Computational & AI-Assisted Methods for Social Sciences

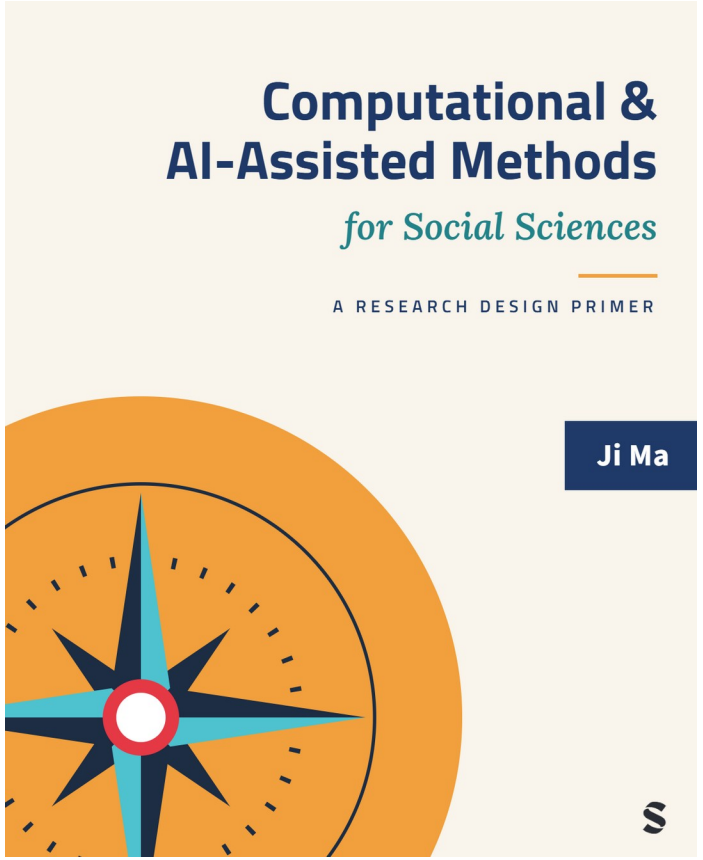
A Research Design Primer

Chapter 10

Communicating Your Pipeline: Tools, Tips, and Exercises

Ji Ma

cssprimer.org



Learning objectives

By the end of this chapter, you can:

- Draft an adapted preregistration for an exploratory computational analysis
- Freeze a computational environment so a peer can rerun your analysis
- Assemble a Rules/Memory/Knowledge documentation package for AI-assisted coding
- Design honest figures with the 3-step framework and Kirk's design process
- Build an integrative artifact that carries its uncertainty with it

One cumulative artifact, three properties

- Your corpus, your coded records, your network and prediction outputs
- §10.1 Transparency: preregistration and a frozen environment
- §10.2 AI documentation: a package an auditor can read
- §10.3 Interpretability: from typed dataframe to uncertainty-aware artifact

Takeaway: The chapter runs one project through inspectable, reproducible, and interpretable in turn.

§10.1

Transparency in practice

Inspectability lets others see the steps; reproducibility lets them rerun the steps.

Register the procedure, not the prediction

§10.1

- Exploratory work can still lock corpus boundaries, preprocessing, and exclusion rules
- Data-dependent choices get a decision procedure: “select *k* by held-out coherence”
- Label every analysis confirmatory or exploratory; unlabeled mixing is the inflation channel

Takeaway: The goal is a useful preregistration, not a performative one.

Exercise 10.1: Adapted preregistration for an exploratory analysis

- Choose one Chapter 8 analysis and a platform (OSF or AsPredicted)
- Lock down what you can: data source, exclusions, outcome, validation
- Write decision procedures for data-dependent choices
- Label each planned analysis confirmatory or exploratory
- Reflect: would a registered report have been feasible?

You will need

- One completed analysis from Chapter 8, with its script
- An account on OSF or AsPredicted

Deliverable

The timestamped preregistration, plus a half-page note listing what you fixed in advance, what you specified as procedure, and what remains exploratory.

Why “the analysis ran” is not “the analysis runs”

- A NetworkX update can change the Louvain partition next year
- A collaborator's older igraph can change it today
- Pinned versions plus recorded seeds close both gaps

Takeaway: Reproducibility is a property of the environment, not just the script.

Exercise 10.2: Freeze your computational environment

- Inventory dependencies: `pip freeze` or `renv::snapshot()`
- Generate a lock file with exact pins, never `>=`
- Seed stochastic steps; run at least three seeds and report the spread
- Archive the OpenAlex snapshot as a dated, read-only file
- Place yourself on Gundersen's R1–R4 scale

You will need

- The analysis script from Exercise 10.1
- The environment it currently runs in
- A version-control repository

Deliverable

A repository a peer can clone and run: pinned environment, seeded script, and a README stating your Gundersen rung and what the next rung requires.

Gundersen's rungs: what did you actually share?

- R1 Description: a methods section the reader must translate into code
- R2 adds the code; R3 adds the data
- R4 Experiment: description, code, and data as one package
- Freezing the environment is what lets an R4 package actually rerun

Takeaway: Most projects sit at R2; the missing rung is usually the data snapshot.

§10.2

Documenting the AI-assisted workflow

When an LLM did part of the analysis, code and data alone cannot carry the method.

“Used GPT-4 with a prompt” is unauditable

§10.2

- The prompt, model version, and iteration history are methodological choices
- End state: promptbook under `kb/codebooks/`, promptlog under `memory/`, discipline in the rules file
- Each artifact belongs to exactly one layer

Takeaway: The promptbook is to an LLM what a codebook is to a human coder.

Exercise 10.3: Build a Rules/Memory/Knowledge package

- Knowledge: save the complete promptbook, with model ID and parameters
- Memory: reconstruct the promptlog and dated run notes
- Rules: codify logging and validation discipline in the project rules file
- Document validation: sample, agreement metric, subgroups, failures
- Hand the package to a peer and have them regenerate a sample

You will need

- Prompts, model settings, and outputs from your Chapter 6 coding
- Whatever record you kept of how they changed
- A peer willing to try reproducing a sample

Deliverable

The three-layer package plus the validation report and a note on what your peer could not reproduce.

↗ **Chapter 2 templates** · cssprimer.org/ch02/

§10.3

Designing for interpretability

From a typed dataframe to a figure, an artifact, and an uncertainty layer.

The chart should be the last place a problem appears

- Visualization is where earlier mistakes become legible: wrong unit, missing dimension, mismatched type
- The frameworks are checkpoints at different scales, not alternatives
- Each checkpoint hands a defined artifact to the next

Frameworks as checkpoints along one pipeline

Stage	Framework	Input → output
Format the data	Tidy data (Ch. 3)	raw records → rectangular tidy table
Bridge data to design	3-step viz-making (§10.3)	tidy table + question → typed dataframe + honest encodings
Run the design cycle	Kirk's four stages (Ch. 9)	finding + audience → finished figure
Evaluate the figure	Trustworthy / accessible / elegant (Ch. 9)	candidate → refined figure
Quantify uncertainty	Statistical / model / algorithmic (§10.3)	refined figure + reruns → uncertainty-aware figure
Frame the pipeline	Inspectable / reproducible / interpretable (Ch. 9)	workflow + figure → scientific communication

Takeaway: One artifact travels the whole table: raw records to communicable science.

Step 1: what is the unit of analysis?

- One mark on the chart represents one unit: author, paper, dyad, document, model run
- “One dot = one author” and “one dot = one paper” are different visualizations
- Mislabeling paper counts as author rates makes prolific authors look like a field trend

Steps 2 and 3: dimensions and their data types

- Every visual encoding consumes one dimension; count both before sketching
- A scatterplot with color and size spends four; small multiples add a fifth
- Classify each dimension: nominal, ordinal, continuous, relational, or textual

Data types constrain honest visual elements

Data type	Honest visual elements	Do not
Nominal	Distinct colors, shapes, or panels	Order on a continuous axis
Ordinal	Small sequential palette; ranked position	Treat as unordered categories
Continuous	Position, length, size, sequential ramp; same-shaped marks	Map to an unordered palette
Relational	Node–link or matrix; the layout is itself a choice	Ignore layout dependence
Textual	Labels, annotations, short glyphs	Place directly on an axis

Takeaway: A categorical variable on a continuous axis implies an ordering that is not in the data.

Exercise 10.4: Apply the 3-step viz-making framework

- Name the unit of analysis for your chosen finding
- List the dimensions; mark the most important one
- Classify each data type and pick an honest element
- Reshape into the typed dataframe: one row per unit, one column per dimension
- Cross-check against the encoding hierarchy

You will need

- One finding from your earlier work you want to communicate
- The dataframe behind it

Deliverable

The visualization-ready dataframe, plus a note recording the unit, the dimensions, each data type, and the encoding assigned to it.

Kirk's cycle turns the dataframe into an argument

- Brief: audience, question, purpose, in two or three sentences
- Data work: sketch at least two chart types before coding one
- Editorial thinking: include, exclude, emphasize, leave to discover
- Design solution: build, then test against trustworthy, accessible, elegant

Takeaway: Sketching two alternatives on paper is cheaper than rebuilding one in code.

Exercise 10.5: One publication-quality figure

- Write the brief
- Sketch two alternatives; pick the better encoding match
- Make the editorial choices explicit
- Build the chart and run Kirk's three principles
- Reflect: which features are data, which are layout?

You will need

- The typed dataframe from Exercise 10.4
- A plotting library and paper for sketching

Deliverable

One figure with its caption, the written brief, the rejected sketch, and one sentence separating data features from layout choices.

Beyond one figure: the integrative artifact

- At least three coordinated views for a non-specialist audience
- Form follows audience: dashboard, reproducible report, or annotated notebook
- A folder of figures is not an integrative artifact

Takeaway: Coordination comes from shared filters or shared narrative, not proximity.

Exercise 10.6: Build an integrative communication artifact

- Name the audience and the three or four questions they will ask
- Choose the form and tool; record the reasoning in project memory
- Implement at least three coordinated views
- Document the analytical choices behind the surface
- Plan graceful degradation: static fallback, keyboard, screen readers
- Write the maintenance memo: what breaks in five years?

You will need

- The dataframe from Exercise 10.4 and the figure from Exercise 10.5
- One tool: Streamlit, Dash, Shiny, Observable, Quarto, or Jupyter
- Starter templates at the companion site

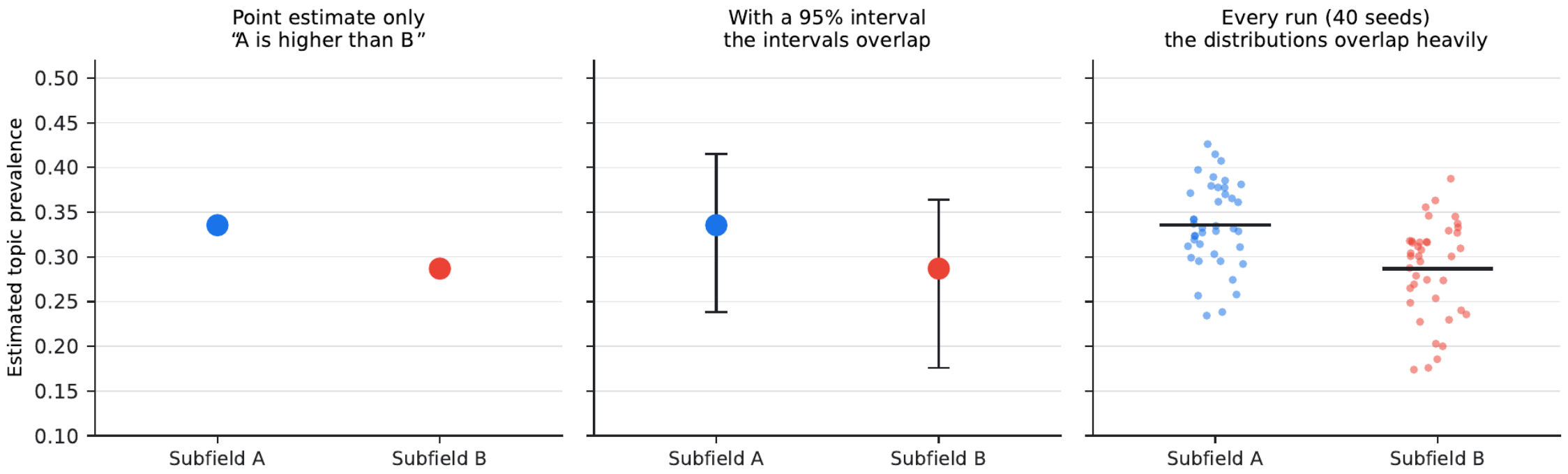
Deliverable

The artifact (a link, or source plus run instructions) and a short note separating the durable analysis from the disposable interface.

↗ **Chapter 10 starter scaffolding** · cssprimer.org/ch10/

The same result under three levels of disclosure

§10.3



One simulated topic-prevalence estimate, forty seeds: point estimates only, then 95% intervals, then every run.

Takeaway: Nothing was recomputed between panels; only how much the reader sees changed.

Match the uncertainty display to the audience

§10.3

- Error bars are the technical default, and widely misread
- Icon arrays make a rate countable: 17 shaded figures out of 100
- Hypothetical outcome plots animate single draws; ensembles overlay many runs
- Sensitivity tables expose dependence on any one modeling choice

Takeaway: Pick one strategy and justify the choice in two or three sentences.

Exercise 10.7: Visualize the uncertainty in one result

- Identify the kinds of uncertainty present: statistical, model, algorithmic
- Quantify: at least three seeds or three specifications, or a bootstrap
- Choose a visual strategy fit to the audience, and justify it
- Integrate the view into your Exercise 10.6 artifact
- Caveat AI confidence: self-reported probabilities are not calibrated

You will need

- One result whose uncertainty you can actually estimate under reruns

Deliverable

The uncertainty-aware figure and a caption stating plainly what the uncertainty represents and what it does not cover.

The pipeline end to end

- 10.4: the typed, visualization-ready dataframe
- 10.5: Kirk's cycle and the encoding hierarchy produce the figure
- 10.6: coordinated views become the integrative artifact
- 10.7: the uncertainty layer makes stability visible

Takeaway: One finding now travels from raw records to audience-ready, honesty included.

Useful tools, organized by task

Task	Options
Preregistration	OSF (full templates, repositories); AsPredicted (short form)
Environment management	venv, conda, renv, Docker
Version control and archiving	Git, GitHub; Zenodo for a citable DOI
Static visualization	ggplot2, matplotlib, seaborn, Plotly
Interactive dashboards	Streamlit, Dash, Shiny, Observable, Tableau Public
Reproducible reports	Quarto, Jupyter, R Markdown

↗ **Communication tools** · cssprimer.org/tools/communication

Takeaway: Tools change quickly; the maintenance memo is why principles must outlast them.

Discussion

- Which analytical choice was hardest to declare honestly in your preregistration, and what does that say about your forking paths?
- Which finding was hardest to communicate to a non-specialist without distorting it? Was the cut a simplification or a loss?
- Where did the uncertainty layer change the story? Is the honest version less persuasive, or differently persuasive?
- If AI drafted captions or alt text, what did verification catch, and what does that imply about publishing unreviewed drafts?

Key concepts from Chapter 10

Adapted preregistration: Lock what you can, write procedures for the rest, label confirmatory versus exploratory

Lock file: Exact package pins that freeze the analysis environment

Gundersen R1–R4: Reproducibility rungs by what is shared: description, code, data, full package

Promptbook: The complete operative prompt set with model ID and parameters; Knowledge

Promptlog: Dated revision history with rationale; Memory

Unit of analysis: What one mark on the chart represents; the most common source of misleading charts

Typed dataframe: One row per unit, one column per dimension, each tagged with its data type

Small multiples: A grid of subplots, one per category; faceting spends a dimension

Integrative artifact: Three or more coordinated views for a non-specialist audience

Uncertainty layer: The added view that makes precision and stability visible beside point estimates

Materials on the companion site

Chapter 10 hub cssprimer.org/ch10/

Exercise materials plus the Quarto report and Streamlit dashboard skeletons

Communication tools cssprimer.org/tools/communication

The maintained list behind this chapter's Useful Tools section

Getting started cssprimer.org/setup/

Environment help when the freezing exercises surface version conflicts

Where to go next

- Revisit: Chapter 9 for the principles behind every exercise you just ran
- Read: Chapter 11, where communication choices become ethics questions
- Use: starter templates and the updated tool list at the companion hub