VS Code with Claude Code

MGMT 675: Generative AI for Finance

Kerry Back

What is VS Code?

- Visual Studio Code: a free code editor from Microsoft
- Works on Windows, Mac, and Linux
- Lightweight but powerful
- Huge ecosystem of extensions
- We'll use it as a UI for **Claude Code**

Why VS Code + Claude Code?

Colab

- Browser-based
- No installation
- Google Drive storage
- Google Gemini AI

VS Code

- Desktop application
- Local file access
- Claude Code AI
- More powerful tools

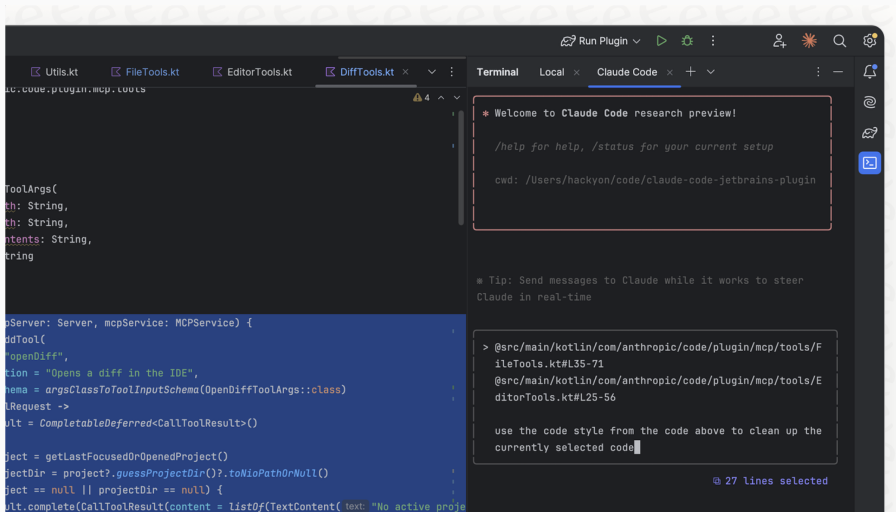Both support Jupyter notebooks!

What You Need

- VS Code installed (free from **code.visualstudio.com**)
- Claude Code extension (free to install)
- Anthropic account for Claude API access
- Python extension (for notebooks)
- Jupyter extension (for notebooks)

Installing the Claude Code Extension

1. Open VS Code
2. Press Ctrl+Shift+X (Windows) or Cmd+Shift+X (Mac)
3. Search for "Claude Code"
4. Click **Install** on the official Anthropic extension
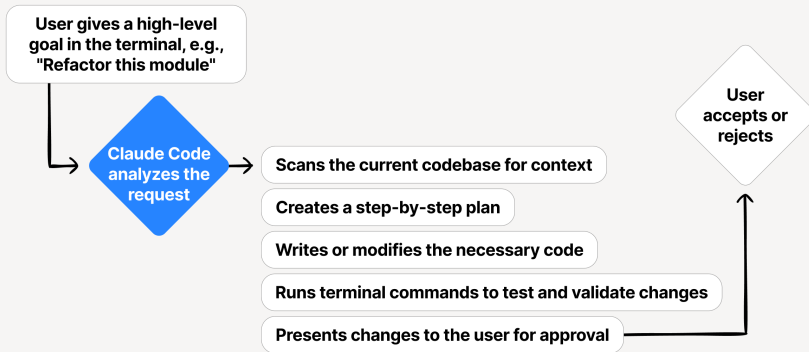5. Sign in when prompted

The Claude Code Interface

Opening Claude Code

- **Spark icon**: Click the spark icon in the top-right corner of any open file
- **Status bar**: Click "Claude Code" in the bottom-right corner
- **Command Palette**: Ctrl+Shift+P → type "Claude Code"
- **Keyboard shortcut**: Cmd+Esc (Mac) / Ctrl+Esc (Windows)

User gives a high-level goal in the terminal, e.g., "Refactor this module"

Claude Code analyzes the request

Scans the current codebase for context

Creates a step-by-step plan

Writes or modifies the necessary code

Runs terminal commands to test and validate changes

Presents changes to the user for approval

User accepts or rejects

www.eesel.ai

eesel

Chatting with Claude

- Type your question or request in the prompt box
- Press Enter to send
- Claude can see your selected code automatically
- Use @filename to reference specific files
- Claude asks permission before making changes

What Claude Code Can Do

- Explain code and answer questions
- Write new code from descriptions
- Fix errors and debug problems
- Edit files (with your approval)
- Run commands in the terminal
- Create and modify Jupyter notebooks

Reviewing Changes

- Claude shows changes in a side-by-side diff view
- Green = additions, Red = deletions
- You can **Accept** or **Reject** each change
- Or tell Claude what to do differently
- Changes are not applied until you approve them

Same concept as Colab, local execution

- Code cells and text cells
- Run cells with Shift+Enter
- Output appears below each cell
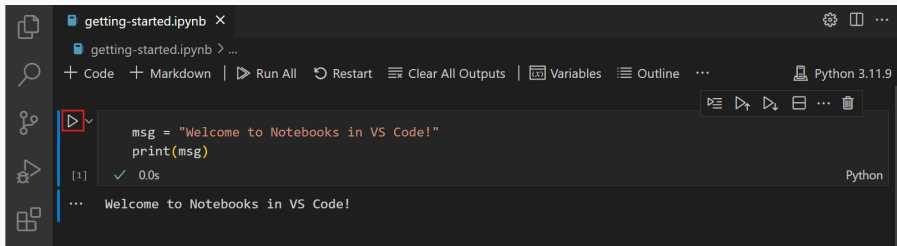- No browser or internet required

Creating a New Notebook

1. Ctrl+Shift+P (or Cmd+Shift+P on Mac)
2. Type "Create: New Jupyter Notebook"
3. Select a Python kernel from the top-right picker
4. Start coding!

Or simply create a new file with `.ipynb` extension

Opening Existing Notebooks

- File $\rightarrow$ Open File $\rightarrow$ select `.ipynb` file
- Drag and drop notebook file into VS Code
- Double-click notebook in the Explorer sidebar
- Download from Colab and open locally

The VS Code Notebook Interface

Running Cells

- **Shift+Enter**: Run cell and move to next
- **Ctrl+Enter**: Run cell and stay
- **Alt+Enter**: Run cell and insert new cell below
- Click the play button (▷) left of the cell
- "Run All" button in toolbar runs entire notebook

Notebook Keyboard Shortcuts

Running Code

- Shift+Enter: Run cell
- Ctrl+Enter: Run, stay
- Alt+Enter: Run, insert below

Managing Cells

- A: Add cell above
- B: Add cell below
- DD: Delete cell

Press **Esc** to enter command mode, **Enter** to edit

Selecting a Kernel

- The kernel runs your code (like Colab's runtime)
- Click the kernel picker in the top-right corner
- Select "Python Environments" $\rightarrow$ choose your Python
- Kernel must be selected before running code

17

Using Claude with Notebooks

- Ask Claude to create a notebook for you
- Claude can add, edit, or delete cells
- Select code and ask Claude to explain it
- Request data visualizations or analysis
- Claude can fix errors in your notebook code

Key Differences from Colab

Colab

- Browser-based
- Gemini AI built-in
- Auto-saves to Drive
- Free GPU access

VS Code

- Desktop app
- Claude Code AI
- Local file storage
- Uses your computer

Other AI Coding Tools

> You need to try them to understand the differences

- VS Code + Claude Code is one of several options
- Three other popular tools:
    - **Cursor**: AI-optimized editor (fork of VS Code)
    - **GitHub Copilot**: Extension for VS Code and other IDEs
    - **Google Antigravity**: Web-based editor (fork of VS Code)
- Each has different strengths and workflows
- Note: You can use Copilot **and** Claude Code together in VS Code

High-Level Comparison

VS Code + Claude Code

- Chat with AI in sidebar
- AI makes multi-file changes
- You approve each change
- Works in familiar VS Code

GitHub Copilot

- Extension for multiple IDEs
- Autocomplete as you type
- Tab to accept suggestions
- $10/month

Cursor

- Like VS Code but AI-native
- Tab to accept suggestions
- Fast, real-time
- $20/month

Google Antigravity

- Web-based (VS Code fork)
- Gemini-powered
- Free (preview)
- Still experimental

Which Should You Use?

They're designed for different tasks—not mutually exclusive

- **GitHub Copilot**: Best for autocomplete while typing
- **Cursor**: Best for fast iteration and staying in flow
- **Claude Code**: Best for complex multi-file changes
- **Antigravity**: Experimental, worth trying when it matures
- Many developers use Copilot + Claude Code together in VS Code

Quick Reference

- **Open Claude**: Click spark icon or Ctrl+Esc
- **New notebook**: Ctrl+Shift+P → "New Jupyter Notebook"
- **Run cell**: Shift+Enter
- **Select kernel**: Click picker in top-right
- **Reference file**: Type @filename in Claude prompt

Exercise: Aggregating Spreadsheets

- Download the zip file containing five regional sales spreadsheets
- Each file has similar but not identical column layouts
- Reconcile column names and aggregate into a single table

Download exercise2-aggregation.zip

Exercise: Estimating Betas

- Download the Excel file containing stock and market returns
- Estimate betas for each stock using regression
- Interpret the results

Download exercise4-betas.xlsx