

Lecture 27

From ChatGPT to Claude Code: AI Agents for Academic Research

Tyler Ransom

ECON 5253, University of Oklahoma

Today's plan

1. What you already know: ChatGPT
2. The evolution: AI agents and Claude Code
3. ChatGPT vs Claude Code: Key differences
4. Getting started with Claude Code
5. **Live demonstration (30 minutes)**
6. Other tools and best practices

ChatGPT: What you already know

Most of you are familiar with ChatGPT and similar chat interfaces:

- Web-based chat at chat.openai.com (and similar: Claude.ai, Gemini, etc.)
- You ask questions, it responds
- Great for: explaining concepts, brainstorming, quick coding help, editing text
- **Key limitations:**
 - No access to your files or codebase
 - Can't run code or tests
 - Each conversation starts fresh
 - Manual copy-paste workflow between AI and your project

The 2026 landscape: GPT-5.2, Claude Opus 4.6, Gemini 3 Pro, DeepSeek R1

- All share the same basic chat paradigm
- Freemium model; ~\$20/month for full features

The evolution: AI Agents

Moving beyond simple question-and-answer:

- **Traditional chat LLMs:** You ask, they answer (one-shot)
- **AI Agents:** Given a goal, they autonomously take steps to achieve it

Chat workflow:

- You describe task in detail
- AI gives you code/text
- You copy-paste into project
- You run it and check
- If broken, back to chat...
- Repeat until it works

Agent workflow:

- You describe the goal
- AI explores your project
- AI writes code in your files
- AI runs tests to verify
- AI iterates until successful
- You review final result

Agents use tools (read files, run code, git), iterate and reason, and verify their own work

What is Claude Code?

Claude Code - Anthropic's AI coding agent for research and development:

- **Not a chat interface** - it's a command-line tool (CLI) in your terminal
- **Works directly on your files** - reads, writes, edits your actual codebase
- **Operates autonomously** - you give it goals, it figures out the steps

Key capabilities:

- Explore and understand existing codebases
- Read, write, and edit multiple files
- Run code, tests, and terminal commands
- Create git commits and pull requests
- Install dependencies and fix errors
- Iterate on tasks until successful

Free tier available; \$20/month for Claude Pro (5x usage)

ChatGPT vs Claude Code: Side-by-side

Feature	ChatGPT	Claude Code
Interface	Web browser	Terminal/CLI
File access	None (copy-paste)	Direct access to your files
Code execution	Can't run code	Runs code, tests, commands
Workflow	One question at a time	Multi-step autonomous tasks
Context	Single conversation	Full codebase exploration
Best for	Quick questions, ideas	Complex projects, refactoring
Integration	Standalone	Works with git, npm, R, Python
Academic use	Brainstorm, explain, edit	Data analysis, code, automate

When to use ChatGPT vs Claude Code

Use ChatGPT when:

- Quick explanations
- Brainstorming ideas
- Text editing
- Single code snippets
- No file access needed

Use Claude Code when:

- Multi-file coding tasks
- Data cleaning pipelines
- Creating figures/tables
- Documentation generation
- Building replication packages
- **Running code in your environment**

Getting started with Claude Code

Installation (takes 2 minutes):

```
# On macOS/Linux:  
curl -fsSL https://cli.claude.ai/install.sh | sh  
claude auth login
```

Start using it:

```
cd ~/my-research-project  
claude
```

Example prompts:

```
analyze this codebase and explain what it does  
create publication-ready figures from my plots  
generate regression tables for all models  
organize messy scripts into logical folders
```

Why academics should care

AI agents are transforming research productivity:

- Joshua Gans had o1 write a paper **published** in *Economics Letters*
- Gans argues: research could become cheaper than search - why look up papers when AI can generate research on demand?

My own experience (October 2024):

- Created a working paper in health economics far outside my expertise ([link](#))
- Would not have been possible pre-2024
- Used: Elicit (lit review), GPT-4/Claude (coding), Claude (explanations, writing)
- **With Claude Code, this would have been even faster and easier**

Research tasks Claude Code excels at

Data and analysis:

- Clean and merge datasets
- Run specs and create tables
- Implement estimators
- Monte Carlo simulations
- Format conversion (Stata/R/Python)

Visualization:

- Publication-ready figures
- Coefficient plots
- Flowcharts

Documentation:

- Add code comments
- Create replication packages
- Generate data dictionaries
- Organize file structures

LaTeX and writing:

- Word → LaTeX conversion
- Beamer presentations
- Regression tables
- Debug compilation errors

Live Demonstration

From Messy Data to Publication-Ready Analysis

Using Claude Code for end-to-end research workflow

(30 minutes)

Other AI tools for academic research

Claude Code is powerful, but specialized tools exist:

Literature review:

- [Elicit.org](#) and [Consensus.app](#) - search and synthesize papers
- [NotebookLM](#) - create podcasts, slides, and summaries from PDFs

Code completion in your editor:

- GitHub Copilot - inline suggestions (free for academics!)
- Works in VS Code, RStudio; great for LaTeX/RMarkdown writing

Search and writing:

- [Perplexity](#) - AI search with citations
- [lex.page](#) - AI-augmented writing
- Microsoft Copilot - Word, PowerPoint integration

Recommendation: Use Claude Code for coding/analysis; combine with other tools for complete workflow

What I've used LLMs for in my research

- Fill out bureaucratic forms
- Write code that automates grading
- Write code that systematizes data analysis (for replication)
- Write code to create data visualizations
- Write unit tests of code
- Prepare discussion slides for conferences
- Prepare peer review reports
- Prepare slide decks (including this one!)
- Reduce word count of abstracts
- Improve sentence clarity in papers
- Write survey questions that methodologists would approve of
- Explain poorly written abstracts/papers in simpler terms
- Invert mathematical functions
- Debug LaTeX compilation errors
- Convert between data formats (Stata \leftrightarrow R \leftrightarrow Python)
- Create research design flowcharts
- ... not to mention personal life applications

Pro tips and best practices

Effective prompting:

- Be specific: ❌ "Fix my code" ✅ "Fix data import error in clean_data.R"
- Let it explore: "Explore project, then create regression tables"
- Multi-step: "Clean data, run analysis, create figures, organize outputs"

Documentation:

- Create CLAUDE.md files for future AI sessions
- Add comprehensive comments to scripts
- Generate READMEs with setup instructions

Best practices:

- Always verify output - agents make mistakes
- Use git/version control to revert changes
- Be careful with sensitive data
- Be transparent about AI use

Key takeaways

1. **ChatGPT → Claude Code:** Chat to autonomous agents
 2. **Different tools, different needs:** Chat for questions, agents for tasks
 3. **Works in your environment:** File access, runs code, uses git
 4. **Easy to start:** Install in 2 min, start small
1. **Transforms research:** Data, docs, viz, replication
 2. **Combine tools:** Elicit + Claude Code + Copilot
 3. **Verify everything:** AI accelerates, doesn't replace judgment

Bottom line: The question isn't whether to adopt AI agents, but how quickly.

Start: claude.ai/code

Staying on top of developments

The AI field moves fast - essential resources:

- **One Useful Thing** (Ethan Mollick) - Academic perspective, teaching/research applications
- **Marginal Revolution** (Cowen & Tabarrok) - Economics blog covering AI
- Anthropic blog for Claude updates
- GitHub blog for Copilot features

Questions?

Get started: claude.ai/code

Documentation: docs.anthropic.com/claude-code