

Proposal: AI-Assisted Research with Claude Code for Economists

Prepared for **University and Research-Center Audiences** Contact: **Aniket Panjwani**
(aniket.panjwani@gmail.com)

Executive Summary

This proposal outlines a three-session short course for PhD students, postdocs, faculty, and adjacent researchers to learn how Claude Code can support real research and teaching workflows. The structure is grounded in the [Claude Code Full Course](#) and AI Economist workshop materials.

Why This Training Now

Agentic coding tools are reshaping intellectual work, including academic research. But most existing trainings are still too generic, too developer-centric, or too disconnected from the workflows researchers care about. Dr. Aniket Panjwani is a Northwestern-trained economics Ph.D. who works daily with agentic coding tools and teaches research-first workflows through AI Economist, YouTube, and private trainings.

This course focuses on the parts of agentic coding that are most useful in practice for economics and adjacent research: data cleaning, replication work, analysis extension, code generation and debugging, and faster iteration on active projects.

Program Story and Structure

Each live session is paired with selected chapters from the free [Claude Code Full Course](#). This lets participants handle setup and basic orientation asynchronously while preserving live time for demonstrations, questions, and applied research workflows. The default orientation is Claude Code as the primary workspace for the workshop.

I recommend reserving part of each session for a volunteer to work with Aniket on a suitable non-sensitive task, such as restarting an analysis, responding to referee comments, or building a reusable figure or table Skill, while the group observes the decisions and verification. A prepared research example is the fallback.

Session 1: AI, Agentic Coding, and Practical Fluency with Claude Code (90 min)

Session 1 introduces the transition from AI as a chatbot to AI as an agent working across files, documents, websites, code, and project context. Participants learn the concepts and habits needed to begin useful work without prior software-engineering experience.

- See how economists are using agentic coding to draft empirical papers, replicate studies, build reusable research workflows, and run long coding projects
- Understand models, interfaces, context windows, project folders, permissions, plugins, and Skills
- Organize bounded workspaces with durable project memory, then work across Claude Code, local files, browser workflows, voice, and mobile

- Provide rough, high-context instructions while retaining human review and approval for consequential outputs
- See participant-centered workflows such as reverse-engineering an unfamiliar replication package, reviving a dormant project by reconstructing where the work stopped, mapping the relationships among a paper, code, slides, and referee reports, or producing a source-grounded seminar or teaching brief

Pre-watch before Session 1

- Selected Claude Code Full Course chapters covering the foundations needed for practical fluency in the primary interface
- Full chapter list appears in the Preparation and Prerequisites section below

Session 2: Skills, Git, and Safe Agentic Research Work (90 min)

Session 2 addresses a central consequence of agentic coding: agents can change many connected files much faster than a person could manually. This makes version control more important, but Claude Code also makes version control substantially easier to use.

- Use Git for visibility, named checkpoints, safer experimentation, focused review, and durable project memory, including the exact files and lines an agent changed
- Use branches and pull requests to protect the stable project and review only what changed
- Learn a small conceptual vocabulary, including commit, branch, pull request, merge, and rollback, while Claude Code handles the command syntax
- Walk through concrete Skills for meeting retrieval, structured AI Q&A, econometric standards, browser control, writing voice, and project memory, then see how instructions, examples, references, scripts, and checks make them repeatable
- Follow the complete loop: brainstorm, plan, work on a branch, inspect the diff, review, revise, verify, and commit

A participant demonstration might update a figure while tracing its effects across code, notes, paper text, and slides; turn referee reports into a reviewable revision plan; or convert a recurring research process into a reusable Skill.

Pre-watch before Session 2

- Selected Claude Code Full Course chapters covering skills, Git, and inspectable project work
- Full chapter list appears in the Preparation and Prerequisites section below

Session 3: Advanced Workflows, Claude Code with Codex, and AI for Research and Teaching (90 min)

Session 3 extends the foundation into advanced research, faculty, and institutional applications. Participants learn how to divide complex work across agents, choose between Claude Code and Codex, and preserve useful workflows as tools and models change.

- Use subagents for parallel source review, code inspection, verification, and specialized critique
- Understand where Claude Code and Codex have different strengths and how to use them together
- Structure complex work through brainstorming, planning, delegation, review, and revision

- Preserve reusable knowledge in project instructions, Skills, plans, tests, diffs, and review notes
- Turn repeated research or faculty judgment into workflows that can be shared

The session also includes a dedicated discussion of AI and teaching. AI allows students to produce advanced-looking work quickly, making traditional signals of effort and understanding less reliable. The central question is therefore not simply whether AI should be allowed, but what learning objective an assignment is meant to enforce.

- Consider when assessment should move into the classroom
- Consider when students should be allowed or expected to use AI, with a higher standard for the resulting work
- Evaluate judgment, understanding, process, and the ability to defend an AI-assisted result
- Discuss responsible AI use in theses, empirical projects, and research papers
- Explore faculty uses for course preparation, problem sets, feedback, explanation, and their own learning

Rather than prescribe one policy, the session gives faculty a framework for matching AI rules and assessment design to the intended learning outcome.

Pre-watch before Session 3

- Selected Claude Code Full Course chapters covering subagents, advanced workflows, and cross-tool use
- Full chapter list appears in the Preparation and Prerequisites section below

Preparation and Prerequisites

All participants should complete the software setup **before the first session** and the assigned pre-watch material **before each live day**.

Required Software Setup

- **Claude Code** as the primary workshop tool
- Windows participants should first install **WSL**, then install Claude Code inside it
- **Wispr Flow** for dictation and higher-context interaction
- **Git**
- **GitHub account** and **GitHub CLI** (gh)
- **VS Code**
- For Session 3: **Codex** if the audience wants a comparison or alternate workflow

Full source material: [Claude Code Full Course](#)

Pre-Watch Before Day 1

- **3:46-15:09** - What Claude Code is, first exercise, plans, usage limits, and installation
- **23:51-49:33** - Terminal basics, interface, context windows, auto-compact, voice mode, permissions, hooks, and safety
- **49:33-1:00:02** - CLAUDE.md and the five practical primitives of Claude Code

Pre-Watch Before Day 2

- **1:10:31-1:18:53** - Skills: what they are, why they help, and where they live
- **1:18:53-1:25:52** - Installing skills and the PDF skill
- **1:25:52-1:38:59** - Creating reusable skills
- **2:03:57-2:12:17** - Git and GitHub explained and set up

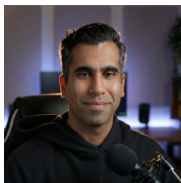
Pre-Watch Before Day 3

- **1:53:11-1:59:00** - Subagents
- **2:23:29-2:38:15** - Ideation, brainstorming, planning, and document review

Delivery Plan

- Format: Online via Zoom or in person
- Session count: 3
- Audience: PhD students, postdocs, faculty, and research staff
- Materials: story-led proposal PDF, selected pre-watch chapters, and workshop resource links

Trainer Profile



Dr. Aniket Panjwani

Economist, AI/ML engineer, and founder of AI Economist

Dr. Aniket Panjwani earned his PhD in Economics from Northwestern University. He helps economists use coding agents in real research and policy workflows and brings experience building production data and machine-learning systems.

- Led an in-person agentic-coding workshop and advisory engagement with **Opportunity Insights at Harvard University**, and delivered invited courses at the **Deutsche Bundesbank**, UCLouvain, the University of Calgary, and other universities
- Delivered one-on-one sessions with dozens of economists, including faculty at Harvard, MIT, Stanford, and the University of Chicago
- At **Early Warning Services (Zelle)**, built model-monitoring and data-quality systems for fraud models serving hundreds of financial institutions
- Publishes agentic-coding education for 9,500+ [YouTube](#) subscribers