AI with Code Execution

MGMT 675: Generative AI for Finance

Kerry Back

Why Code Execution Matters

- LLMs can write code, but writing is not the same as running
- Code execution enables:
  - Data analysis with real calculations
  - Visualizations and charts
  - File processing (Excel, CSV, PDF)
  - Iterative debugging—run, fix, repeat
- Transforms chatbots into computational tools

Three Approaches

ChatGPT

- Python on server
- 300+ libraries
- File upload/download
- Since 2023

Claude

- JavaScript in browser
- Python on server
- Interactive Artifacts
- Since late 2024

Local Execution

- AI writes code locally
- Full environment access
- Any language/library
- VS Code, Cursor, etc.

ChatGPT Code Interpreter

What is it?

- A Python environment that runs inside ChatGPT
- ChatGPT writes code, executes it, and returns results
- If code fails, it reads error messages and fixes them
- Sandboxed environment: secure but limited

Accessing Code Interpreter

- Requires ChatGPT Plus subscription ($20/month)
- Select GPT-4o from the model dropdown
- Code Interpreter is enabled automatically
- Upload files directly to the conversation
- As of March 2025, also available in o3-mini

ChatGPT: What You Can Do

Data Analysis

- Upload CSV, Excel files
- Statistical analysis
- Time series processing
- Create visualizations
- Download results

Finance Tasks

- Portfolio optimization
- Stock return analysis
- Risk metrics (VaR, Sharpe)
- DCF calculations
- Monte Carlo simulation

ChatGPT: Pre-installed Libraries

- **Data:** pandas, numpy, scipy
- **Visualization:** matplotlib, seaborn, plotly
- **Machine Learning:** scikit-learn, statsmodels
- **Files:** openpyxl, PyPDF2, Pillow
- 300+ packages total—no manual installation
- **Cannot install additional packages**

ChatGPT: Limitations

- **No internet access**—cannot fetch live data
- Maximum file upload: 100 MB
- Runtime limit: 120 seconds per execution
- Session state clears when environment resets
- Pre-installed packages only

Workaround: Download data first, then upload to ChatGPT

ChatGPT: Finance Example

Example Prompt

“I’m uploading a CSV with daily returns for AAPL, MSFT, and GOOGL. Please:

1. *Calculate annualized mean return and volatility for each stock*
2. *Compute the correlation matrix*
3. *Find the minimum variance portfolio weights*
4. *Plot the efficient frontier*

”

ChatGPT will write Python code, run it, show results, and let you download the chart.

Claude: Two Code Execution Features

Analysis Tool

- JavaScript in your browser
- Fast, lightweight
- CSV parsing (Papa Parse)
- Utility functions (Lodash)
- Creates artifacts

Code Execution Tool

- Python/Bash on server
- Full Ubuntu environment
- **Can install packages (pip)**
- 9GB RAM, 5GB disk
- No internet access

Claude Analysis Tool

- Runs JavaScript directly in your browser
- No server round-trip—instant execution
- Available on Claude.ai (free and Pro)
- Enable in Settings → Feature Preview → Analysis Tool
- Results feed into Artifacts for visualization

Claude Artifacts

What are Artifacts?

- Interactive content in a panel next to the chat
- Types: code, documents, visualizations, web apps
- View, edit, and iterate in real-time
- Export as files or share with others
- Can create React components, SVG graphics, charts

Claude: Visualization Power

- Analysis Tool + Artifacts = interactive dashboards
- Libraries available in Artifacts:
  - React for UI components
  - Recharts for charts and graphs
  - Tailwind CSS for styling
  - Three.js for 3D graphics
- Charts are interactive—hover, zoom, filter

Claude: Limitations

- Analysis Tool: JavaScript only (not Python)
- Limited libraries: Lodash, Papa Parse
- No direct file system access
- Uploaded files consume context window
- Code Execution Tool (Python): API only, in beta

Best for: Quick calculations, interactive visualizations, prototypes

Claude: Finance Example

Example Prompt

"Create an interactive artifact that:

1. *Lets me input expected returns for 3 stocks*
2. *Lets me input a covariance matrix*
3. *Calculates the tangency portfolio*
4. *Shows a chart of the efficient frontier*

"

Claude creates a React app with input fields, calculations, and a Recharts visualization—all in one artifact.

Running Code Locally

Why Local Execution?

- **Full environment access**—any library, any data source
- Internet access for APIs, databases, web scraping
- No file size limits or context window constraints
- Persistent files and project state
- AI can read, write, and execute code on your machine

Local Execution Tools

IDE Extensions

- **Claude Code**—terminal agent
- **VS Code + Claude/Copilot**
- **Cursor**—AI-native IDE
- **Google AntiGravity**
- AI writes and runs code in your environment

Notebooks

- **Google Colab**—Gemini built-in
- **Jupyter + AI assistants**
- **VS Code notebooks**
- AI can generate and execute cells
- See results inline

Local Execution: Advantages

- **No sandboxing**—access databases, APIs, local files
- **Install anything**—pip install, npm, system packages
- **Large datasets**—no upload limits
- **Persistent state**—pick up where you left off
- **Version control**—code saved in your repo
- **Production path**—code is ready to deploy

Local Execution: Considerations

- **Security**—AI can execute arbitrary code on your machine
- Requires local Python/development environment setup
- More setup than browser-based tools
- Best with permission controls (Claude Code prompts for approval)

Best for: Serious projects, production code, large datasets

Side-by-Side Comparison

Feature	ChatGPT	Claude	Local
Language	Python	JS / Python	Any
Execution	Server	Browser / Server	Your machine
Libraries	300+ pre-installed	Limited / pip	Unlimited
Install packages	No	Yes (server)	Yes
File access	Upload (100 MB)	Upload (context)	Full disk
Internet access	No	No	Yes
Output format	Files, charts	Artifacts, apps	Any
Cost	Plus ($20/mo)	Free tier avail.	Varies

When to Use Which?

ChatGPT

- Quick data analysis
- Statistical tests
- File conversion
- One-off tasks

Claude

- Interactive dashboards
- Quick prototypes
- React/web apps
- Shareable artifacts

Local

- Production code
- Large datasets
- API integrations
- Ongoing projects

Workflow: Data Analysis

ChatGPT Workflow

1. Upload CSV/Excel file
2. Describe analysis in plain English
3. ChatGPT writes and runs Python code
4. View output, ask follow-up questions
5. Download charts and results

Workflow: Interactive Tool

Claude Workflow

1. Describe the tool you want to build
2. Claude creates an Artifact (React app)
3. Interact with the tool in real-time
4. Ask Claude to modify or enhance it
5. Share or publish the artifact

Get Started: ChatGPT

1. Go to chat.openai.com and log in
2. Select GPT-4o from the model menu
3. Upload a CSV file with stock prices
4. Ask: "Calculate daily returns and plot them"
5. Ask: "What is the annualized volatility?"
6. Download the resulting chart

Get Started: Claude

1. Go to claude.ai and log in
2. Enable Analysis Tool in Feature Preview (if needed)
3. Ask: “Create an interactive compound interest calculator”
4. Try the artifact—enter values and see results
5. Ask Claude to add a chart showing growth over time
6. Publish or share the artifact

Finance Exercise: Portfolio Analysis

Try Both Tools

ChatGPT:

1. Download 1 year of daily prices for 5 stocks from Yahoo Finance
2. Upload to ChatGPT and ask for mean returns, volatilities, correlations
3. Ask for minimum variance portfolio weights

Claude:

1. Ask Claude to create an interactive mean-variance optimizer
2. Input the returns and covariance from ChatGPT
3. Explore different risk levels on the efficient frontier

Summary

- **ChatGPT:** Python sandbox for quick data analysis
- **Claude:** JavaScript (browser) or Python (server) + Artifacts
- **Local execution:** Full power—any library, any data, internet access
- Browser tools: convenient but sandboxed, no internet
- Local tools: more setup but production-ready
- Choose based on task complexity and data access needs