COBRA

Mannheim Center for Corporate Behavior and Regulation Analysis

MANNHEIM BUSINESS SCHOOL

Use Cases & Thoughts on AI Agents for Empirical Research

COBRA AI Clinic

Prof. Marcel Olbert · University of Mannheim & COBRA · May 25, 2026



Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

Overview of Mannheim COBRA AI Clinic - May 2026

- **Claude Code - full stack setup**: the You, Co-author, external colleague, RA (Paul Seidel)
- **Claude Code - automated empirical project**: prompt → data collection analysis → paper completion (Paul Seidel)
- **Risks when automating empirical research to AI agents**: cases and discussion (Luca Caprari)
- **Use Cases for AI Agents in Academia**: a Professor's perspective (Marcel Olbert)
  1. **Co-worker inside project**: setup agent → codes, produces output, prepares paper
  2. **Academic writing**: skill and prompt agent → writes/edits directly in paper, BUT YOU review with human edits!
  3. **Teaching/slides/projects**: give agent project context → convert intput into slides, exercises, notes, etc.
- **Discussion: Moving Frontier, Risks, Opportunities** (Marcel Olbert)

the Mindset:

- You just saw a **deep dive into Claude Code**, skills, and a live empirical example.
- This was not an outlier or niche case - BUT THE FRONTIER.
- To prove this, some insights from **my own workflow**:
  - **My setup** — general and project-specific
  - **Concrete use cases** — research, writing, teaching
  - **Where I think this is going** — for us as a field and for you as a *human* scholar

We are **empirical researchers**, not AI engineers of tech geeks.

The bar is simple: does it make my research and teaching better?
$\Rightarrow$ where do I want to be in 1 year, 5 years, 10 years?

Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

My setup — environment and memory

Environment

- **VS Code** — one home for code, LaTeX, and the agent
- **Claude Code** (Max plan) — agent in the terminal: reads, writes, runs
- **Wispr Flow** — voice-to-text, fast prompting while I think out loud

Context layers

- Global `CLAUDE.md` — who I am, conventions across all projects
- Project `CLAUDE.md` — structure, glossary, rules per project
- **Memories** — persistent, file-based; the agent learns over time
- **Skills** — reusable specialised behaviours (e.g. paper editing)

Setup compounds: every new project starts with what the agent already knows about me.

Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

Claude as a (very) skilled RA on an existing project

- **Setting:** mid-stage project — data built, draft written, referee report/comments
- Onboard via project `CLAUDE.md`: data sources, sample, identifier, identification strategy, file map
- What a **(very) skilled RA** does for me here:
  - Read the referee report and conclude **revision plan** (with me!)
  - Expands/revises **analyses** from existing scripts
  - **Improves and cleans** legacy code (your 2022 self thanks you)
  - Draft **extensions** — alternative specifications, additional samples, appendix material

Live: Onboarding into a current revision project — new/revised result and drafting.

Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

Co-writing the paper

- **Overleaf** → Dropbox — but edit locally in VS Code to avoid sync issues
- Agent edits directly in the `.tex` file — no copy-paste loop
- Custom **skill** for empirical-paper editing:
  - Knows target-journal styles (QJE / JAE / JAR / TAR / RFS)
  - **Preserves my voice**; fixes flow, clarity, hedging

⇒ **Massive efficiency gain** on **"compliance" sections**: institutional, appendix, proofreading, e.g., references, text vs tabulated output, consistency of language around the sample, etc.

- **Review every edit, further edit manually** — claims, framing: your must own!
- Give precise instruction: e.g., *lead each paragraph with a topic sentence*, *do not add interpretations to numerical results unless I state so.*

Live: Editing a section of a current paper with the skill.

Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

Teaching as a project

- Teaching slides and similar task in a **project folder** in VS Code with its own `CLAUDE.md`
- Treat each course as a **sub-project folder**, if specifics needed, with its own sub-level `CLAUDE.md`
- Slides in Beamer / LaTeX, exercises/supplemental materials as papers, versioned alongside the code/slides
- `CLAUDE.md`: e.g., reuse visual/format (can refer to a template or raw input provided), structure for slides
- Produce in seconds: **slide drafts and updates, exercises, exam variants, notes** / screen for errors/redunancies, inconcsistencies (knows the whole course and cross-references)

Live: Extending this very deck — or a course module: TAX 303.

Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

What makes this work

- **LaTeX + Python or R** — AI lives where the code lives. *Stata works well too*, just less idiomatic for the agent.
- **Reproducibility:** every figure, table, and number from a script (code in project folder). No black-box outputs from LLM memory.
- **Documentation, but not cumbersome!** `CLAUDE.md`, comments, README — the agent both *needs* them and *writes* them.
- **Writing still needs you.** Agents over-interpret, hedge, and over-explain. Your judgement is the value-add.
- **You must be in control.** To avoid complexity, and be able to defend all output in a seminar Q&A situation!

Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

Vibe coding — with eyes open

- **Vibe coding** = describe the outcome, let the agent write the code.
- Fast, often correct, very tempting.
- **But errors are silent and structural:**
  - Wrong sample boundary, dropped duplicates, mis-merged identifiers
  - Wrong cluster level, wrong fixed effects, leakage from outcome into a ”control”
- Real example from a current project: clean-looking output that hid a subtle data-merge error — only caught on a **manual sanity check**.

⇒ You are still the **senior author**: read the code, eyeball the data, sanity-check the numbers.

Three horizons

Short term

Now – 12 months:
upskill, or fall behind.

Quick wins on **existing** work — revisions, polishing, documentation.

⇒ Short window before AI-fluency is standard.

Medium term

Set up **new** projects **AI-native** from day 1: clean structure, `CLAUDE.md`, all scripts.

⇒ *what is my edge?* Think *big*! How does AI *amplify* it (not substitute it)?

Long term

When everyone has the tool: **ideas & rigour win.**

Non-standard data (proprietary, scraped, hand-collected) gets *more* valuable.

⇒ Key will be the **human element** — ideation, conferences, teaching. Your scholarly personality!

What this means for us

- the **quality bar is rising!** $\Rightarrow$ great, because we can focus on deeper thoughts and make execution efficient and ambitious.

- Skills worth **investing in now**:
  - **Coding hygiene** — project structure, version control, scripts not notebooks
  - **Writing** — still the bottleneck; AI is your editor, not your author
  - **Research taste** — which question, which data, which spec; *harder, not easier*

- The good news:
  - a flood of mediocre AI-assisted papers makes **thoughtful work stand out more**, not less.
  - **Interpersonal research communication** likely gains importance: disseminate your work, provide feedback, network and collaborate

Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

Outline

My Setup

Use Case I: Onboard Claude as a Skilled RA

Use Case II: Paper Editing (Co-Writing)

Use Case III: Teaching Slides & Materials

Takeaways

High-Level Thoughts

Guidelines (Accounting & Taxation)

Resources

Resources for going deeper

- **Paul Goldsmith-Pinkham — "LLMs and Empirical Work"** (June 2024)
  `paulgp.com/2024/06/24/llm_talk.html`
  Basic AI setup for practical tasks — VS Code, Latex, Stata.

- **Anthropic Academy** — `anthropic.com/learn`
  Free courses on Claude, prompting, agents, and the API — introductory to advanced.

- **Markus Academy — "Claude Code for Applied Economists"**
  `markusacademy.substack.com/p/claude-code-for-applied-economists`
  Paul Goldsmith-Pinkham in video sessions on using Claude Code.

- **Sant'Anna — "Claude Code: My Workflow"**
  `psantanna.com/claude-code-my-workflow`
  Thought-through workflow setup — project structure, reproducibility, error handling.

- **Current reflections by economists**
  `causalinf.substack.com/p/what-a-panel-of-economists-said-about`
  Panel of economists on what AI is doing to the production of research.

Thank you!

Reach me at marcel.olbert@uni-mannheim.de

or Marcel Olbert