

Agentic AI Bootcamp

Research Workflows, Teaching & Applications

Erkmen G. Aslim & Emily Beam

Department of Economics, University of Vermont

Session 2 · April 27, 2026 · Old Mill A500

thinkingwithagents.github.io

Where We Left Off & Today's Plan

Session 1

- ▶ Agentic AI vs. chat AI
- ▶ Context window & planning
- ▶ CLAUDE.md, PLAN.md, SKILL.md
- ▶ Permissions & safety
- ▶ Apps: voice file, website, lecture builder spec

Emily: Research

1. **Code audit** — 51 files to 8, found a real bug
2. **Paper audit** — AI vs. 11 real referee reports
3. **New analysis** — Lee bounds, live

One project, three use cases.

Erkmen: Teaching

1. **Lecture Builder** — papers to slides
2. **Research Brainstorm** — stress-test live
3. **Web Scraper** — novel data in 10 min

Setup: Install Your Skill Toolkit



Anthropic skills (one-time)

1. Add the marketplace:

```
/plugin marketplace add anthropics/skills
```

2. Install the document-skills bundle:

```
/plugin install  
document-skills@anthropic-agent-skills
```

Gives you: pdf, docx, pptx, xlsx, frontend-design, skill-creator, algorithmic-art.



Third-party skills via npx

Full economist pack:

```
npx skills add thinkingwithagents/skills
```

One skill at a time:

```
npx skills add thinkingwithagents/skills  
--skill research-brainstorm
```

Includes: research-brainstorm, find-data, academic-beamer-deck.

Setup: Install Your Skill Toolkit



Anthropic skills (one-time)

1. Add the marketplace:

```
/plugin marketplace add anthropics/skills
```

2. Install the document-skills bundle:

```
/plugin install  
document-skills@anthropic-agent-skills
```

Gives you: pdf, docx, pptx, xlsx, frontend-design, skill-creator, algorithmic-art.



Third-party skills via npx

Full economist pack:

```
npx skills add thinkingwithagents/skills
```

One skill at a time:

```
npx skills add thinkingwithagents/skills  
--skill research-brainstorm
```

Includes: research-brainstorm, find-data, academic-beamer-deck.

! /skill-creator is **not** bundled with Claude Code by default — it ships inside document-skills. Install it before App 3.

Emily: Research Workflows

Code Auditing, Paper Auditing & Analysis

Application 1

Code Audit & Refactoring

Bangladesh Remote Education RCT

The Project

- ▶ Paper on remote education in Bangladesh
- ▶ Evaluates TV instruction, adaptive edtech, data subsidies, and teacher support during COVID-era school closures
- ▶ **Keeps getting rejected** — 8 submissions 😞
- ▶ One project, three AI use cases

Three use cases:

1. Code audit

Restructure 6 (📄) years of accumulated code into a clean replication package

2. Paper audit

Adversarial review — catch what referees would catch

3. New analysis

Referee says “do Lee bounds” — so we do it, live

Code Audit: Before

21_power_simple.do	Oct 22, 2024 at 8:51 PM	11 KB	Stata Do-file
21_power.do	Oct 21, 2024 at 1:16 PM	5 KB	Stata Do-file
18_cleaned_tables_het_Mar26.do	Oct 21, 2024 at 1:16 PM	45 KB	Stata Do-file
15_3_coef_plots_2024Sep.do	Sep 4, 2024 at 8:58 AM	33 KB	Stata Do-file
25_behaghel.do	Jun 17, 2024 at 2:48 PM	13 KB	Stata Do-file
06_balance_04Jun24.do	Jun 14, 2024 at 6:19 PM	16 KB	Stata Do-file
18_cleaned_tables_het_attrit_2024_jun06.do	Jun 14, 2024 at 6:03 PM	17 KB	Stata Do-file
03_vargen.do	Jun 14, 2024 at 2:55 PM	78 KB	Stata Do-file
leebounds_temp.do	Jun 14, 2024 at 2:35 PM	2 KB	Stata Do-file
09_attrition_14Feb23.do	Jun 12, 2024 at 2:07 AM	20 KB	Stata Do-file
24_callegain_temp.do	Jun 6, 2024 at 3:38 PM	3 KB	Stata Do-file
06_lasso_2024_apr09.do	Jun 4, 2024 at 3:16 PM	35 KB	Stata Do-file
12_descriptive_insights_ins_3103.do	Jun 4, 2024 at 9:52 AM	66 KB	Stata Do-file
01_master_2024_Apr09.do	Jun 4, 2024 at 6:55 AM	5 KB	Stata Do-file
06_lasso_2022_september12_treat.do	Apr 22, 2024 at 10:04 AM	74 KB	Stata Do-file
lasso.xlsx	Apr 19, 2024 at 1:59 PM	3 KB	Micros...k (.xlsx)
06_lasso_2022_september12_tr.textClipping	Apr 9, 2024 at 2:16 PM	910 bytes	Text Clipping
15_3_coef_plots_2024Mar27.do	Mar 27, 2024 at 11:53 AM	30 KB	Stata Do-file
07_outcome_domains_heterog_gender_age_eb.do	Mar 26, 2024 at 4:48 PM	23 KB	Stata Do-file
18_cleaned_tables_het_attrit_2024_feb9.do	Mar 26, 2024 at 4:23 PM	17 KB	Stata Do-file
07_outcome_domains_heterog_attrition_2024_feb09.do	Feb 18, 2024 at 2:32 PM	23 KB	Stata Do-file
18_cleaned_tables_het.do	Oct 8, 2023 at 1:22 PM	14 KB	Stata Do-file
04_outcome_domains_14Feb23.do	Oct 8, 2023 at 1:22 PM	71 KB	Stata Do-file

The problem: Life happens

- ▶ **51 do-files** across 4 subdirectories
- ▶ Master file from **June 2021** — 5 years stale
- ▶ Date-in-filename: **_07sep, _02July1130am**
- ▶ Author-in-filename: **_eb, _e**
- ▶ No documentation, no version control

This is **extremely common** in economics.

>_ THE PROMPT

What I Asked Claude to Do

```
Create a clean, replicable version of my do-files that generate the analysis in my  
Overleaf paper. I'll give you the file paths.
```

```
Archive all old files -- don't delete anything.
```

```
The goal is a replication package. No results should change.
```

```
Write a comparison script so I can diff the new output against a "gold standard"  
version from the current code.
```

→ One prompt. AI reads 51 files, proposes a new structure, archives the old ones, and builds a test harness.

Code Audit: After

└─ _globals.do	Mar 21, 2026 at 4:55 PM	1 KB	Stata Do-file
└─ 02_datagen.do	Mar 14, 2026 at 12:10 AM	42 KB	Stata Do-file
└─ 03_vargen.do	Mar 21, 2026 at 8:15 PM	79 KB	Stata Do-file
└─ 05_balance_04Jun24.do	Mar 14, 2026 at 12:10 AM	16 KB	Stata Do-file
└─ 06_lasso_2024_apr09.do	Mar 21, 2026 at 8:15 PM	35 KB	Stata Do-file
└─ 12_lasso_2022_version1.do	Mar 21, 2026 at 10:06 PM	71 KB	Stata Do-file
└─ 15_3_coef_plots_2024Sep.do	Mar 21, 2026 at 8:14 PM	32 KB	Stata Do-file
└─ 18_cleaned_tables_het_Mar26.do	Mar 21, 2026 at 8:13 PM	44 KB	Stata Do-file
> ado	Mar 21, 2026 at 4:55 PM	--	Folder
> Archive	Mar 21, 2026 at 11:53 AM	--	Folder
> Maps	Mar 14, 2026 at 12:10 AM	--	Folder
> MHT	Mar 21, 2026 at 8:15 PM	--	Folder

Dimension	Before	After
Do-files	51	8
Master file	2021	Current
Config system	None	Template
Version control	None	Git
Documentation	None	Full

86 old variants preserved in Archive/ — nothing deleted, fully reversible.

Time: ~2 Claude Code sessions (a few hours). **By hand:** days, minimum.

The Bug: A Copy-Paste Typo Hiding in Plain Sight

The \$flags variable list

\$flags = 17 missing-value indicators used as controls in **every regression**. Defined independently in 5 scripts.

3 of 5 scripts (wrong):

```
... _f_c_engaged _f_c_health  
    _c_child_work
```

Missing the `_f_` prefix



Correct:

```
... _f_c_engaged _f_c_health  
    _f_c_child_work
```

Result: `_c_child_work` appeared *twice* in regressions. The actual missing indicator was left out. Survived 10+ versions across 2+ years. No error, no crash — just quietly wrong. **How AI found it:** ran `/code-review` on the new code — it cross-referenced every `global flags` definition across all files.

/code-review: Parallel Agents on the Pipeline

```
run /code-review on /Users/ebase/Dropbox/SiHub/td-remote-ed
Skill(/code-review)
  | Successfully loaded skill
  |
  | Let me explore the project structure and find the code files.
  | Searched for 3 patterns, read 1 file (ctrl+o to expand)
  |
  | Let me read the key project files and identify the active (non-archive) code files.
  | Searched for 2 patterns, read 2 files (ctrl+o to expand)
  |
  | This is a Stata replication package with 13 active do-files. Let me read the pipeline files and do a quick review. I'll dispatch
  | parallel agents to review the files efficiently.
  |
  | Review master, config, and globals. (53s - 1 937 tokens - thought for 6s)
  |  Review master, config, and globals
  |  Review pipeline Stage 1: 03_vargen.do
  |  Review pipeline Stage 1b-1c: lasso scripts
  |  Review pipeline Stage 2a: Regression tables
  |  Review pipeline Stage 2b-2c: plots and MHT
  |  Review non-pipeline scripts
  |  Write consolidated review report
```

```
• Good - infrastructure files are clean and well-organized. Now let me dispatch agents to review the pipeline scripts in parallel.
• Running 5 agents. (ctrl+o to expand)
  | Review 03_vargen.do - 41 tool uses - 41.9k tokens
  |   | Searching for 19 patterns, reading 22 files...
  | Review Lasso scripts - 6 tool uses - 54.6k tokens
  |   | Done
  | Review regression tables script - 3 tool uses - 32.7k tokens
  |   | Done
  | Review coefplots and MHT scripts - 8 tool uses - 42.3k tokens
  |   | Searching for 3 patterns, reading 5 files...
  | Review non-pipeline scripts - 2 tool uses - 10.8k tokens
  |   | Reading 2 files.
  |
  | Review non-pipeline scripts. (3s 10s - 4 8.8k tokens - thought for 7s)
  |  Review master, config, and globals
  |  Review pipeline Stage 1: 03_vargen.do
  |  Review pipeline Stage 1b-1c: lasso scripts
  |  Review pipeline Stage 2a: regression tables
  |  Review pipeline Stage 2b-2c: plots and MHT
  |  Review non-pipeline scripts
  |  Write consolidated review report
```

Q What It Does

- ▶ Reads the full pipeline (master → all do-files)
- ▶ Dispatches parallel review agents
- ▶ Checks correctness, reproducibility, style
- ▶ Produces a prioritized report

This is how it found the \$flags bug — cross-referencing every global definition across all files.

Application 2

Paper Audit with `/econ-audit`

AI vs. 11 Real Referee Reports

Paper Audit: What Is /econ-audit?

Q Adversarial Review Skill

A reusable skill that reads a paper and acts as a **skeptical referee**.

- ▶ Reads full paper (LaTeX, tables, appendix)
- ▶ Checks identification, power, measurement
- ▶ Produces structured audit report
- ▶ Takes ~60 seconds

```
$ npx skills add thinkingwithagents/skills \  
-skill econ-audit
```

What it checks:

1. **Identification:** Is the causal claim valid?
2. **Statistics:** MHT, power, attrition
3. **Measurement:** Are variables measured well?
4. **Methodology:** Design consistency
5. **Presentation:** Framing, length

>_ LIVE DEMO

Running /econ-audit on the Paper

```
$ claude  
/econ-audit main.tex
```

Reads the full paper + tables + appendix.

Produces a structured audit in ~60 seconds.

▶ Switching to terminal...

/econ-audit Results: After One Pass of Revisions

- Audit complete. Report saved to:

```
/Users/ebeam/Dropbox/a2i_project/09_Analysis&Results/10_Paper_Drafts/econ_audit_beam_mukherjee_navarro-sola_2026-04-26.md
```

```
Summary: Minor Issues | 0 critical, 2 high, 4 medium, 3 low
```

```
The core identification – household-level RCT with clustering at caseid – is correctly implemented. The two most important findings:
```

HIGH:

1. **Table notes overstate fixed effects** – Paper tables claim "stratification-cell FE, enumerator FE, and survey-week FE" but the lasso specification code (18_cleaned_tables_het_Mar26.do:221-223) only absorbs strat_cells. The MHT code explicitly confirms this override. Need to verify which specification actually produced the published tables and fix the notes.
2. **Appendix endline tech table displays midline index** – _d3_i_ml_domain3 used where _d3_i_el_domain3 should be (lines 1016, 1032, 1048). Main endline table is correct; only the disaggregated appendix table is affected.

MEDIUM:

3. Interaction model for heterogeneity testing omits _h main effect
4. Sample size varies across Table 4 columns (5,065-5,888) without comment
5. Differential attrition in Round 2 (p=0.061) without Lee bounds
6. 5-fold vs. 10-fold cross-validation discrepancy between code and text

```
The top priority is resolving F1 – determining whether the published tables are the lasso spec and updating notes accordingly.
```

```
* Sautéed for 4m 54s
```

0 critical, 2 high, 4 medium, 3 low. The core identification is correctly implemented. Remaining issues are presentation and robustness.

AI Audit vs. Real Referees: 7 of 10 in ~60 Seconds

Referee Criticism	AI?	Notes
High attrition & differential attrition	✓	Flagged as Critical; suggested Lee bounds
Results don't tell a coherent story	⦿	Caught ID problem, missed internal contradictions
Theoretical model not useful	✓	Flagged — but recommended <i>including</i> it
Short intervention duration (4–8 weeks)	✗	Requires field knowledge
Noisy learning measures (8-item phone test)	✓	Strong match; flagged IRT minimums
Paper too long and hard to follow	✓	Suggested cutting 20–30%
Multiple hypothesis testing issues	✓	Aggressive; added winner's curse framing
Limited external validity / COVID fatigue	⦿	Partial; missed publication landscape
Deviations from pre-analysis plan	✓	Caught the spirit, not the specifics
Complex randomization hard to interpret	✓	Strong match

5 full matches, 2 partial, 1 opposite recommendation, 2 misses.
11 referee reports across 8 journals over 4 years — vs. 60 seconds of AI.

What the AI Missed — And Why It Matters

✘ The Misses

- ▶ **Short duration:** 4–8 weeks is unusually short for an education RCT. Requires knowing field norms.
- ▶ **Internal contradictions:** Data arm increased tutoring *more* than info arm, but only info improved learning. Referees caught this; AI didn't.
- ▶ **COVID fatigue:** “We've seen too many of these papers.” AI can't know the publication landscape.

The opposite advice:

The AI recommended *including* the conceptual model. Every referee wanted it **cut**. Sometimes AI gives exactly the wrong recommendation because it lacks field conventions.

Pattern:

AI catches **statistical/econometric** issues reliably.

AI misses **field knowledge, within-paper logic checks, and publication strategy**.

Takeaway: A Very Good Pre-Submission Check

✓ What It Replaces

- ▶ The first 80% of what a careful internal seminar discussant would say
- ▶ Catches the systematic stuff: identification, power, MHT, attrition
- ▶ In one minute, not one month

What it doesn't replace:

- ▶ Field-specific intuition
- ▶ “Does this make sense given what we know about Bangladesh?”
- ▶ Publication strategy advice
- ▶ Colleagues who know your work

For the code audit: it found bugs I didn't find in two years. That alone justified the time.

For the paper audit: run it before you send a paper out. It's free. It catches the embarrassing stuff.

Caveat: These skills are built for applied micro / dev econ. But they're customizable — share your own referee reports, successful papers, or field-specific concerns and Claude adapts.

/econ-audit vs. /review-paper: Zero Overlap

/econ-audit found 6

- ▶ FE discrepancy in table notes
- ▶ Wrong index variable in appendix
- ▶ SES interaction omits main effect
- ▶ Varying N across Table 4 columns
- ▶ CV fold discrepancy (5 vs. 10)
- ▶ Missing flag typo (\$flags bug)

All invisible without reading Stata code.

/review-paper found 8

- ▶ Attrition severity for learning claim
- ▶ Marginal significance / $q = 1.000$
- ▶ Self-reported outcomes during intervention
- ▶ Mechanism chain is indirect
- ▶ Mother's income imbalance
- ▶ Causal language too strong
- ▶ Cost-effectiveness missing
- ▶ IRT reliability from short test

All require reading the paper as a referee would.

Zero overlap. The code audit verifies that the code does what the paper says. The paper review asks whether what the paper says is convincing. **Run both.**

Application 3

“Do Lee Bounds” — From Referee Comment to Results

Acting on the Audit

The Referee Said “Do Lee Bounds”

From the AI audit (and real referees):

“High attrition & differential attrition.
Flagged as Critical. Suggested Lee bounds.”

This is the single most common referee request on this paper.

The question: Can AI just... do it?

Write the Stata code for Lee (2009) bounds on the app treatment arm vs. control?



Lee Bounds

- ▶ Tightens treatment effect estimates under **differential attrition**
- ▶ Trims the “excess” group to equalize response rates
- ▶ Gives upper and lower bounds on the true effect
- ▶ Standard in development / labor RCTs

>_ LIVE DEMO

Running Lee Bounds

```
$ claude
```

```
I need Lee (2009) bounds for the app treatment arm vs. control. The outcome is the learning index. Use the endline survey flag to define attrition. Write a Stata do-file, run it, and show me the upper and lower bounds with confidence intervals.
```

→ From referee comment to results in one prompt.

The Code It Wrote

```
/* Lee (2009) Bounds – App (Robi) Treatment Arm vs. Pure Control
Outcome : Combined learning index (avg of standardised math + Bangla total)
Attrition: endline survey completion (endline_comp)
Unit    : one child per household (assessed child if endline completed)
----- */

clear all
set more off

include "${repo_root}/config/config_local.do"

* Install leebounds (Tauchmann) if not already present
cap which leebounds
if _rc ssc install leebounds, replace

* ----- 1. Load data -----
use "$domains", clear
drop if missing(treatment1)

* ----- 2. Restrict to App arm + Pure Control -----
*   _treatment_order == 0 → Info control (pure control)
*   _treatment_order == 4 → T2B: only Robi (app arm)
keep if inlist(_treatment_order, 0, 4)

gen treat_app = (_treatment_order == 4)
label var treat_app "App (Robi) treatment"

* ----- 3. One child per HH (unit of assessment) -----
*   Endline completers : keep the assessed child (_la_child_tag == 1)
*   Non-completers     : keep one child per HH (outcome unobserved either way)
gsort caseid _la_child_tag -endline_comp
by caseid: keep if _n == 1

* ----- 4. Build combined learning index -----
. ....
```

Installs leebounds if needed, loads data, restricts to App arm vs. control, builds the outcome, runs the bounds.

The Results

```
• All three ran successfully. Here are the results:
---
Lee (2009) Bounds: App (Robi) vs. Pure Control

Sample: 3,700 households (1,094 control, 1,094 app). One child per HH.
Attrition: Endline completion rate is 53.1% in control vs. 43.7% in the app arm – differential attrition of -9.6pp.
Trimming proportion: 18.1% (trimming from the lower-attrition control group to equalize response rates).

Results (bootstrap SEs, 1000 reps)



| Outcome                 | Lower bound | 95% CI           | Upper bound | 95% CI         |
|-------------------------|-------------|------------------|-------------|----------------|
| Combined learning index | -0.195      | [-0.293, -0.097] | 0.289       | [0.188, 0.391] |
| Math total (std)        | -0.234      | [-0.391, -0.077] | 0.336       | [0.217, 0.456] |
| Bangla total (std)      | -0.246      | [-0.394, -0.097] | 0.319       | [0.203, 0.434] |



Interpretation

The bounds span zero for all three outcomes. The lower bounds are negative and significant (around -0.2 SD), while the upper bounds are positive and significant (around +0.3 SD). The overall 95% CI for the combined index is [-0.270, 0.375]. This means we cannot sign the treatment effect under worst-case attrition assumptions – the differential attrition (app arm lost ~10pp more) is large enough to make the Lee bounds uninformative about the direction.

Notes

- The "combined learning index" is the simple average of grade-standardized math and Bangla total scores (each standardized to control mean=0, sd=1 within grade). There's no pre-existing combined index in the data – let me know if you had a different variable in mind.
- Sample restricted to _treatment_order == 0 (info control) and _treatment_order == 4 (Robi only). If "app arm" should include all Robi-containing arms (3, 4, 5), I can re-run.
- Unit is one child per HH: assessed child for completers, first child for non-completers.

Do-file: /Users/ebear/Dropbox/Github/bd-remote-ed/code/07_Dofiles_academic/Lee_bounds_app.do
• Baked for 10m 58s
```

Bounds span zero for all three outcomes. Differential attrition (−9.6pp) is large enough to make Lee bounds uninformative about the direction. **From referee comment to answer: ~10 minutes.**

Erkmen: Teaching & Applications

Lecture Builder, Research Brainstorm & Web Scraping

Application 3

Lecture Builder Skill — From Papers to a Lecture

Where We Left Off — The Skill Spec

```
/skill-creator
```

```
Create a skill called "lecture-builder" that converts academic documents into lecture materials. The skill should:
```

1. Read documents in their entirety (PDF, DOCX, web pages) -- break reading into parts to avoid excessive token usage. Include a Python parsing script.
2. Extract: key arguments, empirical findings, methodological details, discussion points, potential exam questions.
3. Generate two outputs:
 - a) Structured lecture notes (markdown) with section headers, key takeaways, discussion prompts
 - b) A polished slide deck (draw on \academic-beamer-deck or \pptx) with clean visuals, one idea per slide
4. Adapt tone for undergraduate vs. graduate audiences (configurable).

```
Save SKILL.md + supporting files to .claude/skills/lecture-builder/.
```

*Last week we wrote the spec. **Today we actually build the skill — then run it.***

Building the Skill: /skill-creator Walkthrough



Building the Skill: /skill-creator Walkthrough



`/skill-creator` **interviews you**. Be ready to answer:
name, audience, inputs, outputs, edge cases, where to save.

Anatomy of lecture-builder

📁 .claude/skills/lecture-builder/

SKILL.md playbook

parse_document.py PDF/DOCX reader

slide_template.tex UVM Beamer

notes_template.md notes scaffold

What SKILL.md enforces

- ▶ Chunked reading → token control
- ▶ Mandatory citations from source PDFs
- ▶ Undergrad vs. grad tone toggle
- ▶ Dual output: notes + slides
- ▶ Reuses UVM color palette

Anatomy of lecture-builder

📁 .claude/skills/lecture-builder/

SKILL.md playbook

parse_document.py PDF/DOCX reader

slide_template.tex UVM Beamer

notes_template.md notes scaffold

What SKILL.md enforces

- ▶ Chunked reading → token control
- ▶ Mandatory citations from source PDFs
- ▶ Undergrad vs. grad tone toggle
- ▶ Dual output: notes + slides
- ▶ Reuses UVM color palette

Skills with Python scripts > pure-markdown skills when tokens matter. The script reads; the LLM synthesizes.

The Input Folder



Why this mix?

- ▶ Heterogeneous sources: **policy** + **government** + **academic**
- ▶ Forces the skill to harmonize tone, terminology, and evidence quality
- ▶ Mirrors how you'd actually build a lecture

The Input Folder



Why this mix?

- ▶ Heterogeneous sources: **policy** + **government** + **academic**
- ▶ Forces the skill to harmonize tone, terminology, and evidence quality
- ▶ Mirrors how you'd actually build a lecture

Topic: The labor and health consequences of ICE enforcement. Connects directly to ongoing research at UVM.

The Execution Prompt

```
/lecture-builder
```

```
I have a folder ./inputs/ with: gov_thinktank_ice_arrests.pdf,  
ice_enforcement_report.pdf, aslim_paper.pdf.
```

```
Build a 50-minute undergraduate lecture on the labor and health  
consequences of ICE enforcement. Audience: junior/senior econ majors  
who have taken intermediate micro and one applied econometrics class.
```

```
Outputs to ./outputs/:
```

- 1) lecture_notes.md -- sectioned notes with 5 key takeaways,
4 discussion prompts, 3 candidate exam questions.
- 2) lecture_slides.tex -- Beamer deck (-20 slides, 16:9). Use
a clean academic palette -- default UVM (deep green / gold) or ask
me. In-text citations to all three source PDFs.

```
Be conservative with tokens: parse each PDF in chunks via  
parse_document.py, summarize per-section, then synthesize.
```

Expected Output (Live Demo Running)

▶ Live demo running in the background — final files appear by the end of this section

lecture_notes.md

```
# 1. The setting
Recent expansions of interior immigration
enforcement...
# 2. Identification strategy
287(g) rollout as a staggered DiD...
# 3. Findings
Aslim et al. document...
# Discussion prompts
1. Why might the effect on infant
health attenuate over time?
# Exam questions
1. Sketch the parallel-trends test...
```

synthesizes

lecture_slides.tex

```
% 20 slides, 16:9
% UVM palette
\begin{frame}{Setting}
  ICE arrests timeline
\end{frame}
\begin{frame}{Methods}
  staggered DiD
\end{frame}
\begin{frame}{Results}
  main effect
\end{frame}
```

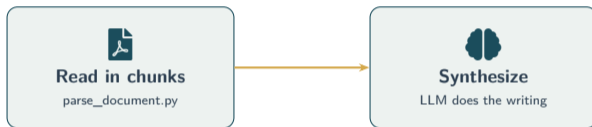
What Just Happened (Under the Hood)



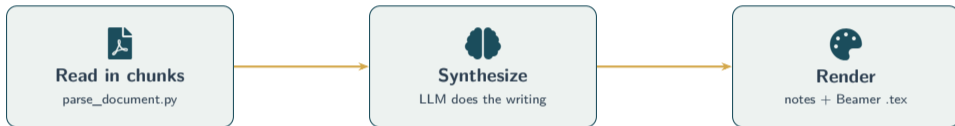
Read in chunks

`parse_document.py`

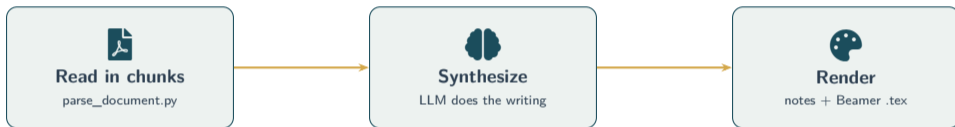
What Just Happened (Under the Hood)



What Just Happened (Under the Hood)



What Just Happened (Under the Hood)



This skill is reusable. Swap the input folder, swap the topic, swap the audience — get a new lecture in minutes. Build once, deploy forever.

Application 4

/research-brainstorm — Stress-Test a Research Idea

What the Research Brief Contains

Brief sections

- ▶ **The question** — structured: unit, population, treatment, outcome, comparison, setting
- ▶ **Closest related work** — 3–6 cites with sharp delta
- ▶ **Contribution type** — setting / identification / reframing / measurement / mechanism / policy
- ▶ **The strongest objection** — one paragraph, unvarnished
- ▶ **Alternative framings** — 2–3 pivots
- ▶ **Feasibility** — data, identification, power, timeline, fatal risks
- ▶ **Next concrete step**

Saved to:

`research_briefs/YYYY-MM-DD-slug.md`

Markdown so it lives next to your other notes.
Re-openable, editable, citeable in your own write-ups.

What the Research Brief Contains

Brief sections

- ▶ **The question** — structured: unit, population, treatment, outcome, comparison, setting
- ▶ **Closest related work** — 3–6 cites with sharp delta
- ▶ **Contribution type** — setting / identification / reframing / measurement / mechanism / policy
- ▶ **The strongest objection** — one paragraph, unvarnished
- ▶ **Alternative framings** — 2–3 pivots
- ▶ **Feasibility** — data, identification, power, timeline, fatal risks
- ▶ **Next concrete step**

Saved to:

`research_briefs/YYYY-MM-DD-slug.md`

Markdown so it lives next to your other notes.
Re-openable, editable, citeable in your own write-ups.

*The conversation is great.
The artifact is what you keep.*

Installing /research-brainstorm



Full economist pack

```
npx skills add  
  thinkingwithagents/skills
```

Installs the full pack in one go. Today we use: research-brainstorm, find-data, academic-beamer-deck.



One skill at a time

```
npx skills add  
  thinkingwithagents/skills  
  --skill research-brainstorm
```

Mix-and-match per project. Lighter footprint when you only need one.

Installing /research-brainstorm



Full economist pack

```
npx skills add  
  thinkingwithagents/skills
```

Installs the full pack in one go. Today we use: research-brainstorm, find-data, academic-beamer-deck.



One skill at a time

```
npx skills add  
  thinkingwithagents/skills  
  --skill research-brainstorm
```

Mix-and-match per project. Lighter footprint when you only need one.

Same `npx skills` mechanism works for any GitHub-hosted skill repo. **Build your own pack and share it with your coauthors.**

Live Demo: Hand Me a Question

Audience:

hand me a half-formed research question.

Live Demo: Hand Me a Question

Audience:

hand me a half-formed research question.

An effect
you suspect
but can't yet measure

A dataset
you have access to
but haven't used

A policy
you want
to evaluate

Live: invoke `/research-brainstorm`, ride through the phases, walk away with a brief.

Application 5

Web Scraper — From `/find-data` to a Live Visualization

/find-data Picks NBER Working Papers

/find-data in one line

We hand it our constraints (applied-micro audience, 24+ months, scrape-able, time-series + cross-section). It scans repositories and surfaces a source.

Today's pick: NBER working papers.

Why NBER: every economist cares; clean HTML; rich JEL + affiliation cross-section; ~1,500 papers / year.

URL:

`nber.org/papers?page=N&perPage=100&sortBy=public_date`

Extract: `nber_number, title, authors, public_date, primary_jel_code, affiliations.`

Goal: novel data on the screen in under 10 minutes. Skip the buildup — go straight to scraping.

Inspect Before You Prompt

The image shows a screenshot of a web page displaying a list of NBER Working Papers. The page title is "NBER Working Papers" followed by a URL parameter "sortBy=public_date". The list contains three entries, each with a paper ID, title, authors, and date. The first entry is "w32145 The Long Reach of... Aslim, Beam, et al." dated "Apr 2026". The second is "w32144 Identification in DiD... Roth, Sant'Anna" dated "Apr 2026". The third is "w32143 Childcare and Labor Supply Goldin, Olivetti" dated "Apr 2026". Below the list are vertical dots and a pagination bar showing "100 per page" and page numbers "1 2 3 ... 152". Three callout boxes on the right point to specific parts of the page: "Listing card title + date" points to the first entry, "Detail page authors + JEL" points to the authors of the second entry, and "Pagination ?page=N" points to the pagination bar.

NBER Working PaperssortBy=public_date

w32145	The Long Reach of . . . Aslim, Beam, et al.	Apr 2026
w32144	Identification in DiD. . . Roth, Sant'Anna	Apr 2026
w32143	Childcare and Labor Supply Goldin, Olivetti	Apr 2026
. . .		

< 1 2 3 ... 152 > 100 per page

Listing card title + date

Detail page authors + JEL

Pagination ?page=N

View-source first. The agent writes better scrapers when you've already named the selectors.

The Scraping Prompt

Scrape NBER working papers for a teaching demo. Build a self-contained

Python script in `./scrape_nber/` using `uv`:

1. Fetch the most recent ~1,000 papers from

`nber.org/papers?page=N&perPage=100&sortBy=public_date`

(paginate until 1,000 or cutoff 2024-01-01).

2. Extract: `nber_number`, `title`, `authors`, `public_date`, `primary`

`JEL code`, `affiliations` (from detail page if needed).

3. Save to `./scrape_nber/data/nber_papers.csv`.

4. Generate three PNGs to `./scrape_nber/figures/`:

- `papers_per_month.png` bars, last 24 months, UVM colors

- `top_jel_codes.png` top 15 primary JEL codes

- `top_institutions.png` top 10 affiliations

5. Print 5-line stdout summary.

Politeness: `real-browser User-Agent`; `sleep 0.5s` between fetches;

cache HTML to `./scrape_nber/cache/` so re-runs are free.

What Claude Does While We Keep Talking

🌀 Fetching pages...

Parsing...

Plotting...

```
$ uv run scrape_nber/main.py

[INFO] fetching page 1/10 ...ok (98 papers)

[INFO] fetching page 2/10 ...ok (100 papers)

[INFO] ...

[INFO] parsing JEL codes ...

[INFO] saved data/nber_papers.csv

[INFO] writing figures/...

== Summary ==

n_papers: 1,000

date range: 2024-04 to 2026-04

top JEL: J (Labor)

top affil: NBER, Harvard, MIT
```

The terminal is a TV. Let it run while you talk through what's coming.

What Claude Does While We Keep Talking

✓ Fetched 10 / 10 pages

⚙️ Parsing 1,000 records...

Plotting...

```
$ uv run scrape_nber/main.py
[INFO] fetching page 1/10 ...ok (98 papers)
[INFO] fetching page 2/10 ...ok (100 papers)
[INFO] ...
[INFO] parsing JEL codes ...
[INFO] saved data/nber_papers.csv
[INFO] writing figures/...

== Summary ==
n_papers: 1,000
date range: 2024-04 to 2026-04
top JEL: J (Labor)
top affil: NBER, Harvard, MIT
```

The terminal is a TV. Let it run while you talk through what's coming.

What Claude Does While We Keep Talking

✓ Fetched 10 / 10 pages

✓ Parsed 1,000 records

⚙️ Plotting 3 figures...

```
$ uv run scrape_nber/main.py
[INFO] fetching page 1/10 ...ok (98 papers)
[INFO] fetching page 2/10 ...ok (100 papers)
[INFO] ...
[INFO] parsing JEL codes ...
[INFO] saved data/nber_papers.csv
[INFO] writing figures/...

== Summary ==
n_papers: 1,000
date range: 2024-04 to 2026-04
top JEL: J (Labor)
top affil: NBER, Harvard, MIT
```

The terminal is a TV. Let it run while you talk through what's coming.

What Claude Does While We Keep Talking

✓ Fetched 10 / 10 pages

✓ Parsed 1,000 records

✓ Wrote 3 PNGs

```
$ uv run scrape_nber/main.py
[INFO] fetching page 1/10 ...ok (98 papers)
[INFO] fetching page 2/10 ...ok (100 papers)
[INFO] ...
[INFO] parsing JEL codes ...
[INFO] saved data/nber_papers.csv
[INFO] writing figures/...

== Summary ==
n_papers: 1,000
date range: 2024-04 to 2026-04
top JEL: J (Labor)
top affil: NBER, Harvard, MIT
```

The terminal is a TV. Let it run while you talk through what's coming.

What to Try First

Start Small, Check Each Step

What to Try First

#	Activity	Time	What You Get
1	Create a voice file	15 min	Your tone in every AI draft — reuse forever
2	Data pipeline (describe → propose → code)	30 min	Orientation on any dataset
3	Install /econ-audit, run on a paper	10 min	Pre-submission check for free
4	Try inbox triage with Gmail integration	30 min	Sorted, triaged, draft replies

All four use the same principle: **AI does the mechanical parts, you check each step.** Start with whichever one matches a task you already have.

What We Built Today

1

Code audit: **51 files to 8**, found a bug hiding for 2+ years

What We Built Today

1

Code audit: **51 files to 8**, found a bug hiding for 2+ years

2

Paper audit: AI matched **7 of 10** real referee criticisms in 60 seconds

What We Built Today

1

Code audit: **51 files to 8**, found a bug hiding for 2+ years

2

Paper audit: AI matched **7 of 10** real referee criticisms in 60 seconds

3

Lee bounds: from referee comment to **results in minutes**

What We Built Today

1

Code audit: **51 files to 8**, found a bug hiding for 2+ years

2

Paper audit: AI matched **7 of 10** real referee criticisms in 60 seconds

3

Lee bounds: from referee comment to **results in minutes**

4

Built and ran a real teaching skill — **lecture-builder**

What We Built Today

1

Code audit: **51 files to 8**, found a bug hiding for 2+ years

2

Paper audit: AI matched **7 of 10** real referee criticisms in 60 seconds

3

Lee bounds: from referee comment to **results in minutes**

4

Built and ran a real teaching skill — **lecture-builder**

5

From “what data?” to “here’s a chart” in **under 10 minutes**

What We Built Today

1

Code audit: **51 files to 8**, found a bug hiding for 2+ years

2

Paper audit: AI matched **7 of 10** real referee criticisms in 60 seconds

3

Lee bounds: from referee comment to **results in minutes**

4

Built and ran a real teaching skill — **lecture-builder**

5

From “what data?” to “here’s a chart” in **under 10 minutes**

Six workflows. Six artifacts on disk. **All reusable.**

Resources



Links

- ▶ **Bootcamp website:**
`thinkingwithagents.github.io`
- ▶ **Skills library:**
`skills.sh`
- ▶ **Econ skills:**
`github.com/thinkingwithagents/skills`
- ▶ **Claude Code docs:**
`docs.anthropic.com/en/docs/claude-code`



Install Commands

- Step 1: Install Node.js**
`nodejs.org` → click LTS
- Step 2: Install Claude Code**
`npm install -g @anthropic-ai/claude-code`
- Step 3: Install econ skills**
`npx skills add thinkingwithagents/skills`
- Step 4: Launch**
`claude`

Thinking With Agents

Online sessions coming this summer

Workshops, deep-dives, and Q&A
on agentic AI for academic research.

Stay tuned for dates and registration:

thinkingwithagents.github.io

