Computational & AI-Assisted Methods for Social Sciences

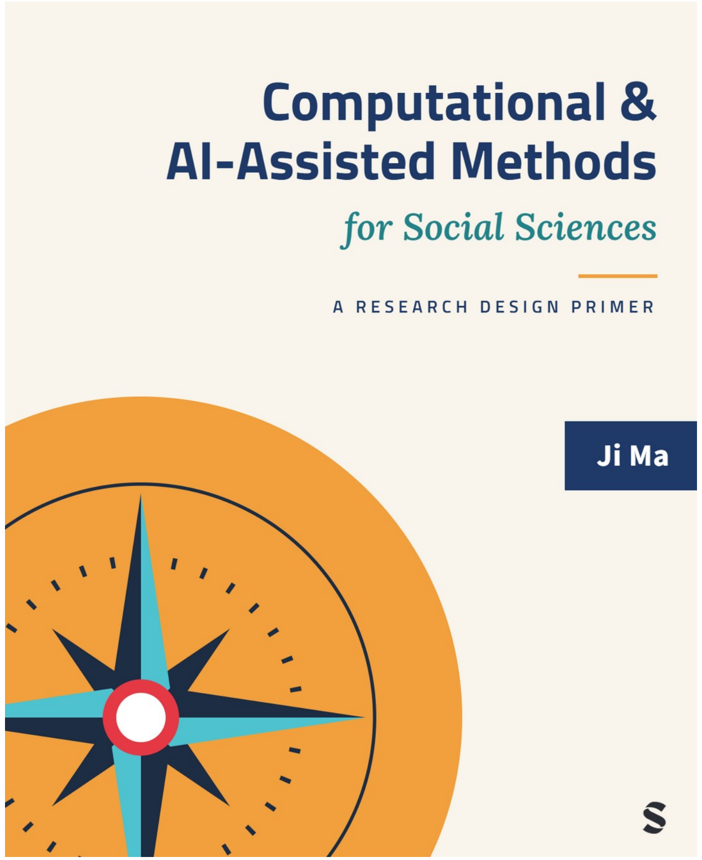
A Research Design Primer

Chapter 8

Applying Computational Methods: Tools, Tips, and Exercises

Ji Ma

cssprimer.org

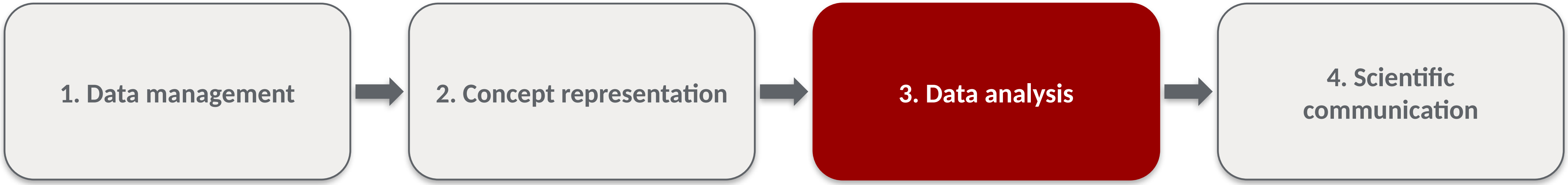


Learning objectives

By the end of this chapter, you can:

- Construct and analyze a coauthor network using iterative algorithms
- Run a diffusion simulation and show how update order changes outcomes
- Train and compare linear and non-linear predictors with appropriate metrics
- Apply centrality, embeddings, and community detection under distinct epistemic purposes
- Draft a validity statement for the AI-assisted parts of your pipeline

Still stage 3: from theory to practice



Chapter 7 said what analysis can claim; this chapter has you earn those claims in code.

Takeaway: Same stage, second half: the exercise chapter for data analysis.

Same data, different questions

- Inputs already exist: the coauthor network (Chapter 4), the coded OpenAlex corpus (Chapter 6)
- The chapter reuses them under every capacity and every purpose
- Arc: capacities → purposes → AI validity → craft reflection
- Worked notebooks, one per exercise, at cssprimer.org/ch08

Takeaway: The same data serve different analytic purposes depending on the question you bring.

§8.1

Computational capacities in practice

Build a network, let a process unfold, bend a line, compress a vocabulary.

Choose your network tools deliberately

- **NetworkX** (Python): intuitive syntax, extensive documentation
- **igraph** (C core; Python and R): faster on large networks
- **Statnet** (R): statistical models of tie formation (ERGMs)
- **Gephi** and **Cytoscape**: interactive, publication-quality visualization
- Record the choice and the reasoning in project memory

Takeaway: Pick for the task at hand; the concepts transfer across libraries.

Exercise: network construction and iterative search

- Build an undirected graph: authors are nodes, coauthorship is an edge
- Report nodes, edges, density, components, and largest-component diameter
- Compute a shortest path between two distant authors; interpret it
- Name the capacity that makes shortest paths feasible

You will need

- Your bibliographic dataset from Chapter 4, with an author field per record
- Python or R with NetworkX or igraph

Deliverable

Construction script plus a half-page note: the basic statistics, the shortest-path finding with interpretation, and the capacity answer.

↗ **Network-analysis notebook** · cssprimer.org/notebooks/ch08-network-analysis.ipynb

Update order is part of the model

- Diffusion rule: each adopter's coauthors adopt with probability p
- Synchronous: everyone evaluates the same snapshot at once
- Asynchronous: B adopts first, so C now sees two adopters
- Same rules, same seed — different trajectories

Takeaway: Timing choices are modeling choices: justify, report, and test them.

Exercise: simulation and emergence

§8.1.2

- Seed one author; let adoption spread with probability p until it stops
- Vary the seed and p (say 0.1 versus 0.3); compare final adoption
- Run synchronous and asynchronous updates; compare speed and extent
- List the assumptions your rules embed

You will need

- The coauthor network from the previous exercise
- The same programming environment; no new data

Deliverable

Simulation script, a comparison table of runs (seed, p, update order, final adoption rate), and a paragraph connecting the update-order result to temporal dynamics.

Exercise: non-linear prediction

- Pick a binary target: above-median citations, or a topic code
- Build features from metadata and TF-IDF abstract vectors
- Fit logistic regression and a random forest; cross-validate
- Compare accuracy, precision, and recall on held-out data
- Explain any gap via activation functions and universal approximation

You will need

- The coded OpenAlex corpus from Chapter 6
- A machine-learning library such as scikit-learn (primer at cssprimer.org/primers/ml)

Deliverable

A performance table for both models on held-out data, plus a half-page interpretation of the comparison and the capacity behind it.

↗ **Supervised learning, gently** · cssprimer.org/primers/ml

Metrics when classes are imbalanced

- A majority-class guesser can score high accuracy
- Precision, recall, and F1 focus on the cases you care about
- AUC-ROC: threshold-free ranking quality, from 0.5 (random) to 1.0
- Severe imbalance: check the precision-recall curve too

Takeaway: Accuracy alone can flatter a model that finds nothing.

Embeddings compress the vocabulary

- Every unique word is a dimension; a corpus has thousands
- Word2Vec learns 100–300 dense dimensions by predicting neighboring words
- Similar meanings land close; analogies become vector arithmetic
- king − man + woman ≈ queen

Takeaway: Co-occurrence patterns, iterated over a corpus, become geometry.

Exercise: word embeddings

- Train Word2Vec on your abstracts, or load a pre-trained model
- Query the 10 nearest neighbors of 3–5 key terms
- Test one domain analogy by vector arithmetic
- Reflect: what does the compression keep, and what does it lose?

You will need

- Abstracts from your coding corpus (Chapter 6)
- gensim or a pre-trained model; worked notebook at cssprimer.org/ch08

Deliverable

Embedding script or notebook plus a results note: the neighbor queries, the analogy test, and the compression reflection.

§8.2

Epistemic purposes in practice

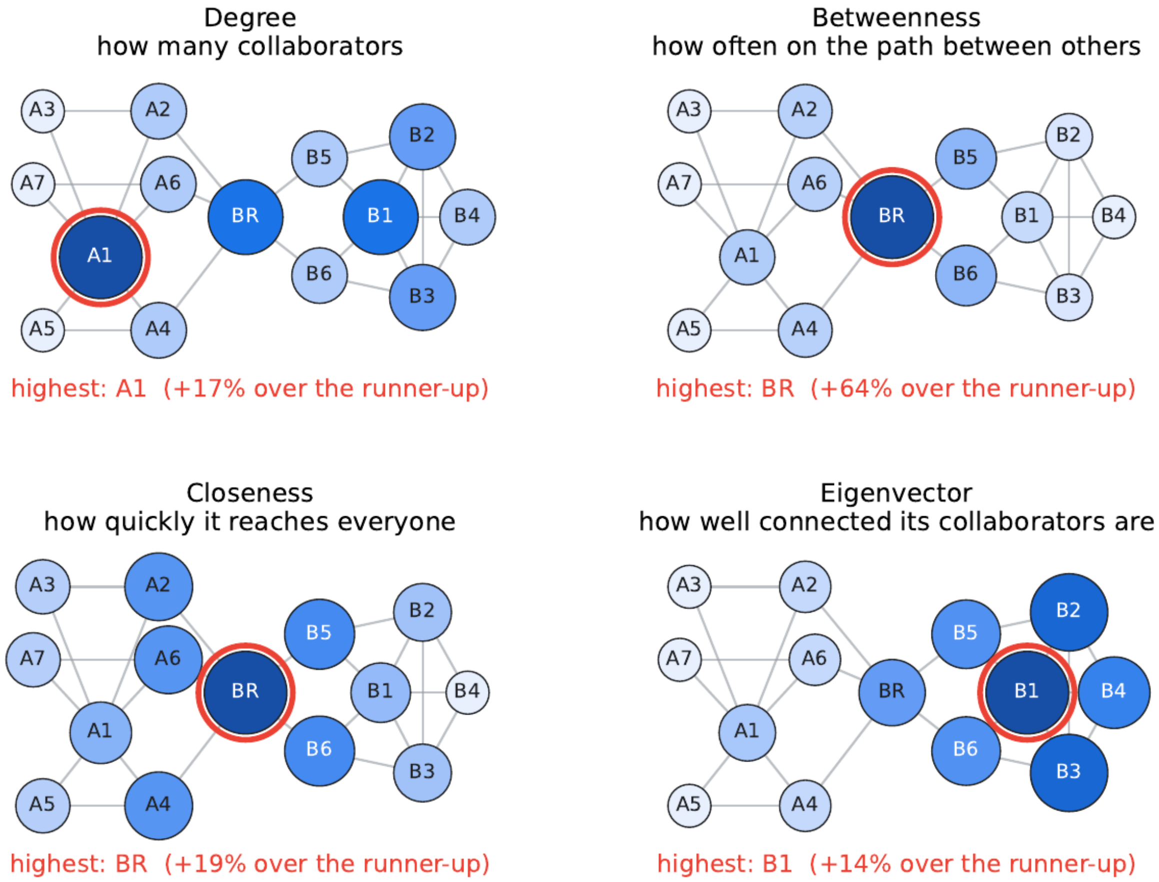
The same tools, run under four different research goals — and the challenge each raises.

Five centrality measures, five meanings of importance

Measure	What it counts	In a coauthor network
Degree	Direct connections	How many distinct collaborators
Betweenness	Presence on shortest paths	Brokers bridging separate groups
Closeness	Distance to everyone else	Reach with few intermediaries
Eigenvector	Quality of connections	Collaborating with the well-connected
PageRank	Weighted incoming links	An eigenvector variant for directed networks

Takeaway: Choosing a measure is a measurement decision, not a display preference.

One network, three different 'most important' authors



Identical graph and layout in all four panels; only the measure changes. Degree picks A1, eigenvector picks B1, betweenness and closeness pick the bridge BR.

Takeaway: Three answers on one graph: popular, well-connected, or well-placed.

Social capital gives the measures meaning

- Social capital: value embedded in relationships — cooperation, information, trust
- Bonding: strong ties within tight, homogeneous groups
- Bridging: weaker ties spanning groups, carrying diverse information
- Matching measure to concept is an instrument-validity exercise

Takeaway: The math of a measure and the concept it represents are not the same thing.

Exercise: centrality and social capital

- Argue which measures operationalize bonding, and which bridging
- Compute degree, betweenness, closeness, and eigenvector for all authors
- Find the top bonding and bridging authors — same person?
- Visualize twice: node size by bonding, then by bridging
- Memo: does high betweenness here really mean bridging capital?

You will need

- The coauthor network from the first exercise
- A visualization tool: Gephi, Cytoscape, or your library's plotting

Deliverable

A centrality table for your top authors, the two visualizations, and the half-page semantics-gap memo.

Embeddings as cultural measurement

- Project words onto axes defined by anchor terms (Kozlowski et al. 2019)
- Embeddings trained on ordinary language reproduce human-like biases (Caliskan et al. 2017)
- Historical embeddings track a century of shifting stereotypes (Garg et al. 2018)

Takeaway: This instrument describes relationships between concepts, not between actors.

Exercise: measuring semantic dimensions

- Choose a contrast for your field (theoretical–empirical, quantitative–qualitative)
- Build the dimension from anchor-word vector differences
- Project key terms; check the poles against domain knowledge
- If your corpus spans years, compare embeddings across periods
- State what would turn this description into an explanation

You will need

- The embeddings from the word-embeddings exercise
- Two hand-picked anchor-word lists defining your poles

Deliverable

A projection table or plot locating your key terms on the dimension, a paragraph of interpretation, and the descriptive-versus-causal reflection.

A detected community is not yet a social group

- Louvain finds densely connected clusters by optimizing modularity
- Density is not shared identity, norms, or collective action
- The resolution parameter changes how many communities you get
- Whether clusters map to real subfields is an empirical question

Takeaway: Try at least two resolutions; no single setting is the correct one.

Exercise: community detection and explanatory analysis

- Run Louvain on the coauthor network; count the communities
- Characterize each with venue, year, and topic codes
- Test whether membership predicts topic (chi-squared or logistic regression)
- Name the likely confounders; sketch a stronger causal design
- Contrast the algorithmic and social definitions of community

You will need

- The coauthor network, plus OpenAlex metadata and topic codes from Chapter 6

Deliverable

A one-page analysis note: community characterization, the association-test result, and the causation and semantics reflections.

Interpretation under opacity

- Complex models resist inspection: many features, many interactions
- Feature importances and partial dependence plots are partial remedies
- They summarize; they do not fully explain
- Rudin: prefer interpretable models when performance is comparable

Takeaway: An approximate explanation of a black box is still not the model.

Exercise: model interpretation and epistemic opacity

- Extract feature importances; do the top features make substantive sense?
- Plot partial dependence for one informative feature
- Memo: could a stakeholder contest your model's verdict on a paper?
- Fit a shallow interpretable model; price the accuracy trade-off
- State what a causal coauthorship claim would additionally require

You will need

- The trained models and held-out data from the non-linear prediction exercise; no new data

Deliverable

The feature-importance summary, one partial dependence plot, the opacity memo, and the interpretable-alternative comparison.

The integrative step: does it survive a shifted world?

- Merge network features (centrality, communities) with text features (embeddings, topics)
- Ask whether each family improves prediction over the other
- Train pre-2015, test post-2015: the drop measures instability
- Causal ML aims for effects that persist across subgroups (Brand et al.)

Takeaway: Cross-domain testing approximates the integrative standard: predicting under change.

Exercise: integrative prediction and cross-domain evaluation

- Merge centrality, community, embedding, and topic features into one table
- Re-run the prediction task with the combined feature set
- Split by period or venue; train on one side, test on the other
- Interpret the degradation: which patterns proved durable?

You will need

- Outputs of the earlier exercises: centrality scores, community labels, embeddings, and the prediction task

Deliverable

The merged analysis table (or the script that builds it), a within-period versus cross-period performance comparison, and a closing paragraph on the integrative standard.

§8.3

Validity of AI-assisted analysis

What exactly are you claiming when a model's outputs are part of your evidence?

Why benchmark accuracy is not validity

- Bisbee et al.: synthetic averages matched, variance collapsed, coefficients flipped
- Outputs drifted with prompt wording and over time
- Broska et al.: LLM outputs as proxies; humans stay the gold standard
- Prediction-powered inference buys efficiency without assuming correctness

Takeaway: State what validity you claim, and what evidence supports the claim.

Exercise: reflecting on AI-assisted validity

- Sample records coded by both the LLM and a human or dictionary
- Map where they agree, where they diverge, and why
- Ask what high accuracy does and does not establish
- Draft a methods-section validity statement

You will need

- The LLM coding outputs and the dictionary or hand-coded labels from Chapter 6, side by side for the same records

Deliverable

A disagreement analysis (which cases diverge and why) plus a methods-section paragraph stating the validity claimed and the evidence behind it.

The toolbox, organized by task

Task	Tools	Note
Network analysis	NetworkX, igraph, Statnet	Statnet for ERGMs
Network visualization	Gephi, Cytoscape	Cytoscape for attribute-rich graphs
Machine learning	scikit-learn, XGBoost, LightGBM	Boosting for tabular performance
Word embeddings	gensim, Sentence-Transformers	Pre-trained models for small corpora
Simulation	Mesa, NetLogo	NetLogo is visual and lower-code
Model interpretation	SHAP	Global and local attributions

↗ **Computational-analysis tools** · cssprimer.org/tools/analysis

Takeaway: The updated list, with per-exercise setup notes, lives at cssprimer.org/tools/analysis.

Discussion: the craft

- Which tool choices did you make this chapter, and which would you change when advising a classmate?
- Where did an exercise break, and what did the failure teach you about the method itself?
- If AI wrote or debugged your code, how did you verify it did what you intended?
- Compare effort spent preparing data against effort spent modeling — what does the ratio say about where the real work lies?

Key terms from Chapter 8

Centrality: Family of measures of structural position; each encodes a different 'importance'

Bonding vs. bridging capital: Strong ties within groups versus weaker ties spanning them

Synchronous vs. asynchronous updating: Whether agents evaluate at once or one by one; can change outcomes

Cross-validation: Estimating performance by rotating held-out folds

AUC-ROC: Threshold-free ranking quality, from 0.5 (random) to 1.0 (perfect)

Word2Vec: Embeddings learned by iteratively predicting neighboring words

Louvain resolution: Parameter controlling community size; no single correct setting

ERGM: Models the observed network as one draw from a stochastic tie-formation process

SHAP: Shapley-value feature attributions, global and local

Prediction-powered inference: Uses model predictions for efficiency without assuming they are correct

Materials on the companion site

Chapter 8 hub cssprimer.org/ch08/

Exercise materials, prerequisites, and the deliverable for this chapter

Network-analysis notebook cssprimer.org/notebooks/ch08-network-analysis.ipynb

Runs end to end on the provided co-citation network

Computational-analysis tools cssprimer.org/tools/analysis

NetworkX, scikit-learn, embeddings, and the rest of the toolkit

Supervised learning, gently cssprimer.org/primers/ml

Background for the prediction exercises, with a runnable demo

Where to go next

- Revisit: Chapter 7's reflection checklist, now that every concept has a hands-on counterpart
- Finish: deliverables into your project repository, choices logged in project memory
- Next: Chapter 9 — communicating and reproducing what you just built