Syllabus for SODA 501-002

Big Social Data: Approaches and Issues (Penn State)
Spring 2026 (Regular Session)

Instruction Mode: In Person
Campus/Location: 124 Pond
Meeting Time: Thursdays, 12:00–3:00 pm
Meeting Dates (session window): 01/12/2026 – 05/01/2026
First/Last Thursday meeting: 01/15/2026 – 04/30/2026
Credits: 3

Instructor: Jared Edgerton
Email: jfe4@psu.edu
Office: Carpenter 513
Office Hours: Tuesdays 12:00 to 4:00 and by appointment

Administrative note: If you are attempting to enroll in **SODA 502**, please contact Mady Forsythe (mjf6218@psu.edu).

Course Description

SODA 501 introduces core approaches and issues in **big social data**, emphasizing the tools and practices by which social data are created, collected, curated, linked, documented, and shared. "Social" data include data about (or arising from) human interactions, including text, network, spatial, audio/video, sensor, and platform-generated data, often at scales or levels of complexity that strain conventional social-science workflows.

Course format. Each class meeting follows a consistent structure (with a short break/transition built in):

1. **Research talk (60 min):** an outside speaker presents their research and workflow.
2. **Coding lab (80 min):** a designated student **Lab Lead** runs an instructor-prepared coding demonstration and guided exercises.
3. **Paper discussion (40 min):** we discuss a social-science article that uses the week's tool/method.

Pre-class video. For each week, students must watch a **20–40 minute video** (posted on Canvas) before class. The video introduces concepts and sets up the coding lab.

Student Learning Objectives

By the end of the course, students will be able to:

1. Identify major sources and modalities of big social data (text, network, spatial, digital trace, administrative).

2. Build end-to-end data pipelines: acquisition → storage → cleaning → linkage → documentation.

3. Implement core tooling in **Python or R** (plus SQL and Git) for scalable data work.

4. Evaluate social-science research that uses big social data, with attention to validity, ethics, and reproducibility.

5. Communicate across disciplines by engaging research speakers and translating methods across domains.

Required Tools

- A laptop capable of running Python and/or R.
- Git + a GitHub account.
- Access to Penn State Canvas for videos, announcements, and submissions.
- Optional (helpful): Docker/Podman, a cloud account (AWS/GCP/Azure), or PSU computing resources (if available).

Tooling Choice (Python vs R) and Repository Standards

Students may complete coding work in **Python or R**. You may switch languages during the semester, but each assignment must be fully reproducible in the language you choose for that assignment.

Baseline requirements for all submissions:

- A clean, reproducible project structure (at minimum: `/data_raw`, `/data_processed`, `/src`, `/docs`).
- A short `README` describing: (i) the data source and access method, (ii) how to run the code end-to-end, and (iii) what outputs are produced.
- Clear provenance notes: where the data came from, when it was collected, and any known limitations.

Recommended defaults:

- Python: `uv` or `conda` environment + `requirements.txt` (or `pyproject.toml`).
- R: `renv` lockfile.
- Optional (best practice): containerize with Docker/Podman for "one command" reproduction.

Evaluation (Grading)

Your final grade will be based on:

- **Participation and in-class engagement (10%)**
  Includes Q&A with speakers, active contribution during coding lab, and substantive participation in article discussion.
- **Weekly coding labs / assignments (30%)**
  Short deliverables demonstrating the week's tool (code + brief README).

- **Student-led coding demo (20%)**
  Each student will serve as **Lab Lead** for at least one class session (more if enrollment requires). The Lab Lead will:

  1. Meet with the instructor **before class** (typically 30–60 minutes, scheduled during the prior week) to review the instructor-prepared demo code.
  2. Rehearse the live walkthrough and confirm the demo runs on a clean environment.
  3. Lead the in-class coding demonstration and handle questions, with the instructor as backup.
  4. Submit a short post-class note (bullet points) describing what worked, what broke, and how it was fixed.

  Lab Lead assignments will be posted after Week 1 on Canvas.
- **Final project (15%)**
  presentation (15%), final repo + writeup (25%).

Letter grades. I will use the following thresholds: A (93–100), A– (90–92.9), B+ (87–89.9), B (83–86.9), B– (80–82.9), C+ (77–79.9), C (70–76.9), D (60–69.9), F (¡60).

Academic Integrity

Penn State expects students to complete all coursework in accordance with course-specific rules and university policy on academic integrity. Unless explicitly permitted, you must not submit work that is not your own, and you must not assist others in ways that violate course rules. When in doubt, ask before submitting.

Students with Disabilities

Penn State welcomes students with disabilities into the University's educational programs. If you anticipate needing accommodations, contact Student Disability Resources (SDR) early in the semester and provide the instructor with the approved accommodation letter as soon as it is available.

Course Norms

Be professional and collaborative. This course is designed to be challenging. The goal is to build durable, transferable workflows—not just to "get it to run."

Course Schedule (Spring 2026 — Thursdays)

Key dates: Spring Break week is Mar 8–14 (no class Thu Mar 12).
Readings/videos: Posted on Canvas; the "application paper(s)" is discussed during the last 30 minutes.

Week 1: January 15, 2026 — Course Overview & What is "Big Social Data"?

- **Coding lab:** Course toolchain setup (Python/R environment, Git/GitHub, repo structure).

- **Application paper:** Lazer et al. (2009) "Computational Social Science" (Science) *or* instructor-selected modern replication.
- **Pre-class video:** Workflow overview + reproducible project scaffolding.

Week 2: January 22, 2026 — The Web as Data (Scraping & HTML)

- **Coding lab:** Scraping with `BeautifulSoup/requests` (Python) or `rvest` (R); respectful scraping; pagination.
- **Application paper:** Brown et al. (2025) Big Data and Society.
- **Application paper:** Mahdavi, P. (2017) PSRM
- **Pre-class video:** Web scrapping overview.

Week 3: January 29, 2026 — APIs, Rate Limits, and Data Collection at Scale

- **Coding lab:** API auth; pagination; retries; logging; storing raw JSON.
- **Application paper:** Bail et al. (2018) PNAS
- **Application paper:** Barbeta, P. (2015) Political Analysis
- **Pre-class video:** API design patterns + data provenance.

Week 4: February 5, 2026 — Surveys, Platforms, and Crowdsourcing

- **Coding lab:** Survey exports; cleaning; codebooks; labeling; quality checks.
- **Application paper:** Berinsky et al. (2012) Political Analysis
- **Application paper:** Bisbee (2024) Political Analysis
- **Pre-class video:** Data quality, missingness, and measurement in platform data.

Week 5: February 12, 2026 — Databases & SQL for Social Data

- **Coding lab:** SQL basics; schema design; indexes; joins; using `duckdb/sqlite`; query from Python/R.
- **Application paper:** Baumgartner et al. (2020) Proceedings of the international AAAI conference on web and social media.
- **Application paper:** Green et al. (2025) APSR.
- **Pre-class video:** Relational thinking + query optimization basics.

Week 6: February 19, 2026 — Record Linkage and Entity Resolution

- **Coding lab:** String distances; blocking; probabilistic linkage; evaluation.
- **Application paper:** Ornstein, J. (2025) Political Analysis
- **Application paper:** Nyseth Nzitatira et al. (2024) ASR
- **Application paper:** Nyseth Nzitatira et al. (2022) JPR (optional)
- **Pre-class video:** Linkage as measurement + error tradeoffs.
- **Milestone:** Final project proposal due.

Week 7: February 26, 2026 — Text as Data Pipelines

- **Coding lab:** Tokenization; embeddings; topic models; basic transformer workflow (lightweight).

- **Application paper:** Alsarra et al. (2025) Computational and Mathematical Organization Theory
- **Application paper:** Burnham et al. (2025) Political Analysis
- **Pre-class video:** Practical text pipeline architecture.

Week 8: March 5, 2026 — Network Data & Graph Workflows

- **Coding lab:** Graph construction; centrality; community detection; scalability tips.
- **Application paper:** Minhas, S & P. Hoff (2025) Political Analysis
- **Application paper:** Olivella et al. (2022) JAS
- **Pre-class video:** Network representation choices and pitfalls.

March 12, 2026 — *No Class (Spring Break Week)*

Week 9: March 19, 2026 — Spatial Data, GIS, and Remote Sensing

- **Coding lab:** `sf/GeoPandas`; joins; rasters; mapping; basic spatial features.
- **Application paper:** Harari, M. (2020) AER
- **Application paper:** Panagopoulos et al. (2023) Land
- **Pre-class video:** Spatial data formats + coordinate systems.

Week 10: March 26, 2026 — Reproducibility: Git, Containers, and Research Artifacts

- **Coding lab:** Git branching; pull requests; Docker/containers; environment capture; data documentation.
- **Application paper:** Heath, Davidson, et al. (2023) The Journal of Finance
- **Application paper:** Holzmeister,et al. (2025) Nature Human Behaviour
- **Pre-class video:** Reproducible pipelines end-to-end.

Week 11: April 2, 2026 — Data Quality, Labeling, and Validation

- **Coding lab:** Validation checks; inter-coder reliability; basic monitoring; documenting limitations.
- **Application paper:** Lazer et al. (2014) Science
- **Application paper:** Parkinson, S. (2024) APSR
- **Application paper:** Sales et al. (2022) Applied Geography
- **Pre-class video:** Validity threats in big social data.

Week 12: April 9, 2026 — LLM-Assisted Data Extraction (Human-in-the-Loop)

- **Coding lab:** Structured extraction from messy text using LLMs; prompt design for schemas; batch processing; uncertainty checks; human validation/spot-audits; documenting failure modes.
- **Application paper:** Gilardi et al. (2023) PNAS
- **Application paper:** Heseltine, M. & Bernhard Clemm von Hohenberg. (2024) Research & Politics
- **Pre-class video:** LLM extraction patterns + evaluation (precision/recall, auditing).

Week 13: April 16, 2026 — Quantum Computing Week (Intro + Social-Science Application)

- **Coding lab:** Quantum basics in Python (e.g., circuits + simulation with Qiskit; what "speedup" means).
- **Application paper:** Moloney, K & S. Al-Kuwari (2025) Science and Public Policy
- **Application paper:** Orus et al. (2019) Reviews in Physics
- **Pre-class video:** Qubits, circuits, and what quantum does/do not change in practice.

Week 14: April 23, 2026 — Student Project Presentations I (Data Pipelines)

- **Deliverable:** 20 minute presentation + demo of repo/pipeline.

Week 15: April 30, 2026 — Student Project Presentations II & Wrap-Up

- **Deliverable:** 20 minute presentation + demo of repo/pipeline.

Final Project

Goal: Build a reproducible pipeline that produces (or substantially augments) a "big social data" asset relevant to your research interests (text, network, spatial, platform trace, administrative, etc.).

Minimum components:

- You can coauthor the paper.
- A clearly documented data source (provenance) and collection strategy.
- A reproducible pipeline (scripts/notebooks + README) that runs end-to-end.
- Basic validation checks (at least two) and a short writeup describing limitations.
- A shareable research artifact: a GitHub repo with instructions to reproduce outputs.

Milestones:

- Weeks 15–16: Presentation.
- **Final submission:** Due **Friday, May 8, 2026**, 5:00 pm ET (end of finals week).