Computational Sociology

Introduction

Dr. Thomas Davidson

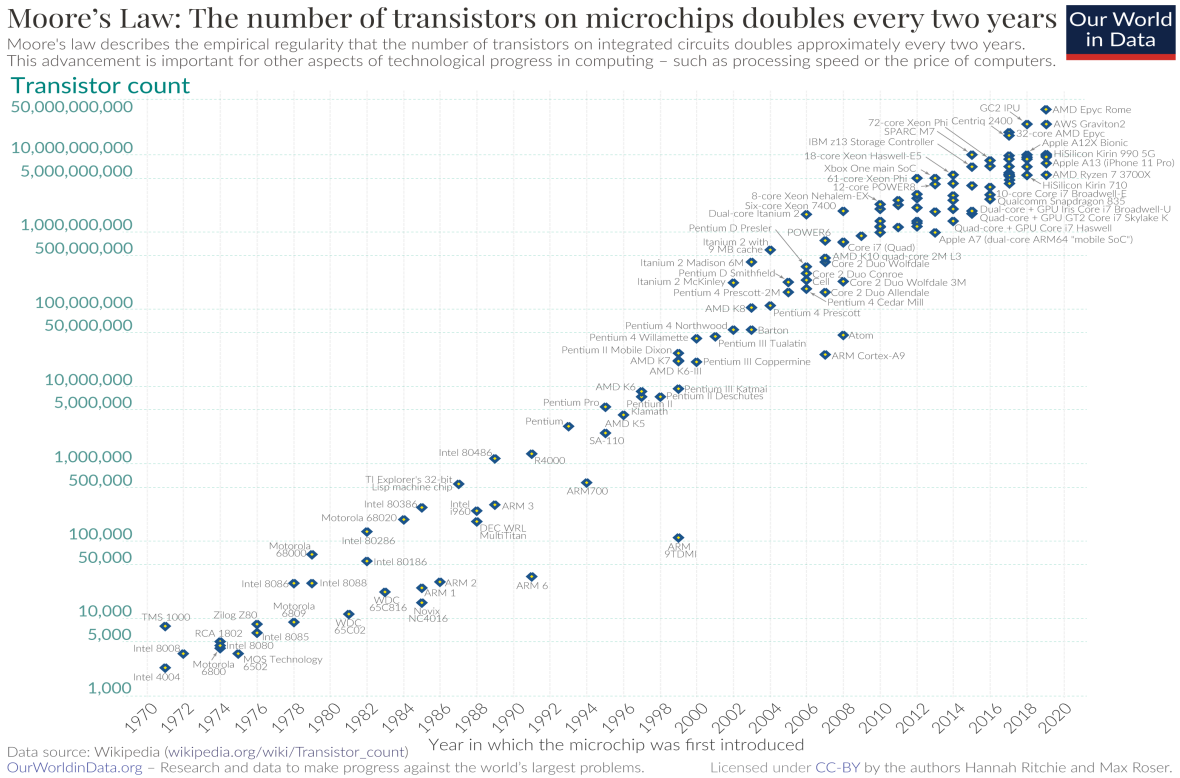
Rutgers University

January 18, 2024

Plan

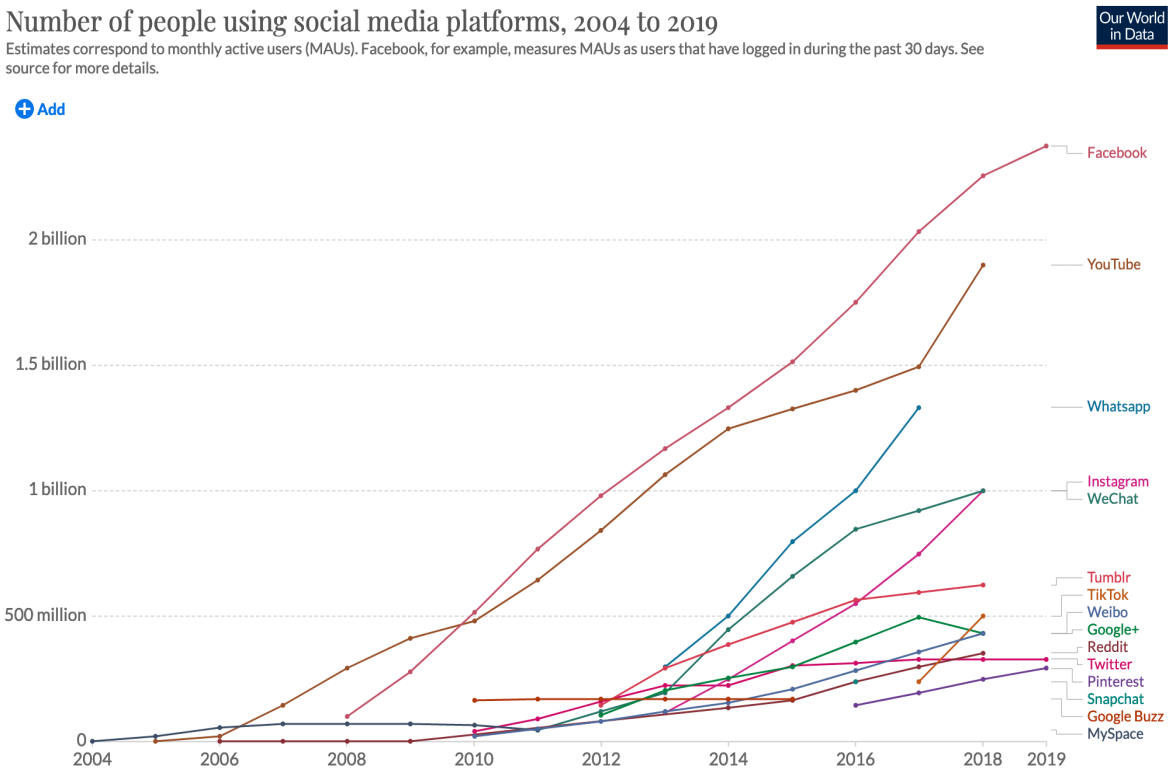
- Part I
  - Introductions
  - A brief introduction to computational sociology
  - Course outline
  - R, RStudio, and Slack
  - ChatGPT and AI policy
- Part II
  - File management and Github

Introduction to Computational Sociology



https://ourworldindata.org/moores-law

Introduction to Computational Sociology



https://ourworldindata.org/grapher/users-by-social-media-platform

Introduction to Computational Sociology

Computational Social Science

The capacity to collect and analyze massive amounts of data has unambiguously transformed such fields as biology and physics. The emergence of such a data-driven "computational social science" has been much slower, largely spearheaded by a few intrepid computer scientists, physicists, and social scientists. If one were to look at the leading disciplinary journals in economics, sociology, and political science, there would be minimal evidence of an emerging computational social science engaged in quantitative modeling of these new kinds of digital traces. However, computational social science is occurring, and on a large scale, in places like Google, Yahoo, and the National Security Agency. Computational social science could easily become the almost exclusive domain of private companies and government agencies. Alternatively, there might emerge a "Dead Sea Scrolls" model, with a privileged set of academic researchers sitting on private data from which they produce papers that cannot be critiqued or replicated. Neither scenario will serve the long-term public interest in the accumulation, verification, and dissemination of knowledge.

Lazer et al. 2009.

Introduction to Computational Sociology

Digital traces and big data

> *[J]ust as the invention of the telescope revolutionized the study of the heavens, so too by rendering the unmeasurable measurable, the technological revolution in mobile, Web, and Internet communications has the potential to revolutionize our understanding of ourselves and how we interact . . . . [T]hree hundred years after Alexander Pope argued that the proper study of mankind should lie not in the heavens but in ourselves, we have finally found our telescope. Let the revolution begin.*
>
> —Duncan Watts (2011, p. 266)

Quoted in Golder and Macy 2014.

Introduction to Computational Sociology

Measuring diurnal rhythms usning Twitter

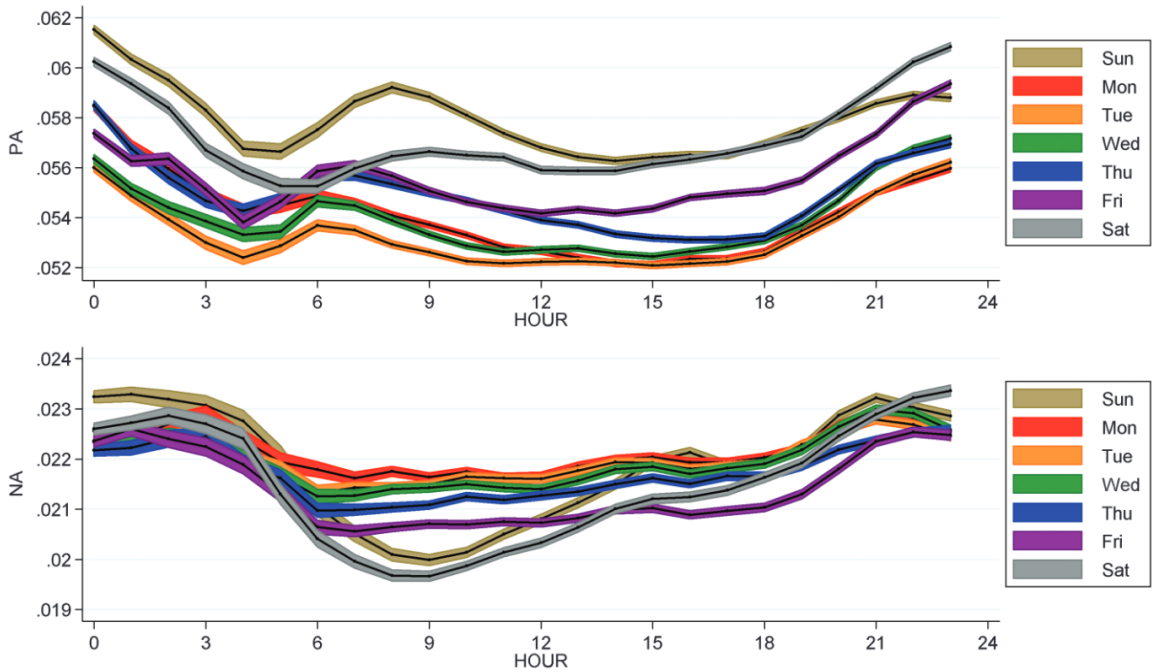


Fig. 1. Hourly changes in individual affect broken down by day of the week (top, PA; bottom, NA). Each series shows mean affect (black lines) and 95% confidence interval (colored regions).

(PA = Positive Affect; NA = Negative Affect. Golder and Macy 2011.)

Readymade data

Readymade

Custommade

Salganik 2017 https://www.bitbybitbook.com/figures/chapter1/bitbybit1-2_readymade-custommade.png

Introduction to Computational Sociology

Multi-modal data and cultural sociology

- Bail (2014) writes that "[S]ocial scientists—and cultural sociologists in particular—have largely ignored the promise of so-called "big data." Instead, cultural sociologists have left this wellspring of information about the arguments, worldviews, or values of hundreds of millions of people from Internet sites and other digitized texts to computer scientists who possess the technological expertise to extract and manage such data but lack the theoretical direction to interpret their meaning."
- CSS tools provide new ways to measure culture and analyze texts, images, and other media previously confined to relatively small-scale, interpretativist methods

Introduction to Computational Sociology

The emergence of a field

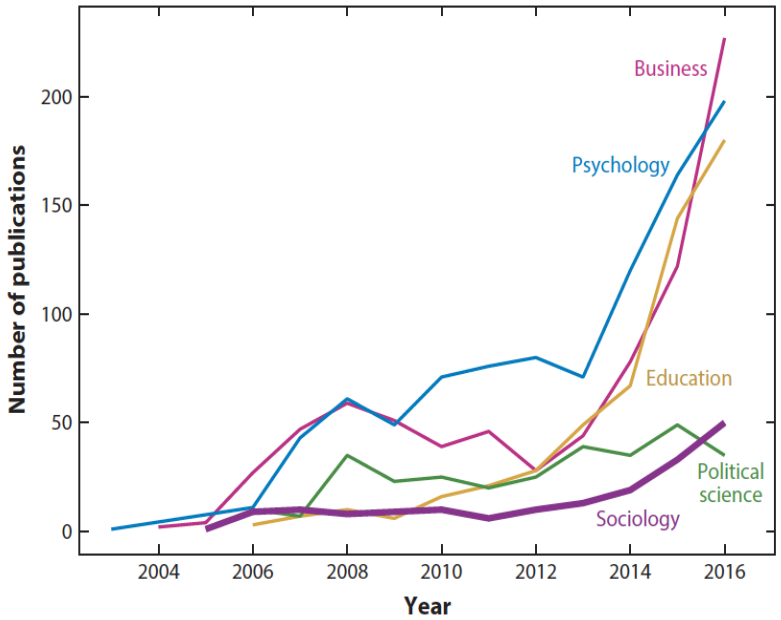


Figure 1

Number of computational social science publications by year—2003–2016—across four scholarly disciplines.

Edelmann et al. 2020

Introduction to Computational Sociology

Some attempts at a definition

1. Use of readymade, multimodal data (image, text, video)
2. New modes of data collection (web-scraping, APIs, online experiments)
3. Application of computational methods developed by computer scientists (topic modeling, word embeddings, deep learning, large language models)

Course outline

Goals

- By the end of this course you should be able to
  - Understand the field of computational sociology and computational social science more broadly
  - Code using R at an intermediate level
  - Understand and implement various computational methodologies for data collection
  - Apply computational methods in your own research
  - Think critically about the use of new data sources and methods

Course outline

Structure

1. Programming in R (Weeks 1-3)
2. Social networks and simulations (Weeks 2-3)
3. Digital data collection (4-6)
4. Natural language processing (7-9, 11-12)
5. Machine learning (10-13)

Course outline

Assessment

- Homework assignments (40%)
  - Programming fundamentals
  - Data collection
  - Natural language processing
  - Machine learning

Course outline

Assessment

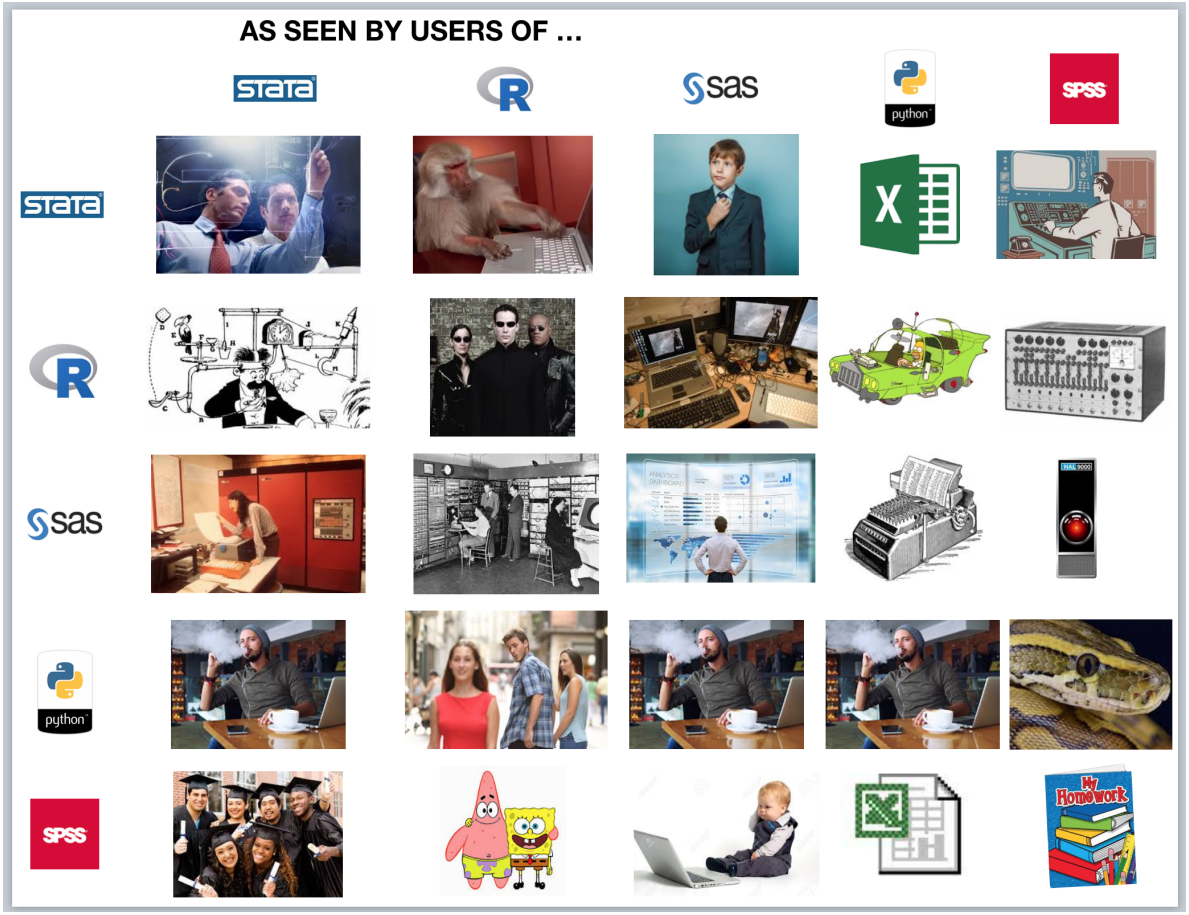
- Final project (60%)
  - Proposal (5%)
  - Data collection and descriptive analysis (5%)
  - Preliminary results (5%)
  - Presentation (5%)
  - Final paper (40%)

Course outline

Policies

- Read the syllabus!
  - Academic integrity
  - Diversity and inclusion
  - Accommodations
  - ChatGPT (more later)
- A note on incompletes:
  - **Please avoid taking an incomplete for this class. It is a bad outcome for everyone involved.**

Why R?



https://kieranhealy.org/blog/archives/2019/02/07/statswars/

Why R?

- Free and open-source
- A statistical programming language
  - Many cutting-edge approaches now implemented in R before Stata
  - Rutgers Sociology has transitioned to R for the grad stats sequence
- Alongside Python, it is one of the main programming languages used by data scientists
- Active developer community
- RStudio

RStudio

Overview

- RStudio is an Integrated Development Environment for programming in R
  - Run code in the console or in scripts
  - View data, objects in memory, plots
  - Create output like papers or slides
  - Integrations including Github and Python interpreter

RMarkdown

Overview

- RMarkdown is an interactive coding environment
  - RMarkdown documents can combine text, LaTeX code, R code, and any output.
  - Write in Markdown or Visual Editor
  - You will be using RMarkdown for your homework assignments and hopefully your papers

Slack

Why Slack?

- Quick and easy communicate
  - Reduces email
  - Shared problem solving
- Code formatting
  - Use “““” to start a code snippet (Slack should auto-complete)
- Emojis and other media
- But please contact me via email for important matters

Other resources

- StackOverflow
  - An online community for coding questions
    - Search for error messages or snippets. In most cases you should be able to find answers to your issues.
    - Sometimes it can take a while to figure out the appropriate query to use to find an answer.
    - If you can't find an answer, you can make your own question - but the formatting requirements are quite strict and users can be unforgiving.
  - A useful thread for posting an R question and example: https://stackoverflow.com/questions/5963269/how-to-make-a-great-r-reproducible-example

Generative AI

- ChatGPT and other models are trained on large amounts of text data as well as code from Github
  - This means they can "understand" R and can write and comment on code
  - And these models have "knowledge" about most of the topics covered during the class
  - Some can even write and run code and produce results!

Generative AI

Two uses of generative AI

- Generative AI as a *research method*
  - Later in the semester, we will discuss these technologies in more detail and how they can advance computational social science
- Generative AI as a *pedagogical tool*
  - These tools can help you learn computational methods, but they also pose challenges to academic integrity

Generative AI as a pedagogical tool

Benefits and drawbacks

- *Benefits*
  - "Bespoke" problem solving reduces need to tailor queries for answers
  - Often correct for simple types of problems
  - Conversational modality can help you to learn
    - e.g. Ask for comments on your code or explanations
- *Drawbacks*
  - Overreliance may hamper ability to learn for yourself
  - Solutions can be incorrect and struggles for more complex problems
  - Less reliable for niche tasks and packages

Generative AI and academic integrity

Policies

- Attempt to solve problems yourself first to avoid overreliance
- *You are not permitted to use ChatGPT to solve homework problems or write code/text for your projects*
  - But you can use these to help you to learn
  - Contact me first if in doubt
- Document usage of these tools in any submissions
  - e.g. "I used ChatGPT to help me to understand how to use the `dplyr` package"
- *You are not permitted to use Github CoPilot in RStudio*
  - Autocompletion violates academic integrity
  - And the solutions are often distracting and not very helpful

Excercise

Problem solving in R

- Download and inspect this code in RStudio (paste it into a new R file)
  - https://raw.githubusercontent.com/t-davidson/computational-sociology/main/2024/course-materials/code/l1_problem_1.R
- Task: Identify and fix the errors

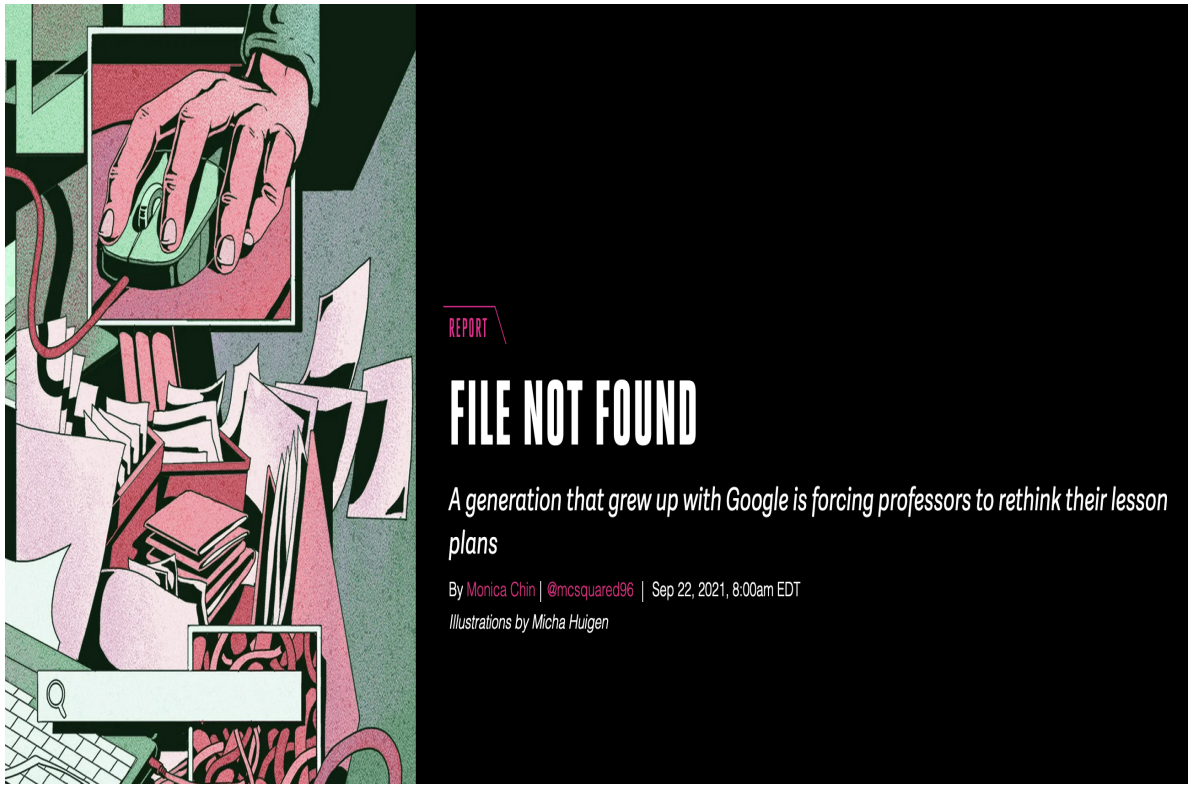
Excercise

Problem solving in R

- Three groups:
  1. Solve it without using the internet
  2. Solve it by searching StackOverflow
  3. Solve it by using ChatGPT

Break

File management



Source: The Verge, 2021.

File management

- File management in this class
  - Keeping track of course materials
  - Working on homework assignments
  - Organizing final projects
  - Reproducibility

Organizing your files

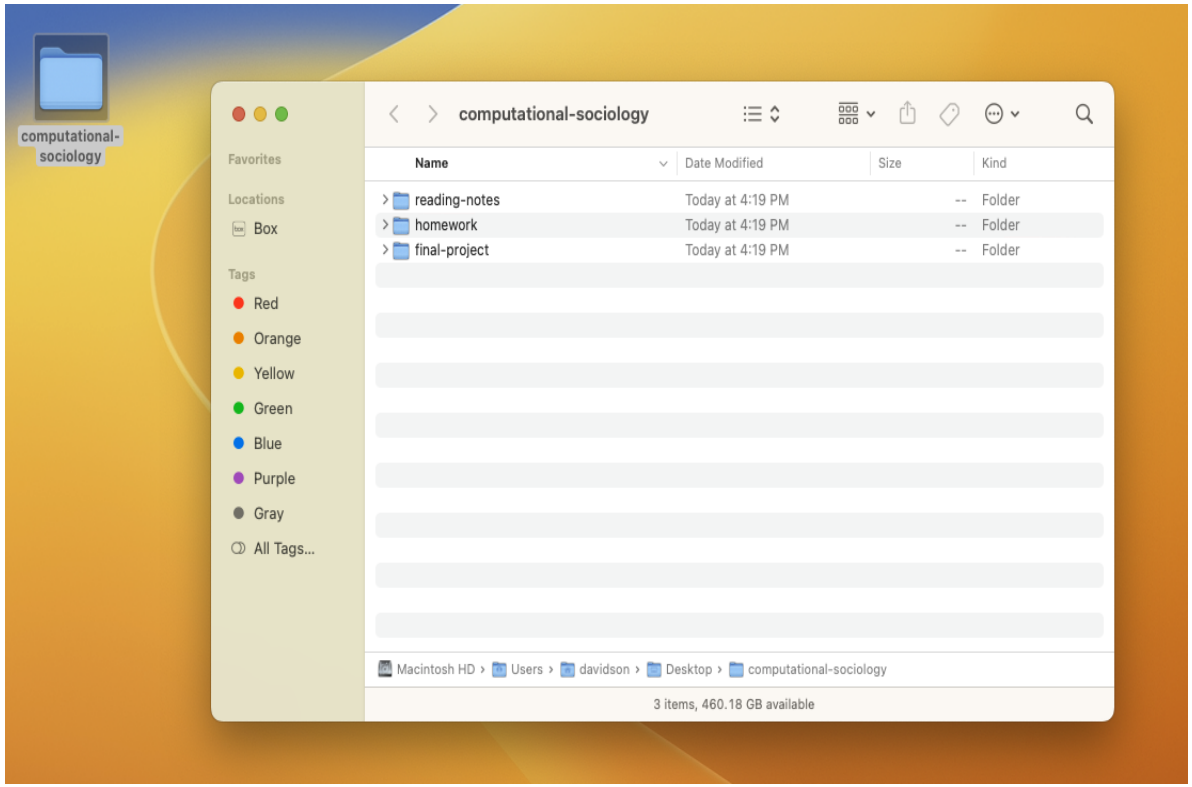
- Make a directory to contain all materials for this class
- Store this somewhere practical
  - e.g, `/Users/me/Documents/computational-sociology`, `/Users/me/Desktop/Classes/computational-sociology`
  - Do not leave files in your `Downloads` directory!

Organizing your files

- Within this directory, create separate directories for the class materials, readings, and homework assignments.
- Do not use spaces in directory names, use hyphens or underscores instead

Organizing your files

It might look something like this once you have added other sub-directories.



Github

- Github is a version-control system
  - This allows you to easily control and manage changes to your code (similar to Track Changes in Word)
  - It can facilitate collaboration
  - Version-control helps to ensure reproducibility
  - It makes it easy to share code
- Github is *not* designed as a place to store large datasets (100Mb file size limit)

Terminology

- A Github *repository* (or *repo* for short) contains all files and associated history
  - A repository can be public or private
  - Files should be organized into directories
  - Github can render Markdown files (suffix `.md` in Markdown), useful for documentation
- Github repositories exist online and you can *clone* them to your local computer

Using Github

- You can interact with Github in several different ways:
  - RStudio integration (recommended)
  - Github Desktop (redundant if using RStudio)
  - Through your browser (not recommended, but viewing is fine)
  - Using the command line (recommended for advanced users)

Using Github

Follow the instructions on the course website to set up Github with RStudio: https://github.com/t-davidson/computational-sociology/blob/main/2024/course-materials/github_setup.md

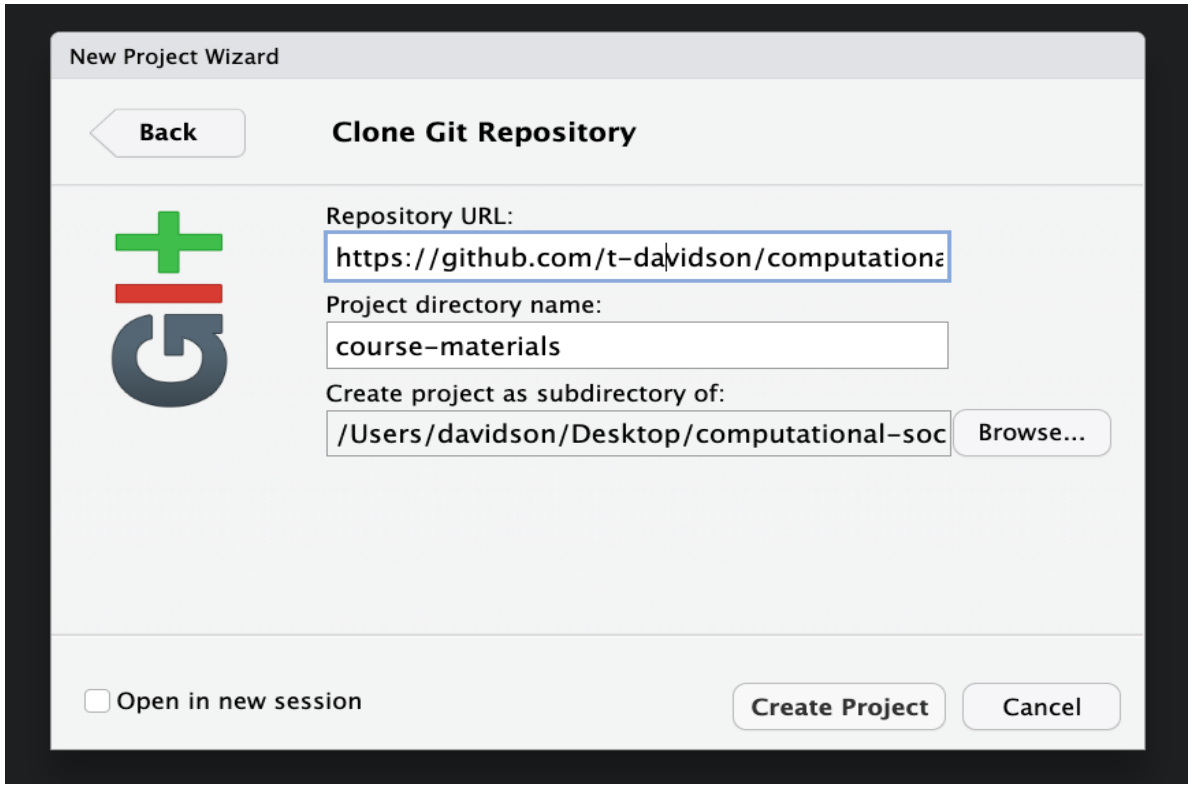
1. Register for a Github account
2. Install Git
3. Sync your Github account with RStudio

Cloning the course website

- Once you have this set up, navigate to the course website on Github and copy the URL
  - https://github.com/t-davidson/computational-sociology

Cloning the course materials

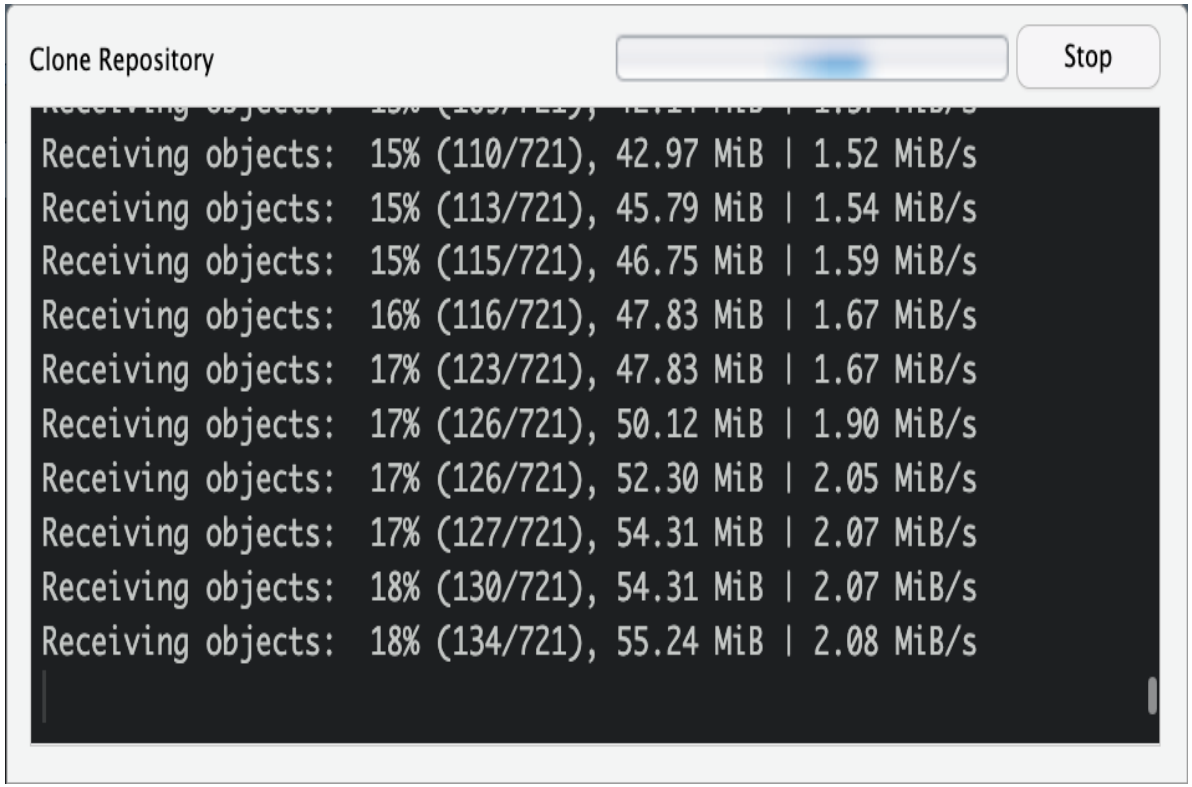
- In RStudio, click `File > New Project > Version Control > Git`



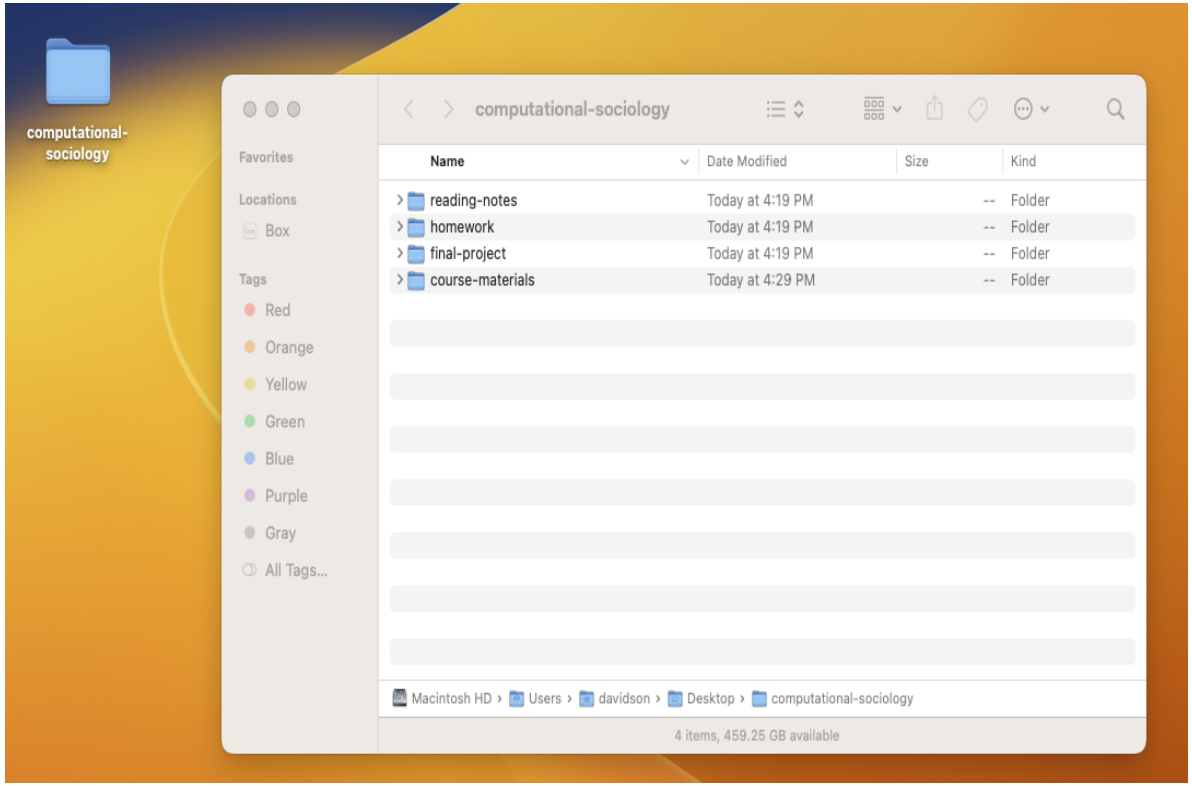
Cloning the course materials

- Paste the URL into the `Repository URL` field
- Write a suitable name in the `Project directory name` field
  - This will be the name of the folder that is created on your computer
- Choose a location to store the repository on your computer
  - Recommend using the folder we created earlier
- Then click `Create Project`

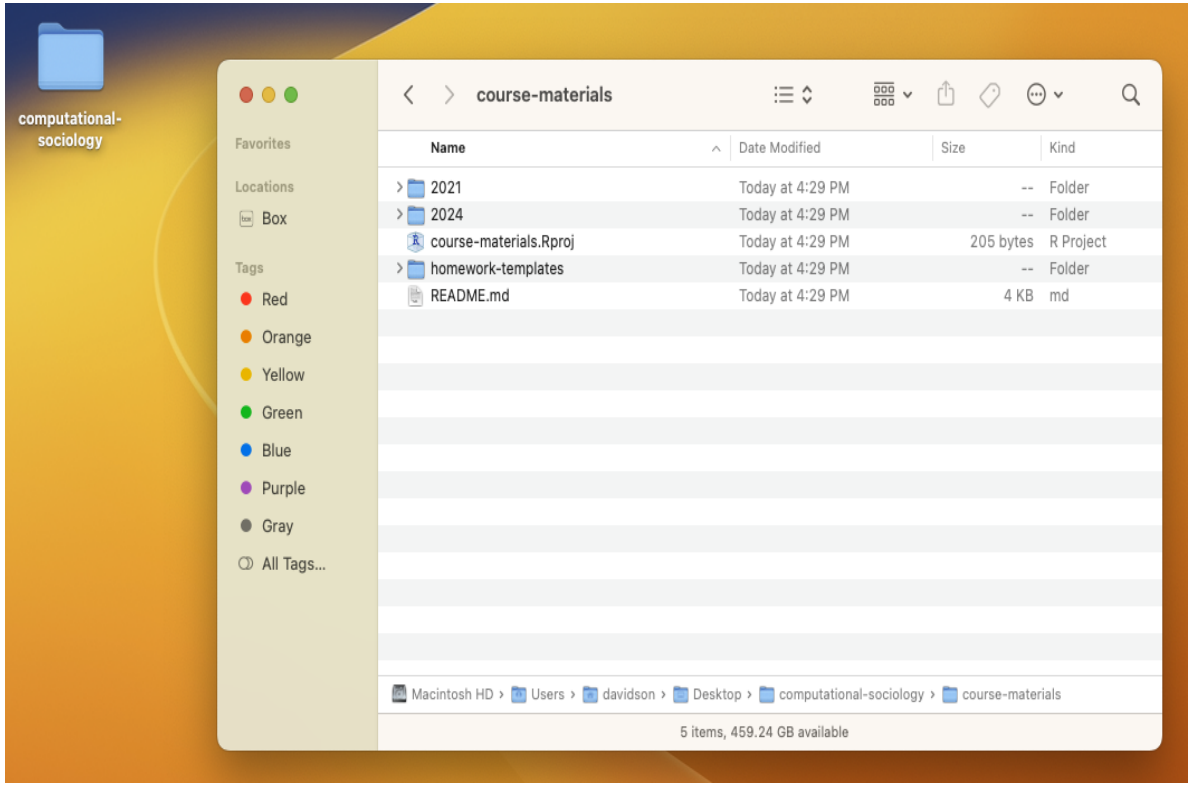
Cloning the course materials



Cloning the course materials

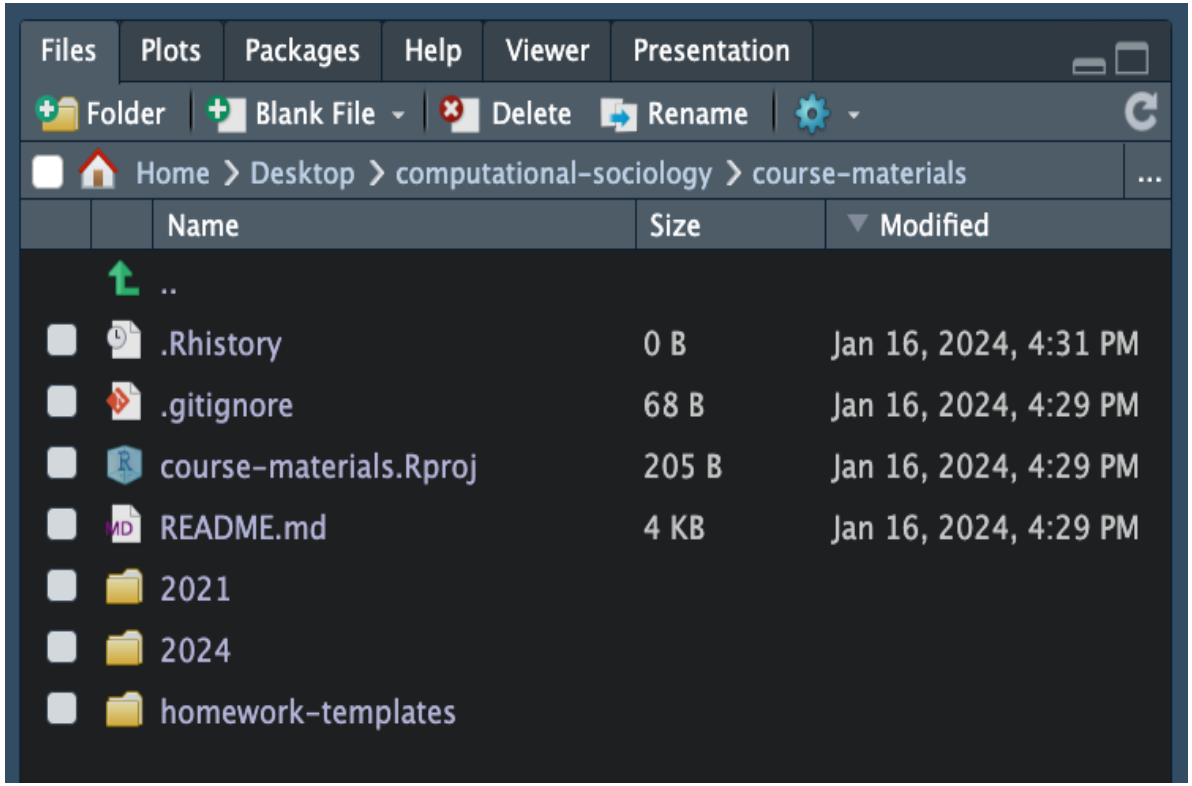


Cloning the course materials



Navigating files using RStudio

You can also use the `Files` pane in RStudio to navigate folders.



Opening this .Rmd file

- Navigate to `2024/course-materials/slides/` and open `lecture1-introduction.Rmd` by double-clicking the file

Navigating files using RStudio

Run the `getwd()` command to show the current working directory in R.

```
getwd()
```

Navigating files using RStudio

Run the `setwd()` command to change your working directory. Try going one step back from the current directory using `..` You can then run `getwd()` to verify it has changed.

```
setwd("..")
getwd()
```

Navigating files using RStudio

- When a `.Rmd` file is opened RStudio defaults to the directory where the file is contained.
  - If you run `setwd()` it will change within a chunk, but other chunks will revert back to the working directory.

Navigating files using RStudio

The working directory is important when considering loading files. Navigate to the `code` directory. You can use the `list.files` function to see a list of the files in the current directory.

```
setwd("../code/")
list.files()
```

Using file paths

- The working directory is critical because it defines a path we need to use to load files. Different files will fall into one of three groups:
  1. File contained in the working directory.
  2. Files contained in outer directories.
  3. Files contained in inner, nested directories.

Files contained in the working directory

If a file exists in the working directory, you will see it when running `list.files`. It can be loaded by using the file name.

```
library("tidyverse")
read_file("file.txt") %>%
    print()
```

Files contained in an outer directory

If a file is contained in an upper level directory, you need to use the `..` to escape the current directory. For each step out from the current directory you need to add another `/..` to the file path. In this case, the file is contained one level above the current directory.

```
read_file("../file_outer.txt") %>%
    print()
```

Files contained in an inner directory

If a file is contained in an inner directory, you need to add the name of the directory to the file path.

```
read_file("nested/file_inner.txt") %>%
    print()
```

Files contained in an inner directory

If a file is contained in an inner directory, you need to add the name of the directory to the file path.

```
read_file("nested/nested2/file_inner_inner.txt") %>%
    print()
```

Excercise

Uncomment the code and omplete the path to print the contents of the file hiding in the 2021 directory.

```
read_file("") %>%
    print()
```

File navigation

Common errors

- It is common to get errors if you misspecify a path when trying to load a file. You will get an error like the following:
  - `Error: 'filename' does not exist in the current working directory ('directory').`
- If this happens, check whether the file is in your current working directory.
  - You can always use your `Files` tab or your normal file viewer to verify the location.

Next week

- Substantive topic
  - Social networks and social network analysis
- Technical topic
  - Data structures in R
    - Vectors, matrices, lists, dataframes