Python Data Collection and Management for Public Policy Research

Day 1: Course Intro, Computational Social Science

Blake Miller[†]

1 July, 2024

[†]Assistant Professor, Department of Methodology, London School of Economics and Political Science (E-mail: b.a.miller@lse.ac.uk)

Agenda for Today

- Introductions
- Motivations
- Logistics
- Why Python?
- What is Programming?
- Best Practices for Excellent CSS Research
- Coding Session: Getting Set Up, a silly recipe

Introductions

About Me

- Assistant Professor in Computational Social Science at the Department of Methodology, LSE
- PhD in Political Science and Scientific Computing, University of Michigan
- Past work as a software engineer and founder at A16z-funded Silicon Valley startups.
- **My research:**
  - Chinese politics and political communication
  - Applied machine learning for text data
- **Contact:**
  - `b.a.miller@lse.ac.uk`
  - `www.blakeapm.com`

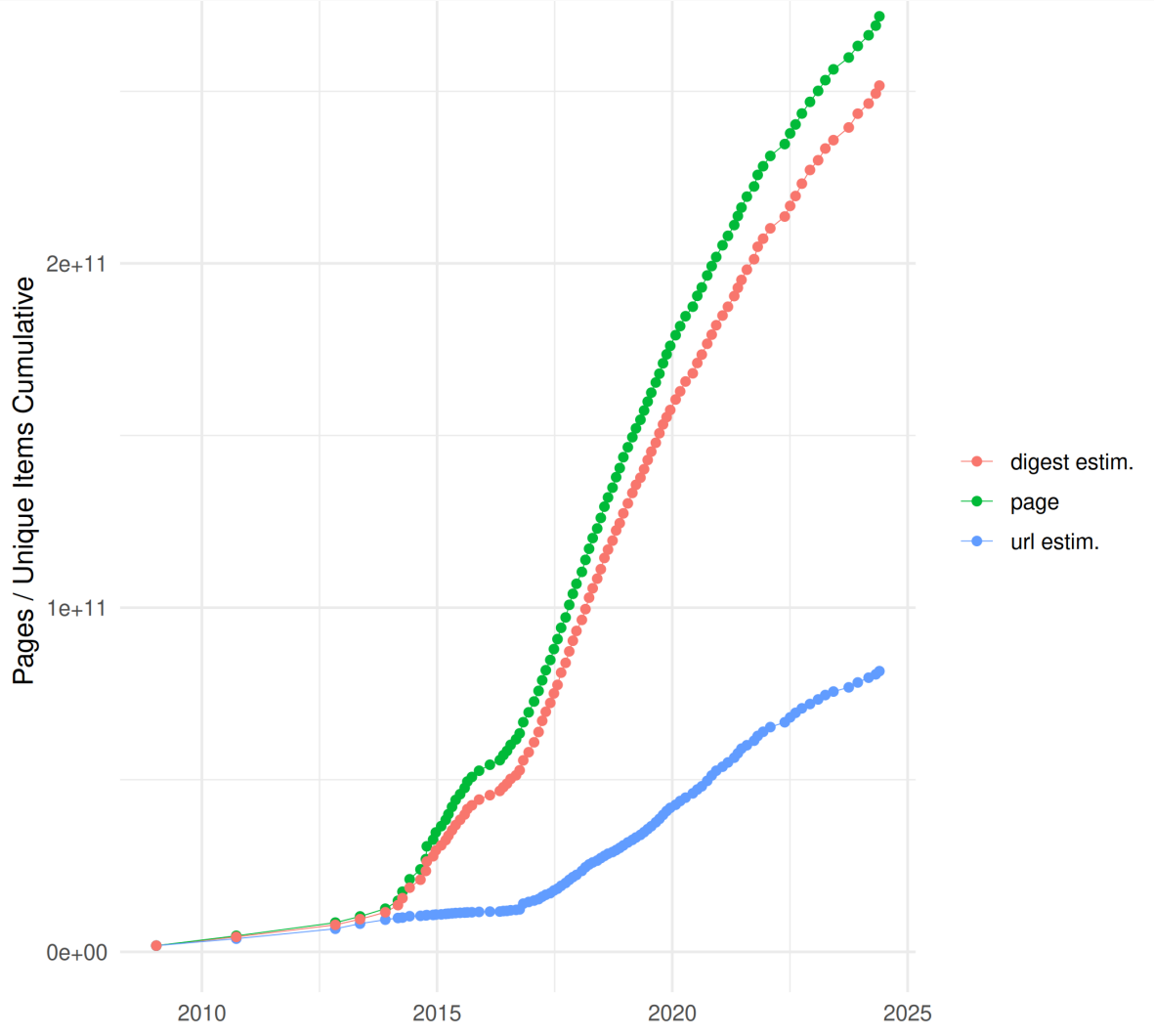
About You

Tell us a bit about yourself:

- What year in school are you?
- What do you study?
- What interests you in learning Python?
- Do you have any previous coding experience?

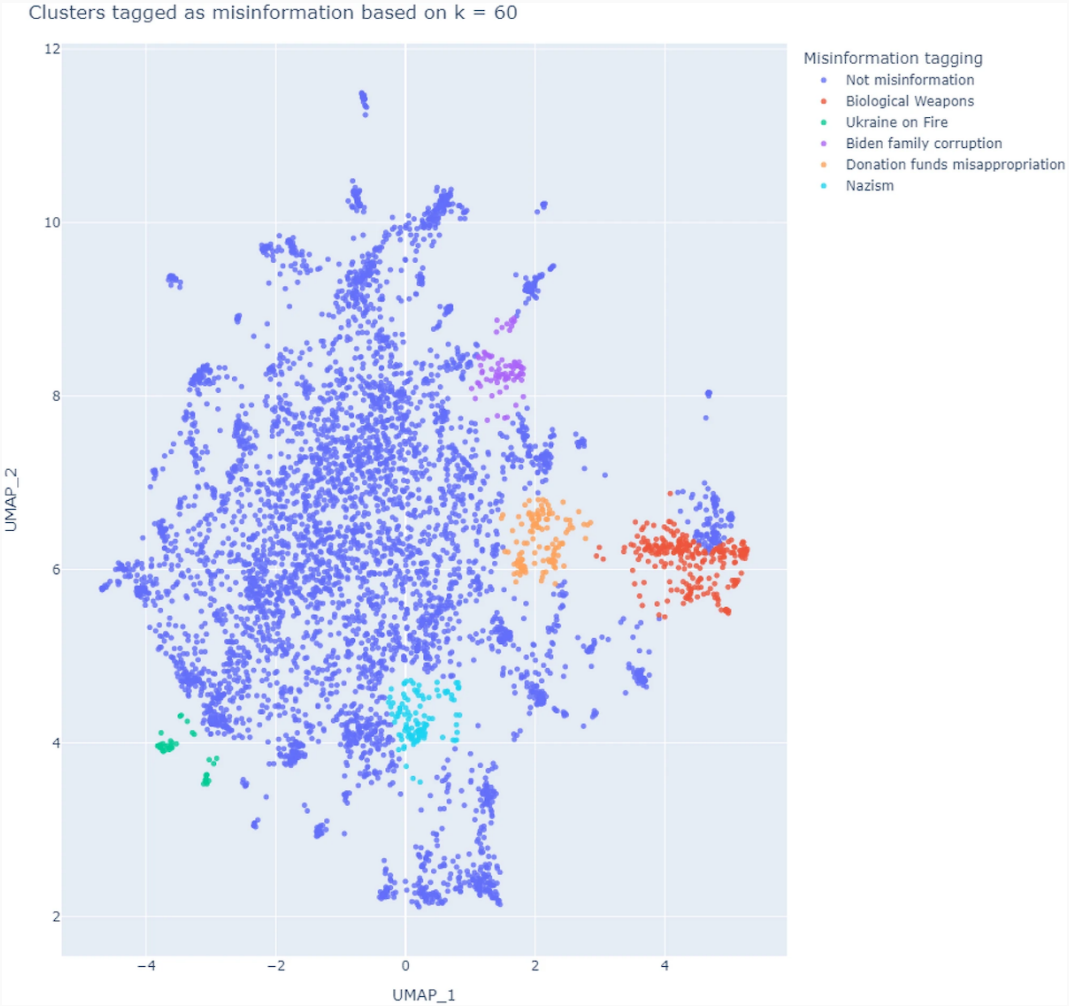
Motivations and Course Goals

Motivations: Size of Available Data is Increasing



Cumulative size of the Common Crawl. Source: Common Crawl

Motivations: Complexity of Available Data is Increasing



2-dimensional representation of popular tweets shared by pro-Russian accounts labeled as misinformation. Source: [Lai et al., 2024]

Course Goals: Introduction to Essential Tools

- This course will introduce a broad array of foundational tools and concepts, which may initially seem overwhelming.
- **Remember:** It's not necessary to master or memorize everything presented.
- Our primary objective is to **equip you with fundamental tools and skills** essential for **high-quality computational social science research**.
- Please feel free to ask for clarification at any point; I aim to clarify any jargon and ensure things are as accessible as possible.

Course Goals: Basics and Beyond

- Given our limited time of 10 sessions, we will focus on core concepts while providing resources for further exploration of advanced topics.
- Some lectures will include "Building on _____" slides, which will guide you on where to find more in-depth information and additional learning materials.

Course Goals: Setting the Stage for Advanced Learning

- My approach to Python includes using the Python shell, text editors , and the command line interface (CLI).
  - Alternative methods such as Jupyter Notebooks and integrated development environments (IDEs) are also viable.
- The objective of this course is to impart essential skills and frameworks I wish I would have had early in my career.
- I'll teach to best practices as I see them, you are free to adapt your workflow as you develop your own style.

Logistics

Course Schedule

- **Lectures**: July 1-5, 8-12 14:00-16:30
  1. 14:00-14:50: Lecture
  2. 14:50-15:00: Quiz (2, 4, 9, 11 July) or Q&A
  3. 15:00-15:10: Break
  4. 15:10-16:00: Applied coding exercises

Course Outline

Day	Topic
1	Course Introduction, Computational Social Science
2	Introduction to the Command Line, Sublime Text
3	Introduction to `git` and Github
4-6	Basic Python
7	Introduction to Data and Data Ethics
8-9	Introduction to `pandas`
10	Obtaining Data from the Web

Readings

- Readings for each session are in the syllabus.
- Textbooks used in this class:
  - **Think Python: How to Think Like a Computer Scientist**, 2nd Edition (English, Chinese)

Assignments

- Four quizzes on readings and lecture (60%)
  - 2 July
  - 4 July
  - 9 July
  - 11 July
- Final problem set (40%), due 19 July

Final Problem Set

- Using a dataset we discuss in class, among other things, you will be asked to:
  - Use `git` and Github to manage versions of the code as you develop it.
  - Write functions to clean and prepare the raw data.
  - Merge, concatenate, subset the data in various ways.
  - Store the data in a variety of formats.
  - Describe trends in the data, draw insights.

Why Python?

Why Python?

- It's open source and free
- It is simple ("high level")
- Very commonly used across sectors
- Enormous developer community
- Very similar to other languages you may want to use or have already used (like R)

Why Python?

- Python offers state-of-the-art software packages for many methodologies of great interest to computational social scientists:
  - Natural Language Processing (NLP): natural language toolkit (`nltk`), classical language toolkit (`cltk`), `spaCy`
  - Machine Learning (ML): `scikit-learn` (basic ML), `TensorFlow` (deep learning, low level), `pyTorch` and `keras` (deep learning, high level)
  - Network Analysis: `NetworkX`
  - Computer Vision: `OpenCV`, `scikit-image`
- This is particularly important given the need for more collaboration between computer scientists and social scientists.

What is Programming?

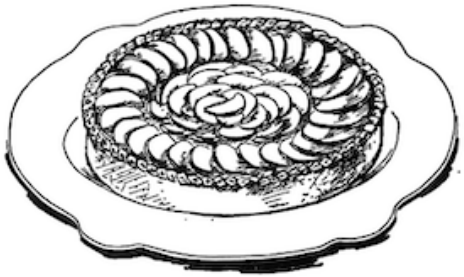
Programs as Recipes?

TARTE AUX POMMES

[Apple Tart—warm or cold]

The classic French apple tart consists of a thick, well-flavored applesauce spread in a partially cooked pastry shell. Over it thinly sliced apples are placed in an overlapping design of circles. After baking, it is coated with apricot glaze.

Apple Tart



For 8 people

A 10-inch partially cooked pastry shell set on a baking sheet, page 634	Use the sweet short paste on page 633 for your pastry shell.
4 lbs. crisp cooking or eating apples 1 tsp lemon juice 2 Tb granulated sugar A 2-quart mixing bowl	Quarter, core, and peel the apples. Cut enough to make 3 cups into even ⅛-inch lengthwise slices and toss them in a bowl with the lemon juice and sugar. Reserve them for the top of the tart.
A 10-inch heavy-bottomed pan: enameled saucepan, skillet, or casserole A wooden spoon ⅓ cup apricot preserves, forced through a sieve ¼ cup Calvados (apple brandy), rum, or cognac; or 1 Tb vanilla extract ⅔ cup granulated sugar 3 Tb butter Optional: ½ tsp cinnamon, and/or the grated rind of 1 lemon or orange	Cut the rest of the apples into rough slices. You should have about 8 cups. Place in the pan and cook, covered, over low heat for about 20 minutes, stirring occasionally, until tender. Then beat in the ingredients at the left. Raise heat and boil, stirring, until applesauce is thick enough to hold in a mass in the spoon.
	Preheat oven to 375 degrees.
	Spread the applesauce in the pastry shell. Cover with a neat, closely overlapping layer of sliced apples arranged in a spiral, concentric circles, or as illustrated at the beginning of this recipe.
A cake rack or serving dish ½ cup apricot glaze, page 593 2 cups heavy cream, or *crème fraîche*, page 16	Bake in upper third of preheated oven for about 30 minutes, or until the sliced apples have browned lightly and are tender. Slide tart onto the rack or serving dish and spoon or paint over it a light coating of apricot glaze. Serve warm or cold, and pass with it, if you wish, a bowl of cream.

What is a Recipe?

- A recipe is essentially an **algorithm**:
  1. A sequence of simple, procedural steps.
  2. A flow control process that specifies when each step is executed.
  3. A means of determining when to stop.
- Together, these elements define a recipe as a methodical approach to reaching a final outcome, much like how a program operates on inputs to produce outputs.

Computers as Machines that Follow Recipes

- How to capture a recipe in a mechanical process:
  - Fixed program computers (e.g., calculators) operate on a preset sequence of operations.
  - Stored program computers can store and execute instructions dynamically.
- This flexibility allows stored program computers to adapt the sequence of instructions based on conditional logic, similar to how a chef might alter a recipe based on available ingredients or tools.

Stored Program Computer

- In stored program computers:
  1. The sequence of instructions is stored internally.
  2. Instructions are built from a predefined set of primitive operations including arithmetic and logic, simple tests, and data movement.
- A special program (interpreter) executes each instruction in order, using tests to change the flow of control through the sequence, stopping when the program is complete.
- This methodical execution is analogous to following a complex recipe step-by-step until the dish is ready.
- More on this on day 4...

Introduction to Computational Social Science

Introduction to Computational Social Science (CSS)

- The CSS subfield has grown over the last decade as tool for leveraging big data in social sciences.
- Making use of computers and programming languages like Python to extract, organize, retrieve, and analyze data related to human behavior and social institutions.
- Programming is important here because of the expanding size of datasets of importance to social scientists.
  - Relationship between the size of data and the time needed to compute is often not linear.
  - This means that data analysis—in many cases—gets much more difficult as n increases
  - Example: traveling salesman problem

Example: The Traveling Salesperson Problem

- Problem: Find the shortest route for a salesman to visit each city once and returning to the first city.
- Time Complexity: A measure of the relationship between the number of inputs (cities) and the time needed to compute an output.
- For the TSP, searching through all permutations (brute force approach) has an super exponential time complexity:
  - 5 cities: $(5-1)! = 24$ possible routes
  - 10 cities: $(10-1)! = 362,880$ possible routes
  - 100 cities: $(100-1)! = 9.3 \times 10^{155}$ possible routes
- We use big O notation to classify computing problems into "types"
- The big O notation for brute force search of TSP is $O(n-1)!$, which belongs to the category of $O(n!)$ computing problems

Example: The Traveling Salesperson Problem

But computational methods and algorithmic tricks can improve on brute force (a bit):

- 5 cities: $(\frac{5-1}{2})! = 2$ possible routes
- 10 cities: $(\frac{10-1}{2})! = 52$ possible routes
- 100 cities: $(\frac{100-1}{2})! = 4.2 \times 10^{63}$ possible routes

Examples of Data Sources of Interest in CSS

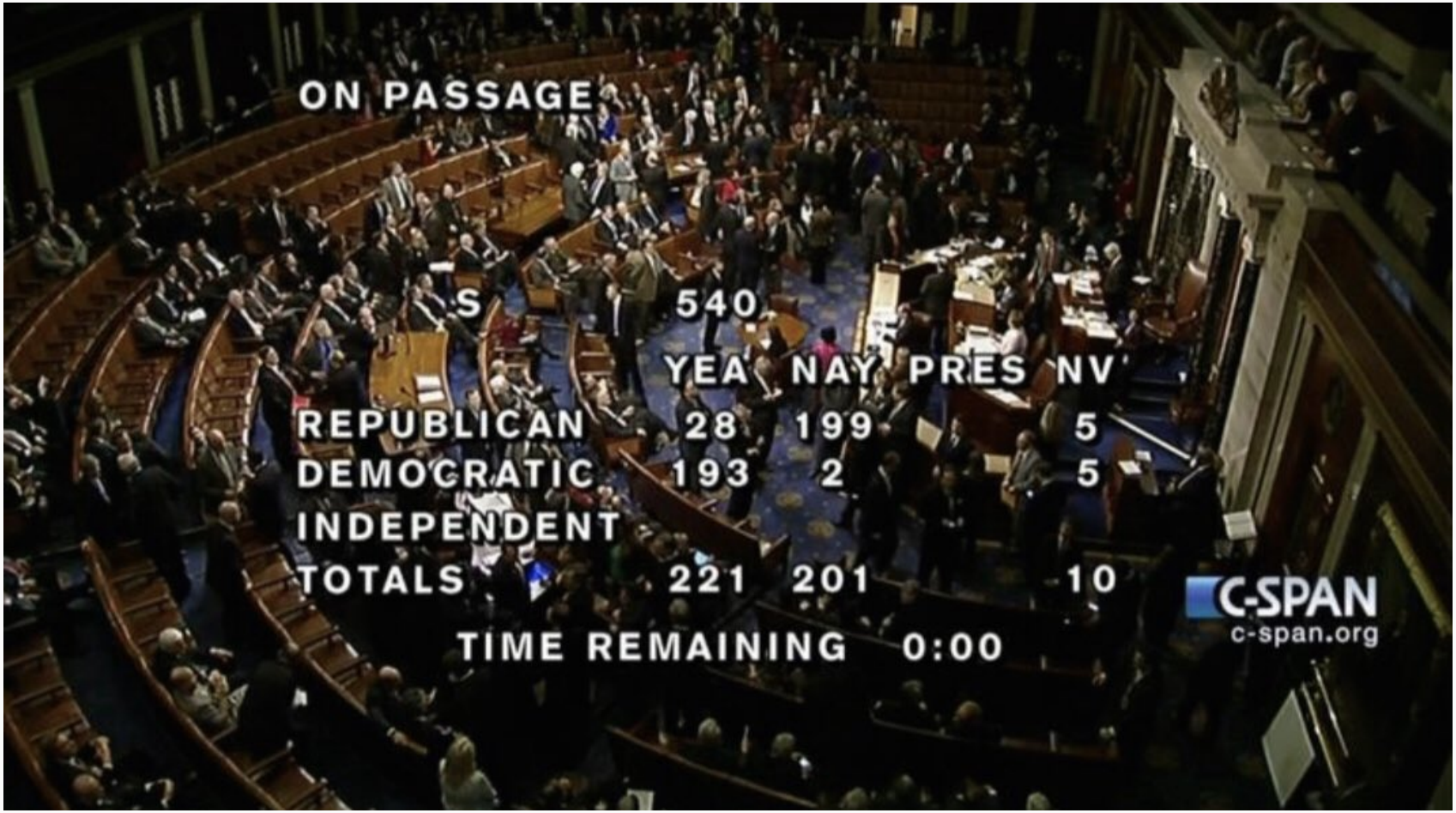
- Digital Traces: Data from social media, mobile phone usage, internet browsing, and online transactions.
- Experimental Data: Controlled experimental setups for studying behavioral responses.
- Simulated Data: Use of models to simulate social interactions and predict outcomes.

Methodological Approaches in CSS

- Machine Learning: the use of models to approximate the functional relationship between inputs and outputs from a sample of data. This approximated function can be used to predict and model complex relationships.
- Network Analysis: using models of how nodes (e.g., individuals) in a network interact and are connected; using this information to infer and predict the behavior of these nodes.
- Natural Language Processing/Quantitative Text Analysis: quantifying, modeling, or measuring features of textual data to better understand patterns in content, behavior, opinion, etc.
- Computer Vision: quantifying, modeling, or measuring features of image and video data to better understand how visual information captures social meaning.

Some Examples of CSS Research

Computer Vision [Dietrich, 2021]



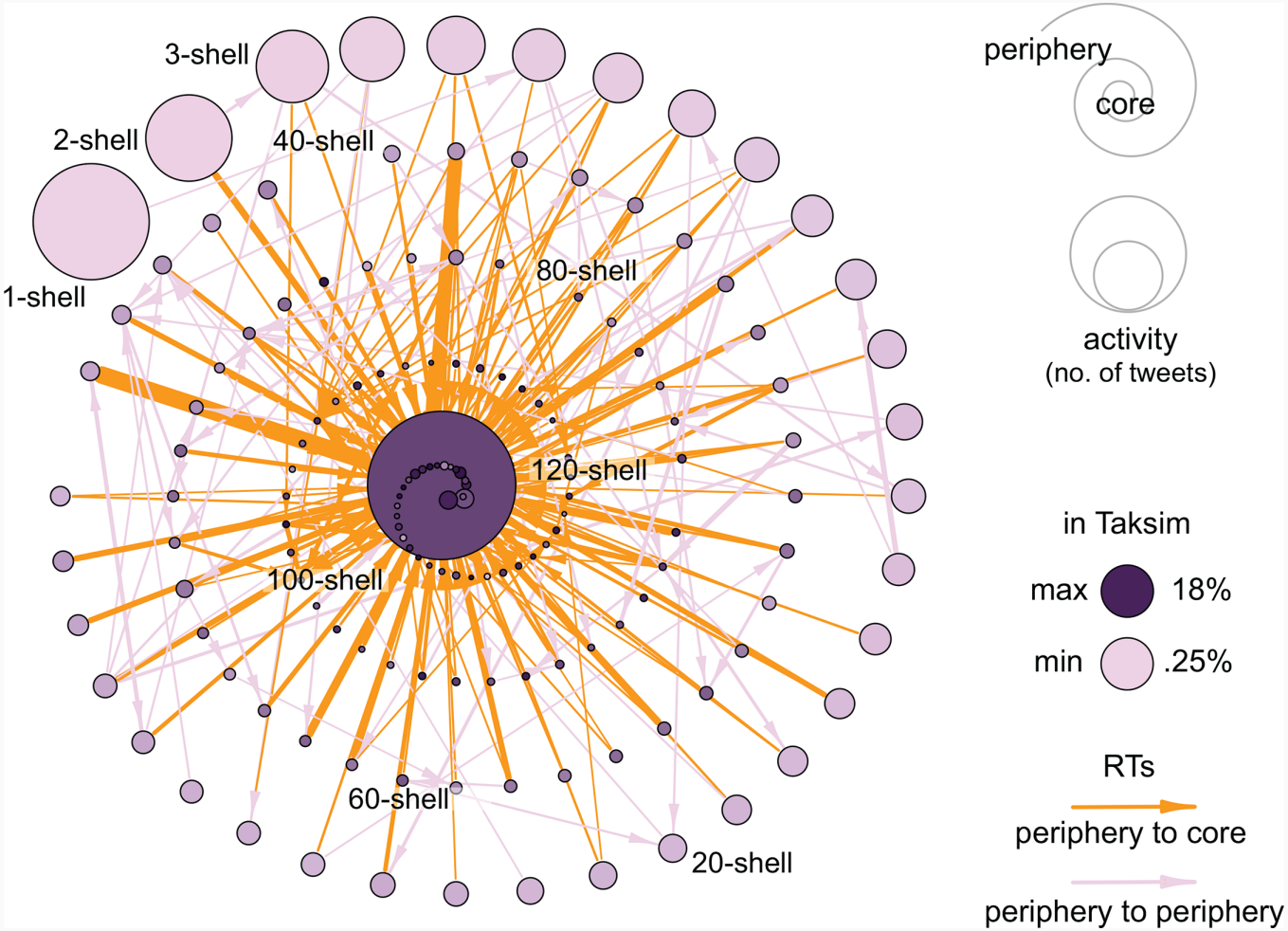
Application of computer vision to video data to the measure social polarization of the U.S. House of Representatives.

Computer Vision [Miller and Thompson-Brusstar, 2024]

Areas of Study in CSS

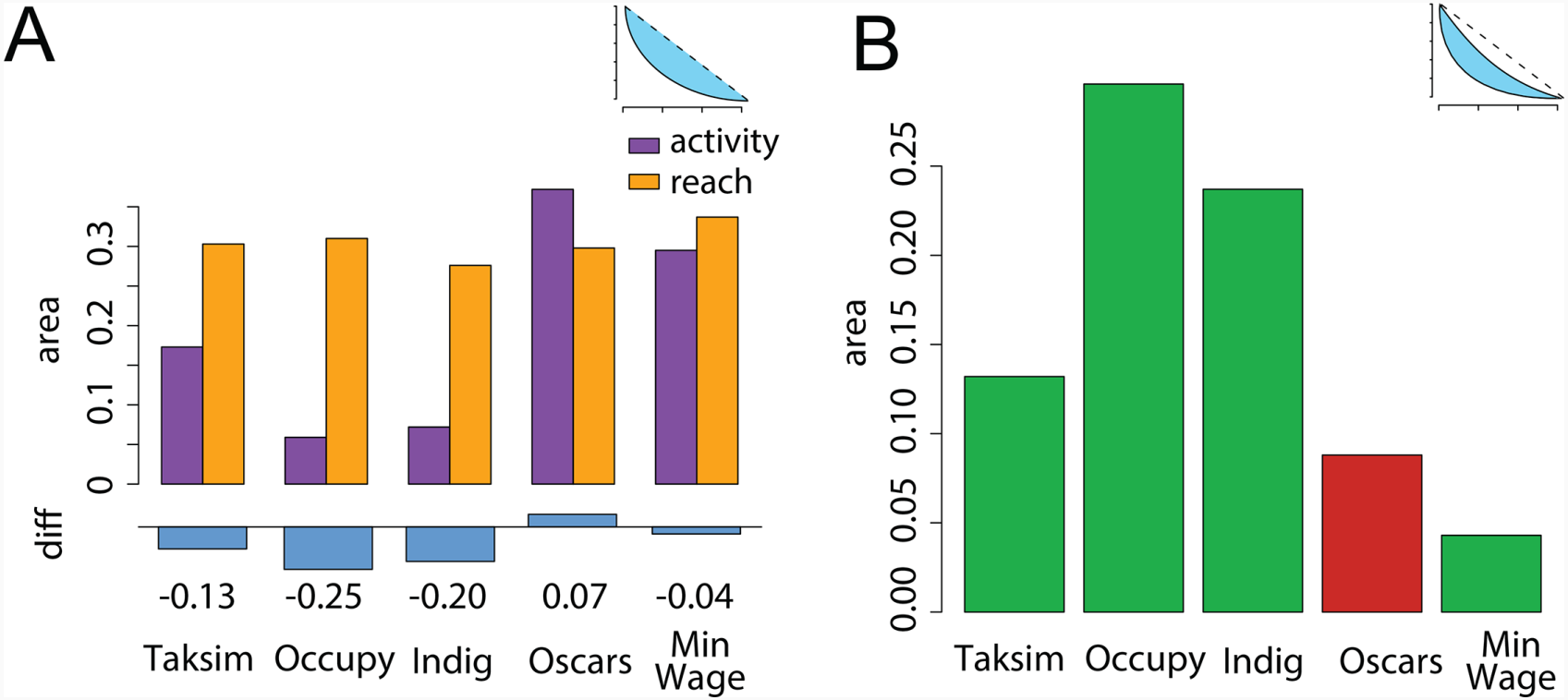
- Social Networks: Analysis of social structures, social relationships.
- Human Behavior: Study of group and individual behavior.
- Policy and Governance: Analysis of policy impact, societal norms, digital governance, state-society interaction.

Social Networks [Barberá et al., 2015]



Main RQ: How important are peripheral participants in social networks in the spread and impact of protest messages?

Social Networks [Barberá et al., 2015]

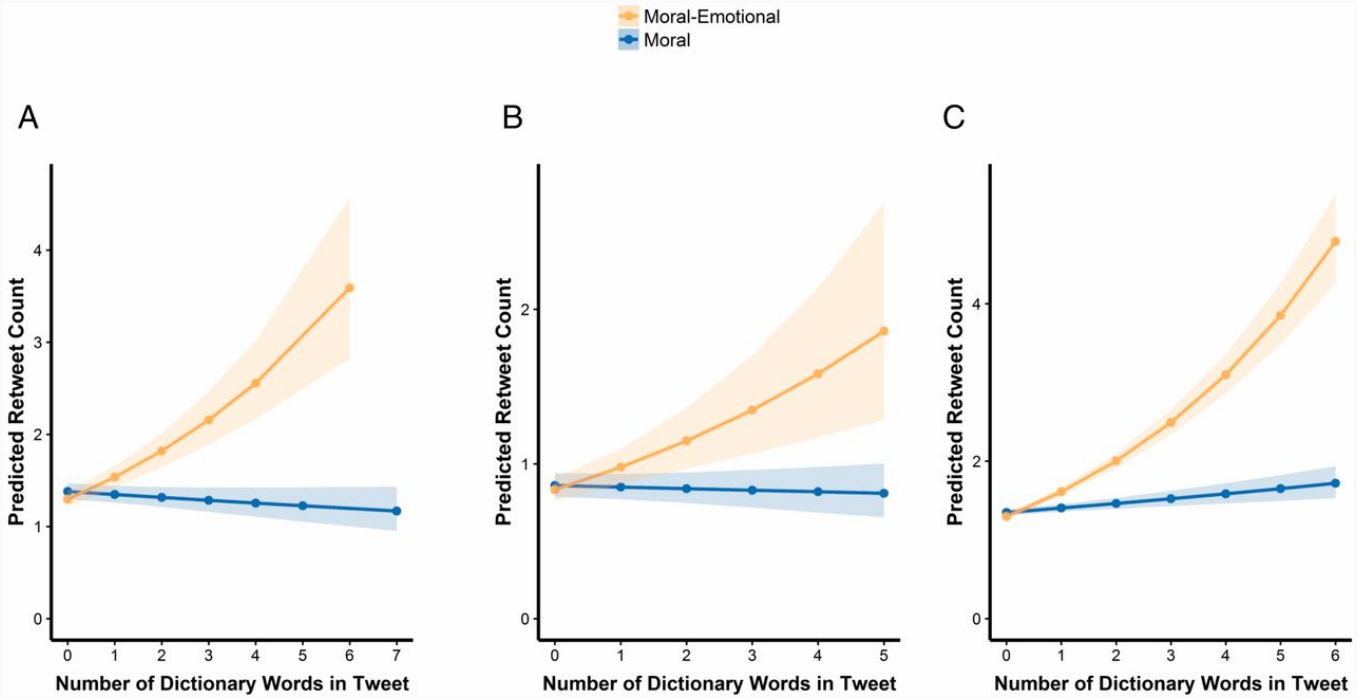


Main RQ: How important are peripheral participants in social networks in the spread and impact of protest messages?

Key Findings in [Barberá et al., 2015]

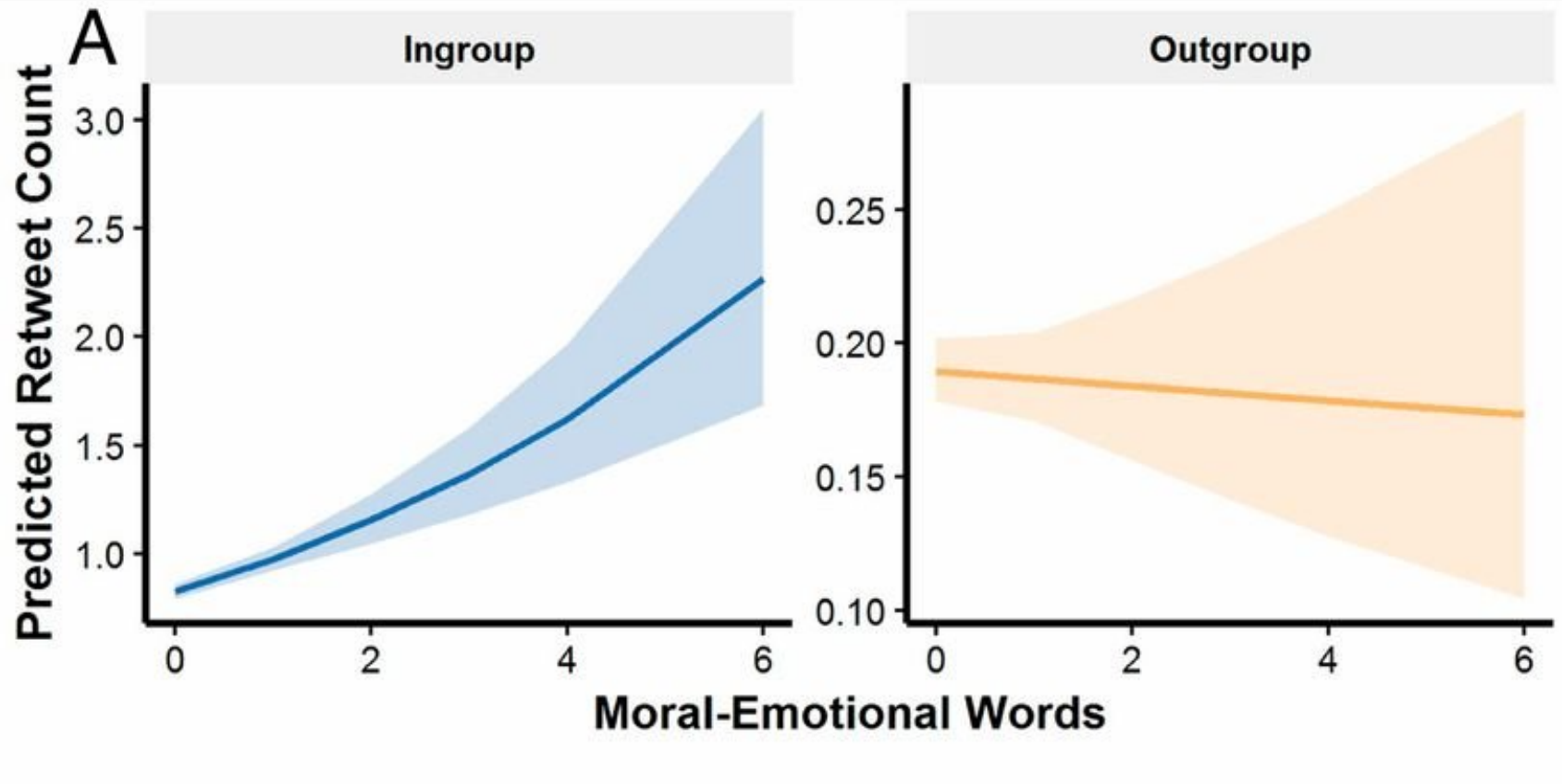
- Using network analysis, they examine the hierarchical structure of protest networks. In large protest networks, they find there is a small, active core, and a larger, less active periphery.
- Peripheral users play a critical role in expanding the reach of protest messages (this is surprising!)
- This core-periphery distinction is much less pronounced difference in non-protest networks.
- Removing peripheral participants reduces the overall reach of protest messages by more than 50%.

Human Behavior [Brady et al., 2017]



Main RQ: How do moral and emotional appeals on social media platforms influence their diffusion within and across ideological group boundaries?

Human Behavior [Brady et al., 2017]



Main RQ: How do moral and emotional appeals on social media platforms influence their diffusion within and across ideological group boundaries?

Key Findings in [Brady et al., 2017]

- Emotional expressions linked to moral appeals significantly boost the spread of messages (20% more likely to be retweeted for each additional word).
- The spread of moral-emotional messages is stronger within ideologically similar networks (liberal or conservative).
- The spread of moral-emotional messages is stronger for members of the in-group (compared to the out-group).

Contributions of [Brady et al., 2017]

- Demonstrates relationship between moral-emotional content and the spread of messages online.
- Contributes to study of echo chambers, message diffusion, group polarization.

Policy and Governance [Farrell, 2016]

Main RQ: How does corporate funding influence the thematic content and ideological language of climate-contrarian organizations over time?

Key Findings

- Using quantitative text analysis (topic models) and social network analysis, they find that organizations with corporate funding are more likely to produce texts that polarize climate change debate.
- The thematic content of these texts is influenced by corporate funding.

Issues Faced by Researchers

- Data Privacy: Challenges in maintaining privacy while using personal data.
- Algorithmic Bias/Fairness: Issues related to non-representative samples or biased data collection.
- Data Access: Difficulties in accessing high-quality, large-scale data, especially from private companies.
- Research Ethics: Developing ethical guidelines for handling sensitive data.
- More about this on day 7.

Best Practices for Excellent CSS Research

Best Practices in Scientific Computing

- Organization is important!
  - Store code and data in a logical, clean and understandable way
  - Keep your code well-commented and well-documented
  - Use version control software to track changes
- This is important because:
  - It will help you make changes in response to feedback/peer-review
  - It will help you spot problems in your work and make it easier to make necessary changes
  - It helps others (including you in 3 years) understand and replicate your analysis.

Best Practices for Data Files

- Make redundant backups.
  - In cloud services (e.g., Dropbox, OneDrive, etc.)
  - Set aside local backups on labeled external drives.
  - Use version control software like git/Github
  - Think: What if your drive fails? What if you accidentally delete files on a cloud platform?
- Choose helpful and informative file names:
  - "data.csv" does not tell you anything about the data. If you needed to find it later, how would you search for it?
  - An example of an informative file name: "kremlin_complaints_data_06-16-2024_raw.csv"
  - Store data files in subfolders of project folders to aid in file access.

Best Practices for Data Collection

- Cause no harm:
  - Be aware of and follow ethical and legal requirements related to data collection and scraping.
  - Limit your load on servers if you are scraping.
- Document your process:
  - Record the specific parameters of the data collection (when, how data were accessed)
  - Keep data in raw form when possible (e.g., storing html files allows reanalysis if a parsing error is detected after collection.)
  - Retain code used to collect the data for your records.

Best Practices for Data Processing/Cleaning

For each stage of data pre-processing, document and store:

- What each component does (in code comments or file names)
- Serialize (save to disk) data output at each step in pre-processing. This helps limit the use of computational resources, and aids in spot-checking output.
- Record notes documenting exactly was done (in a Supplementary Materials section in your paper, or in a README.md)

Best Practices with Code Files

- Number scripts according to sequence of execution
  - Example: `01_scrape_data.py`, `02_process_raw_data.py`, `03_models.py`, `04_figures.py`
- Use triple quote blocks at the start of each script to describe what the code is doing
- Add informative comments before each section of code to explain what the code is doing in plain language.
- When using git/Github, at each update write informative commit messages.
- Write DRY code (**d**o not **r**epeat **y**ourself)
  - Any task you run more than once should be a function
  - Write simple, clear, descriptive variable names

Best Practices for Organizing Projects

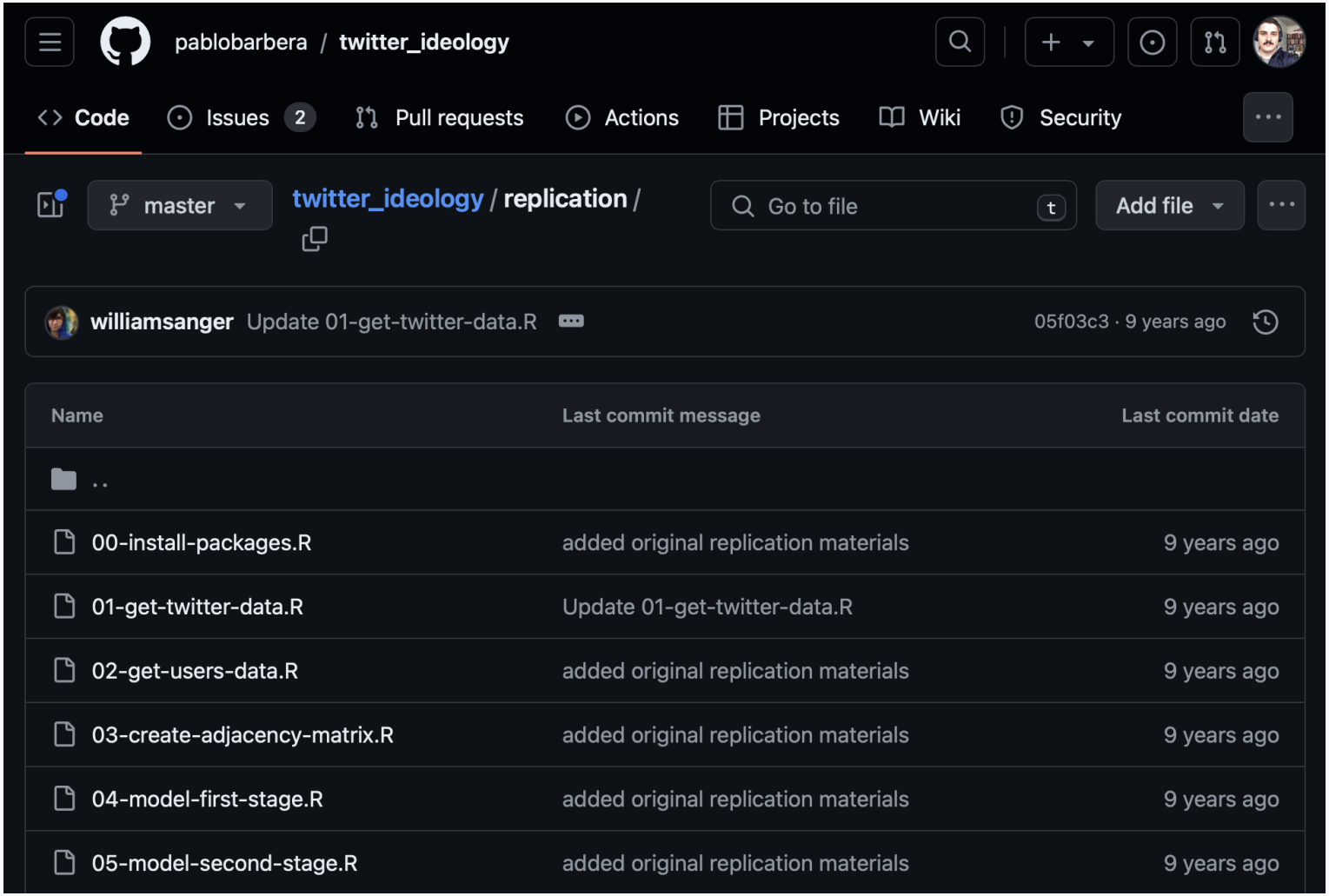
- One project should be in one directory/repository
- Subdirectories should be organized by type of data (code, figures, tables, etc.)
- My favorite approach: Overleaf and Github

Skills to Learn for CSS Research (outside of this course)

- Using LaTeX for word processing (Overleaf.com for collaboration)
- Multiprocessing (how to run code across multiple CPUs)
- Advanced plotting (I use `ggplot2` in `R`, `matplotlib` or `seaborn` in Python are also good)
- Statistical modeling (econometrics, regression, machine learning, quantitative text analysis)

Getting Things Set Up

Example: Replication materials for Barbera (2014)



See Replication Archive on Github

Building on Best Coding Practices

- Learn the official Python style guide (PEP 8)
- Learn how to write clean, modular, and D.R.Y. code (some video explainers on DRY code, and on writing clean code)

References i


Barberá, P., Wang, N., Bonneau, R., Jost, J. T., Nagler, J., Tucker, J., and González-Bailón, S. (2015).
The critical periphery in the growth of social protests.
PloS one, 10(11):e0143611.


Brady, W. J., Wills, J. A., Jost, J. T., Tucker, J. A., and Van Bavel, J. J. (2017).
Emotion shapes the diffusion of moralized content in social networks.
Proceedings of the National Academy of Sciences, 114(28):7313–7318.

References ii


Dietrich, B. J. (2021).
Using motion detection to measure social polarization in the us house of representatives.
Political Analysis, 29(2):250–259.


Farrell, J. (2016).
Corporate funding and ideological polarization about climate change.
Proceedings of the National Academy of Sciences, 113(1):92–97.

References iii

Lai, C., Toriumi, F., and Yoshida, M. (2024).
A multilingual analysis of pro russian misinformation on twitter during the russian invasion of ukraine.
Scientific Reports, 14(1):10155.

Miller, B. and Thompson-Brusstar, M. (2024).
Validating automatic text digitization: How to ensure quality of automatically digitized text data from standard ocr models and large language models.
Unpublished paper draft.