

Introduction & Syllabus

POS6933: Computational Social Science

Jake S. Truscott, Ph.D

University of Florida
Spring 2026



Overview

- Course Introduction & Overview
- Readings, Grading, Evaluation, and Expectations
- Introductions (Get to Know You!)
- Break!
- Downloading R and Python
- File (Data) Storage (Management and Reproducibility)

Computational Social Science

- Computational Social Science is an incredibly broad sub-field within scope of political science. Particularly useful for concepts including (but not limited to):
 - **Text as Data:** Analyses of legislative speeches, party platforms, and court opinions to measure ideology, framing, and rhetorical strategies.
 - **Social Media & Online Behavior:** Mined to study political communication, polarization, misinformation, and mobilization.
 - **Networks & Relationships:** Mapping legislative coalitions, donor connections, and influence in lobbying or policymaking.
 - **Simulation & Modeling:** Agent-based and computational models simulate collective action, polarization, and institutional processes.
 - **Elections & Campaigns:** Modeling to help forecast elections.

Computational Social Science (Cont.)

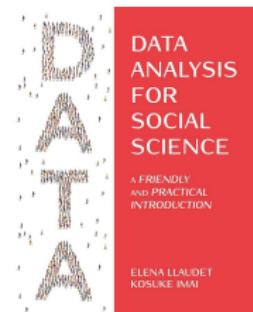
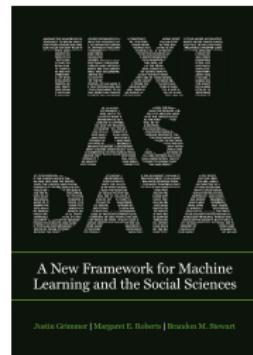
- We don't have time to cover every layer of methodology or substantive concept (*nor would I feel qualified...*)
- Instead, we are largely going to focus on two goals for this semester:
 - ① Improve your proficiency and competency with programming in R and (to a lesser extent) Python
 - ② Introduce methodologies primarily in the realm of Text as Data – with an eye towards both conceptual knowledge and application.

Computational Social Science (Cont.)

- By the end of the semester, you should be able to:
 - Conduct advanced computing routines, including multi-layered and hierarchical coding structures (e.g., iterative loops, data organization and manipulation, functions, package construction, etc.)
 - Retrieve, process, and organize non-traditional data sources (e.g., text) for classification tasks using a combination of natural language processes (e.g., supervised and unsupervised learning models)
 - Create informative representations of data and other summary findings (e.g., tables and figures)
 - Produce replicable coding routines using compilation suites in R, Python, and LaTex (Beamer)

Materials and Evaluation – Textbooks and Readings

- **Required:** Grimmer, J., Roberts, M. E., & Stewart, B. M. (2022). *Text as data: A new framework for machine learning and the social sciences*. Princeton University Press.
- **Recommended:** Llaudet, E., & Imai, K. (2023). *Data analysis for social science: A friendly and practical introduction*. Princeton University Press.
- Articles Posted to Canvas
 - Have all readings complete before coming to class!



Materials and Evaluation – Grading & Evaluation

Weekly Problem Sets	40%
Participation	10%
Final Project & Presentation	50%

Materials and Evaluation – Evaluation (Problem Sets)

- Problem sets due to Canvas on Sundays by 11:59pm
- I will drop the two submissions with lowest marks
- Must be compiled using RMarkdown – We will discuss file organization today
- We will spend the first 30min-1hr of each class period reviewing the problem set from the week before.

Materials and Evaluation – Evaluation (Final Project)

- Topic Selection Due Sunday (2/8)
- Submission needs to emphasize some element(s) of course material
- Instructor assessment & peer review

Materials and Evaluation – Expectations

Expectations are fairly simple:

- ① Come to Class Prepared & Ready to Learn
- ② Don't Be Afraid to Ask for Help – We have a **ton** of flexibility
- ③ Embrace the Learning Curve

Materials and Evaluation – Other Notes

- The course is technology driven – You need to come prepared with a laptop capable of downloading R, RStudio, and Python
- I am maintaining a GitHub Repository and website for the course: https://jaketruscott.github.io/CSS_POS_UF/
- The website has many additional resources for each week of the semester.

About Me – Jake S. Truscott, Ph.D.

- Research Focus: American Judiciary & CSS
- Proud UGA Grad
- Fav Conspiracy Theory: The New York Mets are a Though Experiment Created by Harvard University to Study the Emotional Pain Tolerance of American Sports Fans
- My Wife is Very Pregnant – We may need to skip a week last minute in late-February/early-March (TBD)



About You

- Name
- Research Focus or Major
- Fun Fact or Hot Take

Downloading R

Downloading R

Downloading R



- R is a programming language and environment for statistical computing and data analysis.
- It is designed specifically for working with data, running statistical tests, creating models, and visualizing results.
- **Why R and not Stata or Python (exclusively)?**
 - R is incredibly flexible, intuitive, easily accessible, and free!

Downloading R (Cont.)



- We will use R for just about everything in this course.
- Even when we shift to tools better suited for Python, I will show you how can implement through R
- To download Python, visit: https://jaketruuscott.github.io/CSS_POS_UF/class_1/downloading_Python.html

Downloading Python



Downloading Python



Downloading Python



- Python is a versatile, general-purpose programming language – it can be used for web development, data analysis, automation, machine learning, and more.
- Many of you may already have exposure to Python from Dr. Arfi's classes – You are welcome to use more Python if you so choose, though I will be teaching R primarily.
- To download R and its companion IDE RStudio, visit:
https://jaketruuscott.github.io/CSS_POS_UF/class_1/downloading_Python.html

Data Storage and Organization

- It is imperative that you put yourself in the best position to succeed.
- Part of that entails a system for organizing your data and R/Python scripts.
- I am going to require everything you complete for this course – both during our classroom meetings and as homework – be completed using RMarkdown.

Data Storage and Organization (Cont.)

- We are going to create a directory (folder) to house all of your data, code, and other materials.
- R is able to create localized project directories and provides the ability to load or export from a relative path – i.e., you do not need to include the full path to a file.
- Ex: Assuming R Project is Located in
"C:/Users/JakeTruscott/Folder"

```
load("C:/Users/JakeTruscott/Folder/SubFolder/File.rdata")
# Full Path

load("SubFolder/File.rdata")
# Relative Path
```

Data Storage and Organization (Cont.)

- ① Create a new folder – somewhere obvious and unlikely to be accidentally deleted.
- ② Open RStudio
- ③ Navigate to File → New Project...
- ④ Select Existing Directory
- ⑤ Navigate to and Select the folder you created in Step 1
- ⑥ Check Open in New Session and Select Open

Data Storage and Organization (Cont.)

- Navigate to the new repository folder on your local machine
- Create a `data`, `code`, and `practice_set` folders
- The purpose of this is simple: Promote reproducibility through relative paths.
- Now all of the work we do in class and through problem sets will be easy to troubleshoot – we won't need to mess around with the location of data or files!

Next Class

- Next class will be a short introduction to Intermediate Programming in R
- Will focus on:
 - Random Number and Distribution Generation
 - Loops and Iterative Programming
- Before Next Class:
 - Visit the course Github – Review Reading Materials
 - Secure copy of Text as Data (GRS) – Llaudet and Imai text only necessary if not fully comfortable with R's basic syntax.