHub Classroom are currently in development and I may elect separate methods for distributing assignments and providing feedback.

LaTeX software A LaTeX software distribution that is compatible with RMarkdown.

- **TeX Live:** Installation instructions: Use the "easy install" option for your operating system.
- **tinytex** Installation instructions: If you are using RStudio, you can install tinytex by running `install.packages("tinytex")` in the R console. Then run `tinytex::install_tinytex()` to install the LaTeX distribution.

Recommended but not required:

You are ready to start this course once you have installed R, RStudio, and Git (as well as created an account on GitHub), and some TeX software. Make sure they are fully up-to-date.

Here are some other useful tools:

Visual Studio Code VSCode is free and open-source, and is available for Windows, Mac, and Linux. You can download it at https://code.visualstudio.com/download. Once you have installed VSCode, you will need to install a variety of extensions. We will cover installations during the problem set (or as they become necessary), but here is a list:

- The *R* extension by REditorSupport – https://code.visualstudio.com/docs/languages/r
- *LaTeX Workshop* by James Yu – https://marketplace.visualstudio.com/items?itemName=James-Yu.latex-workshop

Other stuff

- GitHub Copilot by GitHub
- ChatGPT - Genie AI by Genie AI
- Anaconda or PIP - largely used for Python installations, there are a few quality of life packages for R that are distributed via Anaconda or PIP.
- Radian - Radian allows you to use VSCode similar to how you would use RStudio. You will be able to run code directly into a terminal with `Ctrl+Enter`, but also have access to GitHub CoPilot coding assistance.

Operating system-specific recommendations:

- **Linux:** You should be good to go.
- **Mac:** Install the Homebrew package manager. I also recommend that you make sure your C++ toolchain is configured/open. Just download the macOS Rtools installer and follow the instructions.
- **Windows:** Install Rtools. While it's not essential, I also recommend that you install the Chocolatey package manager for Windows. Furthermore, please install the Windows Subsystem for Linux (WSL) and the Ubuntu distribution. Instructions here.

I will provide instructions for any further software requirements as the need arises; i.e. when we get to the relevant lecture. Each week's lectures will be posted by the preceding Sunday on the course website. Each lecture lists all the *R* packages and external libraries (if relevant) required for a particular class. Please ensure that you have them installed *before* we start class.

Textbook and other readings

There's no set textbook for this course. Readings from select free sources are listed below:

Writing, Research, and Presenting

- The Introduction Formula by Keith Head
- The Middle Bits by Marc F. Bellemare
- The Conclusion Formula by Marc F. Bellemare
- Code and Data for the Social Science: A Practitioner's Guide by Matthew Gentzkow and Jesse M. Shapiro
- How to Give an Aplied Micro Talk by Jesse Shapiro

Econometrics, Statistics, Data Science with R examples

- Causal Inference: The Mixtape by Scott Cunningham
- The Effect by Nick Huntington-Klein
- Mostly Harmless Econometrics by Joshua D. Angrist and Jörn-Steffen Pischke
- Data Science for Economists and Other Animals
- An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani
  - ISLR Labs
- Practical Econometrics with R by Christoph Hanck
- Spatial Data Science by Edzer Pebesma and Roger Bivand
- Data Visualization: A practical introduction by Kieran Healy
- Curated List by Nathan Tefft
- Library of Statistical Techniques (LOST)

On R

- R For Data Science by Hadley Wickham and Garrett Grolemund
- Advanced R by Hadley Wickham
- Geocomputation with R by Robin Lovelace, Jakub Nowosad, and Jannes Muenchow
- Posit Cheatsheets
- R Programming for Data Science by Roger D. Peng
- Bates Alumni Eli Mokas and Ian Ramsay's RStudio Tutorial
- RStudio Gallery
- R Markdown: The Definitive Guide by Yihui Xie, J. J. Allaire, and Garrett Grolemund

Staying organized

- Code and Data for the Social Sciences: A Practitioner's Guide by Matthew Gentzkow and Jesse Shapiro
- Coding for Economists: A Language-Agnostic Guide
- happygitwithr by Jenny Bryan

Large Language Models You are actively encouraged to use generative AI assistants in this class. These can be used to improve your code, refine your writing, iterate on your ideas, and more.

- Sign-up for ChatGPT
- Sign-up for GitHub CoPilot (Note: you do not signup through this organization, you signup through your own personal GitHub account as a student.)
- Tips to get better with ChatGPT

- Integration of AI with R

Taking a step back, one of the goals of this course (and most Data Science courses) is to make you aware of the incredible array of instruction material that is freely available online. I also want to encourage you to be entrepreneurial. In that spirit, many of the lectures will follow a tutorial on someone's blog tutorial, or involve reproducing an existing study with open source tools. Each lecture will come with a set of recommended readings, which I expect you to at least look over before class.

Prerequisites

Prerequisites: ECON 255 and ECON 260 or ECON 270 The course assumes background in econometrics and statistics.

Teaching Assistant

There is no teaching assistant for this course. The course does have a Course-Attached Tutor (CAT), who is a student who has taken the course before and is available to help you with the course. The CAT for this course is Charlie Berman. He will hold office hours and review sessions. The CAT will also grade your problem sets and final project.

Course-Attached Tutor Charlie Berman is our Course-Attached tutor. He will host office hours in the SASC and will be available for individual appointments. His hours are:

- **SASC Drop-in Hours**: MTR 2:30-4
- **Evening Help Session:** R 7:30-9 (room assignment to come)

You can also schedule private office hours at the Calendar Link: https://calendar.app.google/2Abyp3LqY3NPeg4u8.

He can help you troubleshoot *R*. He does not have solutions to the problem sets, but he can help you figure them out.

Student Academic Support Center

Scheduled hours for R held in the Student Academic Support Center (SASC) of the Library are:

- Sunday - 7:30-9pm
- Monday - 12-1pm, 2:30pm-4pm
- Tuesday - 12-2:30pm, 6-7:30pm
- Wednesday - 11am-1pm, 6-7:30pm
- Thursday - 12-4pm, 6-7:30pm
- Friday - 11am-12pm

Grading policy

Component	Weight	Graded
6 × problem sets (12.5% each)	50%	Top 5
1 × 5-minute presentation	5%	Top 1
1 × GitHub participation	5%	Overall

Component	Weight	Graded
1 × group final project	30%-40%	In parts
1 x Lewiston Hackathon	0%-10%	Optional
Classroom participation	Bonus up to 2.5%	Discretion
Open source material contribution	Bonus up to 2.5%	Provide evidence

Problem sets

Throughout the course you will engage in problem sets that deal with actual data. These may seem out of step with what we do in class, but they are designed to get you to think about how to apply the tools we learn in class to real data.

- Problem sets are coding assignments that get you to play with data using R
- They are extremely challenging, but also extremely rewarding
- With rare exceptions: You will not be given code to copy and paste to accomplish these data cleaning tasks, but instead given a set of instructions and asked to figure out how to write code yourself
- You are encouraged to work together on problem sets, but you must write up your own answers (unless it is a group assignment)
- All problem sets will be completed and turned in as GitHub repositories
- I will drop the two lowest problem set grades

What you will turn in:

- Each problem set will be posted as a GitHub repository, which you will fork, set to private, and then clone to your computer (instructions provided in each problem set)
- You will then work on the problem set on your computer or a Codespaces server, and push your code to GitHub (push often!) (**Note: You have to push in Codespaces. If you delete your Codespace without pushing changes, you will lose your answers.**)
- For each problem set, you will turn in modular code (i.e. separate files do separate things) that accomplishes the tasks outlined in the problem set
- You will also turn in a `.Rmd` file that contains your answers to the questions in the problem set along with a knitted `.html` or `.pdf` of your `.Rmd`
  - This `.Rmd` will "source" the code you wrote, so I can easily run your code from start to finish by `knitting`
- Your problem sets will have a sensible folder structure that is easy to navigate (name folders `code`, `data`, `output`, etc.)
- You will turn in your problem sets by pushing your code to GitHub.

Grading

Your problem sets are (generally) graded on four criteria:

1. Submission via GitHub (10%): Did you use GitHub to stage, commit, and push your code? Did you submit the assignment on time? Did you submit the assignment in the correct format?
2. Quality of code (30%): Is it well-commented? Is it easy to follow? Can I run it?

- Any scripts needed to run your code should be included in the repository and sourced in the `.Rmd` file
- Write code that automates as much of the process as possible. For example, if you need to download a file, write code that downloads the file automation
- If you cannot figure out how to automate a step, you can write a comment explaining what I need to do to run your code (you will lose very few points)

3. Quality of presentation of graphs and tables (30%): Are they well-labeled? Do they have titles? Do they have legends? Are they formatted well?
4. Quality of answers (30%): Are they clear? Do they answer the question?

I will provide feedback and a grade in a `feedback` branch of your problem set repository. That will let me add feedback without overwriting your work in the `main` branch.

Solutions The solutions are made public within a week of the problem set being posted.

Improving your grade In an effort to incentivize you to see coding as an ongoing process of learning and improvement, I will allow you to improve the coding and presentation quality portions of your grade on any problem set. However, you cannot just copy and paste the solutions.

Instead, you must provide carefully commented explanations of each step of the code – whether from the solutions or of your own invention. This is a great way to learn, but it is also a lot of work.

Example. You might write add a comment like this to the top of your code:

```
# Create directories, suppress warning that the directory already exists.
suppressWarnings({
    dir.create(data)
    dir.create(documentation)
    dir.create(code)
    dir.create(output)
    dir.create(writing)
})
```

Submission process To be eligible to resubmit to improve your grade, you must have submitted an initial version of the problem set on time.

1. View my feedback on the `feedback` branch of your problem set repository.
2. Fix your problem set answers and comment your code as needed. Write "CORRECTED" in all caps next to any changes.
3. Push changes to the `main` branch of your problem set repository.
4. Navigate to the `Issues` tab of your problem set repository and create a new issue titled "Resubmission for Problem Set X". Briefly describe your changes in the body of the issue and tag my username, @kgcsport.
5. **Deadline for resubmissions**: All resubmissions must be pushed within one week of the solutions being posted.

Within your own private problem set repository, you can create an `Issues` tab within the Settings tab for interfacing only with me and any group partners.

Requests for reconsideration On occasions, you may disagree with the grade you received on a problem set. Here are my policies for reconsideration:

- **Deadline for requests**: All requests for reconsideration must be submitted within one week of the solutions being posted.
- **Full regrade**: Any request for reconsideration will result in a full regrade of your problem set. This means that your grade can go up or down.
- **Regrading high scores**: If you scored a 90 percent or above on a problem set, I will not change your grade. This is not because I do not want to help you, but because we both have limited time and I want to focus my efforts on cases where an incorrectly graded problem set could significantly impact your grade in the course. **This does not apply to re-submissions. This only applies to the cases where you want me to review your score in full separate from corrections and re-submissions.**

If you would like reconsideration, please raise an `Issue` in your private problem set repository. Title the issue "Reconsideration request for Problem Set X". Briefly describe your request in the body of the issue and tag my username, @kgcsport.

Within your own private problem set repository, you can create an `Issues` tab within the Settings tab for interfacing only with me and any group partners.

Presentations

Each of you will give a 5-minute presentation summarizing a key lecture reading, or an (approved) software package/platform.

Please sign up here at the start of the semester.

Final Project

You will write a final project over the course of the semester as part of a group. Further details are available here. If you participate in the Hack-a-thon, your final project will be worth 30 percent of your grade. If you do not participate in the Hack-a-thon, your final project will be worth 40 percent of your grade.

Lewiston Hack-a-thon

This semester, we will be working with the City of Lewiston to help them solve a problem using data. Specifically, we will help the city understand how to use existing administrative data to complement, and at times substitute, for survey data.

We will specifically be engaging in a Hack-a-thon. A hack-a-thon is a short (often 24 hours), intense period of collaboration between a group of people to solve a problem. Scheduling is still in the works.

The Hack-a-thon is planned to be optional and replace a quarter of the final project grade.

Data Requests Several weeks before the hack-a-thon, we will brainstorm datasets that your group would like the City of Lewiston to provide for you. You will then write a short report on how you would use those datasets to solve a problem.

What you will do

- Compete in groups of 3-4 to each propose solutions to the same problem
- Present your solution to a group from the City of Lewiston
- Write a short report on your solution
- Maintain all code and necessary documentation to the City of Lewiston in a GitHub repository
- Provide any additional documentation the City of Lewiston requests

Your solution may include a variety of things, including:

- A data visualization
- Suggestions of new databases to maintain
- Examples from similar cities that have tackled these problems

GitHub participation

Participation on GitHub is 5 percent of your grade. Please use GitHub Discussions and `Issues` to ask questions about the course materials and problem sets. You can also suggest improvements to the course materials. Here are the guidelines:

- When starting a discussion, posting an issue, or suggesting a pull request, please use a clear title (e.g. Problem Set 1: Question about Question 2) and description ("What does term X mean?")- If posting about an error you are encountering, follow these steps:
  - Briefly state the expected behavior
  - Write the minimal code needed to reproduce the error (a minimally reproducible example)
  - Write the full error message you are receiving
  - Write the steps you have taken to troubleshoot the error
- If posting a clarifying question about someone's post, follow these steps:
  - Briefly clarify what you are confused about
  - Suggest potential interpretations of the post
- If posting about a suggestion to improve the class materials, follow these steps:
  - Briefly state the improvement you are suggesting
  - Write the steps you have taken to troubleshoot the error
  - If you are suggesting a change to the course materials, please fork the repository, make the change, and submit a pull request
- Be kind to one another. Coding is hard. We are all learning.

This policy guidelines are taken from stackoverflow.com. You can read more about how to ask a good question here and how to answer a question here.

I will rate participation based on the following criteria:

- Are you posting thoughtful questions? Follow the guidelines on stackoverflow for posting a good question.
- Are you replying to questions? Follow the guidelines on stackoverflow to write a good answer.
- Do you pull request improvements to the course materials? This can be a typo fix, a bug fix, or a new feature.

Note: I hope to add Issue templates across the entire organization. All in good time.

For each problem set, use the `Issues` tab for that specific problem set. For course materials, please use the organization Discussions tab.

I will be monitoring the GitHub `Issues` tab for each repository and will participation points to those who are actively engaging per the guidelines from stackoverflow. To receive full credit, you must be asking thoughtful questions and thoughtfully answering each other's questions. Thoughtful questions come after you've spent some time re-reading your code, Googling, and working with ChatGPT to try to solve the problem yourself first. Thoughtful answers may not solve the problem, but they should be clear, concise, good faith efforts to help. You will receive a lower participation grade if you do not follow the posting guidelines.

The goal is to encourage you to work together to solve problems. This is one of the most important skills you can take away from this, and really any, course. I also want to incentivize you to think carefully about how you post. Be kind and respectful, as much as you endeavor to be clear, concise, and helpful.

Furthermore, I want you to take ownership over your learning. You get more out of a course when you are actively engaged in the material. Actively engaging on this repository and suggesting changes to the course materials is a very tactile way to do that.

Bonus points:

There are several opportunities for bonus points during the semester:

1. A 2.5% bonus on your final grade for issuing a *pull request* to any open source material. This can be to fix a typo or to fix a bug in the code.
2. A 2.5% participation bonus on your final grade that I will award at my discretion.
3. I will offer a 2.5% participation bonus to the person with most "good faith" posts/answers in GitHub Issues and Discussions within this organization. "Good faith" means:
    - The posts are made to actually ask about a problem you are having with a problem set/your final project or to answer someone else's question
    - The posts/answers follow the guidelines above
    - The posts are not duplicates of existing posts – you have to search before you post to see if folks are already working on this problem.
    - The posts and answers are not spam – I'm a child of the 90s. I can spot spam from a mile away.
    - The posts and answers are respectful and constructive.
    - An improvement on existing answer is fine, if it actually improves on the existing answer. The intent of this bonus is to encourage you to work together to solve technical problems in a way that resembles professional software development and data analysis on this platform. This is one of the most important skills you can take away from this, and really any, course.
4. I offer a bonus point for each typo corrected on problem sets *and* solutions. This is capped at 10 points per student per problem set. You must pull request and/or raise an Issue on the corresponding GitHub repository to get credit.

I have given instructions on how to execute a pull request of a *specific* commit (instead of your entire commit history) in the FAQ.

Extensions

I offer extensions for two things:

1. Major health issues

2. Major family emergencies

Please flag either with Bates Reach and email me with the subject "[ECON 368] *subject here*", so I can be aware of the situation. Together we'll figure out an appropriate extension.

Rough schedule

The most update schedule is available on the course website.

Date	Day	Topic	Do before class	Due
Data Science Basics				
2024-01-11	Th	Introduction to Big Data (.html, .pdf, .Rmd)	Read and Install Ch 1, 4-8 of happygitwihtr	
2024-01-16	T	Git slides (.html, .pdf, .Rmd)	Work through Ch 9-19 of happygitwithr	
2024-01-18	Th	Empirical Organization slides (.html, .pdf, .Rmd)	Read Code and Data for Social Sciences	Problem Set 0 due 1/18 at 11am
2024-01-23	T	Data Tips (.html, .pdf, .Rmd)	Read Code and Data for Social Sciences	
2024-01-25	Th	R Basics (.html, .pdf, .Rmd), Data Tips (.html, .pdf, .Rmd)	Watch basics of RStudio by Bates alumni Eli Mokas and Ian Ramsay	
2024-01-30	T	Data Table (.html, .pdf, .pdf) Tidyverse (.html, .pdf, .Rmd)	Ch 1 DS4E	Problem Set 1 Due 1/29 at 11:59:59pm
2024-02-01	Th	CSS (.html, .pdf, .Rmd), Scraping Notes by Jesus Fernández Villaverde and Pablo Guerrón	SelectorGadget (Chrome), ScrapeMate (Firefox)	
2024-02-06	T	APIs (.html, .pdf, .Rmd)	JSONView, Sign-up and register for Personal API Key	
2024-02-08	Th	Catch-up		
Causal Inference				
2024-02-13	T	Regression Review (.html, .pdf, .Rmd)	Read Effect Ch 13 or Mixtape Ch 2, Watch Causal Effects of Neighborhoods	Problem Set 2 due 02/12 at 11:59:59pm

Date	Day	Topic	Do before class	Due
2024-02-15	Th	Opportunity Atlas (.html, .pdf, .Rmd) and Spatial Analysis (.html, .pdf, .Rmd)	Watch Geography of Upward Mobility in America starting at 39min	
2024-02-20	T	Winter Break		
2024-02-22	Th	Winter Break		
2024-02-27	T	Causal Inference (.html, .pdf, .Rmd)	Read Effect Ch 13 or Mixtape Ch 2, Watch Causal Effects of Neighborhoods	
2024-02-29	Th	Difference-in-differences (.html, .pdf, .Rmd), Panel data and two-way fixed effects (.html, .pdf, .Rmd)	Watch first 40min of Teachers and Charter Schools	
2024-03-05	T	Regression Discontinuity Design (.html, .pdf, .Rmd), RDD activity (.html, .pdf, .Rmd)	Read Effect Ch 20 or Mixtape Ch 6	
2024-03-07	Th	Catch-up		Problem Set 3 due 3/8 at 11:59:59pm
Machine Learning				
2024-03-12	T	Bootstrapping, Functions & Parallel Programming (.html, .pdf, .Rmd), Bootstrapping activity (.html, .pdf, .Rmd)	Refer to Chapters 2-4 of DS4E, Chapter 9 of R for Data Science	
2024-03-14	Th	SQL (see work by Tyler Ransom)		

Date	Day	Topic	Do before class	Due
2024-03-19	T	Intro to Machine Learning (.html, .pdf, .Rmd), ISLR tidymodels lab (.html), Oregon Schools Decision Tree application by Cianna Bedford-Petersen, Christopher Loan, & Brendan Cullen (.html)	Read Athey & Imbens (2019), Mullainathan and Spiess (2017), Refer to ISLR 8.1	
2024-03-21	Th	March Recess		
2024-03-26	T	Machine Learning: Bias and Judicial Decisions (.pdf by Raj Chetty and Greg Bruich)	Watch Improving Judicial Decisions	Problem Set 4 due 3/25 at 11:59:59pm
2024-03-28	Th	Causal Forests (.html, .pdf, .Rmd), Application: Causal forests with **grf**	Refer to ISLR Ch 6.1, 6.2	
2024-04-02	T	Regression regularization/penalization (.html, .pdf, .Rmd), Application (.html, .pdf, .Rmd)	Read ISLR 8.2	
2024-04-04	Th	Regular expressions, WordClouds (.html, .pdf, .Rmd), Tidy text activities (.html, .pdf, .Rmd)	Read Gentzkow (2019): Text as Data	
2024-04-09	T	Sentiment Analysis (.html, .pdf, .Rmd)	Read Stephens-Davidowitz (2014)	
2024-04-11	Th	Topics Modeling, LLMs	Read Ash and Hansen (2023): Text Algorithms	Problem Set 5 due 4/11 at 11:59:59pm
If time		AI and bias	Read Rambachan et al (2020) and Cowgill et al. (2019)	

Course Policies

During class

We will be doing active coding projects during class, so please bring your personal laptops. Please refrain from using computers for anything but activities related to the class. Phones are prohibited as they are rarely useful for anything in the course. Eating and drinking are allowed in class, but please refrain from it affecting the course. Try not to eat your breakfast/lunch in class as the classes are typically active.

Academic Integrity and Honesty

Students are required to comply with the Bates policy on academic integrity in the Code of Student Conduct at https://www.bates.edu/student-conduct-community-standards/student-conduct/code-of-student-conduct/. Don't cheat. Don't be that person. Yes, you. You know exactly what I'm talking about. See https://www.bates.edu/student-conduct-community-standards/student-conduct/academic-integrity-policy/ for a detailed explanation of academic integrity.

Academic integrity is always important, but is especially important to a senior thesis. IT is at the heart of the mission and values of Bates College and is an expectation of all students.

Plagiarism: Violations of academic integrity are serious and can result in severe consequences at both the course and college level. Some intermediate assignments and the final proposal require writing about other polished work. You may not borrow text from the original papers without proper attributions. Plagiarism of any kind for any assignment in this class will result in a grade sanction up to and including failing the course. Depending on the circumstances of the violation, there may be a referral to the dean of students for possible institutional actions. If you are unsure about issues of academic integrity or what is expected or permissible, just ask!

Attendance:

- Attendance is essential for learning; you are warmly invited, encouraged, and expected to attend all class meetings. Attendance will be important not only for your learning, but also for our ability to build a community together and maintain a sense of connection and commitment to one another. Your presence in class matters.
- I recognize that extraordinary circumstances may prevent you from attending class. If you are unable to attend class due to illness or if you are unsure whether to attend class, please contact Health Services for guidance. If for any reason you will not be in class, it is your responsibility to inform me in advance via email. It is also your responsibility to figure out a way to get notes and make up any work that you missed in your absence.
- If we meet for class online over Zoom, attendance is still important. Throughout the course, we will have in-class activities that will require you to come to class prepared to participate. In the event that we meet for class from different physical locations, we are still one class and one community. I will expect you to be prepared to meet at the regularly scheduled time (U.S. Eastern-time). If you are unable to meet virtually at the regularly scheduled times, it is your responsibility to email me to make alternative arrangements.

Artificial Intelligence

I encourage each of you to make use of artificial intelligence-driven digital assistants, like ChatGPT and Github CoPilot. These tools are not a substitute for your own ingenuity, but instead a complement as they are incredibly useful for tasks like coding or proofreading. Please cite whether and where you used ChatGPT in your written work, as you would cite your (human) sources.

Policies on Incomplete Grades and Late Assignments

Throughout the course, I will provide you feedback on your work. It is your responsibility to turn in your work on time.

End of course: If an extension is not authorized by the instructor, department, or college, an unfinished incomplete grade will automatically change to an F after either (a) the end of the next regular semester in which the student is enrolled (not including short-term), or (b) the end of 12 months if the student is not enrolled, whichever is shorter.

Incompletes that change to F will count as an attempted course on transcripts. The burden of fulfilling an incomplete grade is the responsibility of the student.

Accommodations for Disabilities

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with the Office of Accessible Education and Student Support (AESS) in Ladd Library G35. For more information on Bates' policy on working with students with disabilities, please see the AESS webpage on Requesting Services (https://www.bates.edu/accessible-education-student-support/requesting-services/how-to-register-for-accommodations/).

Non-Discrimination Policy

Bates College provides equality of opportunity in education and employment for all students and employees. Accordingly, Bates College affirms its commitment to maintain a work environment for all employees and an academic environment for all students that is free from all forms of discrimination.

Discrimination based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation is a violation of state and federal law and/or Bates College policy and will not be tolerated. Harassment of any person (either in the form of quid pro quo or creation of a hostile environment) based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation also is a violation of state and federal law and/or Bates College policy and will not be tolerated. Retaliation against any person who complains about discrimination is also prohibited. Bates's policies and regulations covering discrimination, harassment, and retaliation may be accessed at https://www.bates.edu/here-to-help/policies/equal-opportunity-policy/. Any person who feels that he or she has been the subject of prohibited discrimination, harassment, or retaliation should contact the Director of Title IX & Civil Rights Compliance and Title IX Coordinator, Gwen Lexow, at titleix@bates.edu or https://www.bates.edu/here-to-help/make-a-report/.

Accommodations for Families

If you are a parent or guardian of a child, and you are unable to attend class and care for that child for class one day, please be in touch in case you need further accommodations. You are invited to attend the lecture

via Zoom or watch it asynchronously if that will make it easier to not miss course material.
