

6. AI for Research

Rice Business 2026

Kerry Back
Rice Business

Today's Session

1

Knowledge Database

Papers, abstracts, and notes — linked and searchable



2

Team of Agents

Guidance, data analysis, writing, review



3

Logs

"What did my coauthors do last week?" "Why did we change that?"



4

WRDS and jgsrc1

Use Claude with WRDS, use Claude on jgsrc1

Knowledge Database

What Do We Want?

- A local, private store of PDFs, abstracts, and notes
- Full-text keyword *and* semantic (meaning) search
- Import PDFs, a folder of PDFs, web search results, Zotero
- Notes tied to the papers that prompted them
- Export BibTeX

Three Storage Types

Relational (SQL)

- The facts: title, authors, year, DOI, citekey, reading status, project tag
- Exact filters and joins — “unread, 2020 or later, this project”
- Keyword search lives here too — over the full text

Vector

- Meaning, not wording — finds the paper when you don’t recall the author’s vocabulary
- Passages get embedded, so a hit returns the paper **and the page**
- Your own notes embed here, alongside the papers

Graph

- Who cites whom — edges, not rows
- Trace a result forward to its critics and back to its roots
- The query it unlocks: what my library leans on that I **don’t own yet**

LitDB: All in One SQLite Database

- Ordinary tables for papers, notes, passages, projects
- SQLite's own full-text index for keyword search
- Every chunk compared on every query; cosine similarity is one matrix multiplication
- Citations as an edge table, queried with joins
- No server, no extension, nothing to keep running

Perfectly fine (according to Claude) for a research corpus of a few thousand papers.

Notes

Whatever format you like. Plain text is fine. Tell Claude: “add <filename> to litdb” Claude will read and try to classify.

- Kind — idea, summary, critique, question, todo, or quote
- Which paper in your library it is about, by searching for it
- How it relates to that paper — about, supports, contradicts, extends, uses-method, uses-data
- The page or the verbatim quote it points at
- Project tag — defaults to the folder you are working in
- Whether it is confidential — unpublished ideas and referee material are embedded locally, never sent to an API

It says what it filed, so you can correct a guess you never had to make

Finding Papers You Don't Have

Where it looks

- Your own library first, always
- Then **OpenAlex** (free, no key) and **Semantic Scholar**
- Every hit says whether you already have it
- Or work the citation graph: this paper's references, who cites it, what your library leans on that you don't own

What lands in the database

- Bibliographic data — authors, year, venue, DOI — and a citekey for your .bib
- The abstract, indexed and embedded like any other text
- Then Claude goes after the PDF: open-access copy, arXiv/SSRN/NBER, the author's page
- No PDF? The abstract still makes the paper findable by meaning

When the Paper Is Behind a Paywall

Claude can drive your browser

- Uses the Claude in Chrome extension
- You sign in to the library proxy
- Claude pauses at login pages and CAPTCHAs and hands the keyboard back
- Then it carries on: find the PDF and download it
- My Claude frequently chooses the Playwright MCP server instead; not sure why

One paper you are entitled to, fetched through your own subscription, is ordinary use. Scripted bulk downloading is not — it violates publisher terms and gets the campus IP range cut off. Claude is instructed never to touch Sci-Hub.

Comparisons

	litdb	NotebookLM	Obsidian	Zotero
Where it lives	Your disk	Google’s cloud	Your disk	Your disk + their cloud
Searches full PDF text	Yes	Within one notebook	Only with plugins	Yes, keyword
Search by meaning	Yes	Yes	Plugin	No
Citation graph	Yes	No	The links you drew, not citations	No
Writes a .bib	Yes	No	Plugin	Yes
Driven by	Claude, in chat	Its own web UI	You, by hand	You, by hand

Get Started

- Tell Claude to install the plugin `litdb@kerryback` (tell it first to install the marketplace `github.com/kerryback/skills` if you haven't done that).
- Start a new conversation. Ask Claude to describe the `litdb` skill and how to use it.

Research Agents

What We Want

Guidance

"I have an idea. What do you think?" to "How deeply should I go into X?"

Data analysis

Pull data, analyze, figures, tables

Writing

"Write a draft of the data section", "Edit the introduction", ...

Review

From "check that the numbers in the text match the tables" to "give me a full referee report"

Actor-Critic

Actor

- Produces the thing: the idea, the sample, the paragraph
- Charged with being generative and specific, not with being right
- Optimized to make something exist

Critic

- Charged with finding what is wrong with it
- Has its own standard of “good” and its own charter — not “check this over”
- Every objection must name the **check that would settle it**

Ask one agent to draft and then review, and it reviews the thing it just decided was good. The two jobs have to be separate calls with separate instructions — and the more independent the critic, the more it catches.

Agent Teams

Guidance

A proposer comes up with ideas, and adversaries shoot them down

Data analysis

Analyst and replicator build the sample independently, and must agree before analysis starts

Writing

Read from different angles: accuracy, persuasion, devil's advocate

Review

Multiple independent reports

How to Get Independent Views

Subagents

Ask Claude to generate them. A subagent starts fresh with instructions from the main agent. It does not see the conversation history — which is the whole point.

Different models

Sonnet, Opus, Fable for different seats. Different capability, but still correlated priors — they were trained by the same people.

Different families

Claude can call ChatGPT, Gemini, DeepSeek, ... Genuinely different blind spots. Uses your API keys (and your credit card).

Research Plugin

Ask Claude to install `research@kerryback`

- `/research-setup` generates a quiz.
- Empirical or theory? Who are the coauthors? Where does the data live? Does anyone use Overleaf?
- Then it sets up a folder structure and wires other pieces.

- `CLAUDE.md` is the operating manual
- `protocols.html` is the same project explained to a person — what a coauthor reads on their first day.
- [Example protocols.html](#)

Writing Guide

The plugin has a style guide

- Mostly Lu Han's econ-write that contains Cochrane's Writing Tips for PhD Students plus other style guides
- Also some "avoid AI speak" instructions

It should be a living document

- Claude is instructed to learn from your edits and add what it learns to the style guide
- For your next project, tell Claude to start with the style guide from your last project

I don't know how well this will work. It is brand new. But if it doesn't work, tell Claude to fix it!

Overleaf and the Git Bridge

- On setup, you can set up Overleaf integration
- The draft folder (with figures and tables subfolders) is set up as a Git sub-tree and linked to an Overleaf project
- Pull the sub-tree when you start to work locally and pull again (for any changes since you started) and push (to share your changes) when you finish editing to stay in sync with coauthors on Overleaf (who work as usual)

Slash Commands

Command	What it does
/research:setup	The interview, then builds or restructures the repo
/round	Finishes a round — writes what you took up and what came of it, and promotes whatever settled
/round start	Optional. Claims the round when a coauthor might be working right now
/refresh	Context is getting long: writes your handoff and stops, so you can /clear. Keeps the round open
/report	An HTML report: executive summary, what is settled, open issues, chronology
/critique	The reviewer panel and the paths forward
/econ-review	Referee-grade report, if installed
litdb	No command — just ask, in plain language

Example report

Logging

I said to Claude this morning:

Kevin said he was working on optimizing over industry group and market cap for peer selection. Explain what he is doing and what the status is.

It explained the method in detail, then told me the work was finished and explained the results — and waiting on a decision about whether it goes in the main text or the internet appendix.

Where The Answer Came From

state.md

Current state of the project and coauthor/date/time stamped record of all changes in state. Claude updates periodically. Can force update with '/round'

runs.jsonl

Every script: date/commit/round and the files they read and wrote
Written automatically by a hook

git commit log

What landed and when
chronology merges all three into one timeline.
/report produces a human-readable version

Get Started

- Tell Claude to install the plugin `research@kerryback` (install the marketplace github.com/kerryback/skills first if you haven't).
- Start a new conversation in the folder you want the project in. Ask Claude how to use the research skill.
- Then `/research:setup`.

WRDS

Getting Data from WRDS

Ask Claude to install wrds@kerryback

Contains

- Python scripts from WRDS,
- Python scripts from OpenAssetPricing (Chen & Zimmerman)
- variable definitions from French's data library

SKILL.md

- Install wrds with no-deps option
- Aware of annual June 30 FF convention and monthly 4-month lag convention
- Warns user if WRDS, Chen-Zimmerman, and/or Fama-French define a requested variable differently

Use Claude on jgsrc1

Remote SSH

1. Update

- Help → Check for updates
- Academic Studio to 1.3

2. Open Remote SSH

- Settings (bottom left) → Command Palette → Remote SSH: Connect to Host
- You must be on Rice Owls or the VPN

3. Log in

- `<netid>@jgsrc1@rice.edu`
- Can create a profile

4. Connect

- Prompted for your password.
- Open a folder — prompted **again**.

What You Should Get

Claude Code

On this first connection, the Claude Code extension for VS Codium should install itself into your home folder **on jgsrc1** — not on your laptop

Everything else

The other extensions bundled with Academic Studio that need to live on the remote machine should install as well

Install Python

If you don't have a Python environment installed, I recommend you create one (one time setup).

Prompt

Install uv. Use it to create a Python 3.14 venv. Install the data analysis stack. Install wrds with the no-deps option. Add to CLAUDE.md to use the venv.

Try It

Ask Claude anything. Check that it works.