RICE | BUSINESS
Jones Graduate School of Business

AI-Written Code in Jupyter Notebooks

MGMT 675: Generative AI for Finance

Kerry Back

A Different Approach

Code Environment + Chatbot

- ChatGPT and Claude: Chatbots with code execution added
- Google Colab: Code execution environment with chatbot added
- Colab started as Jupyter notebooks in the cloud (2017)
- Gemini was integrated into Colab later (2024)
- Philosophy: Write and run code first, use AI to assist

What is Google Colab?

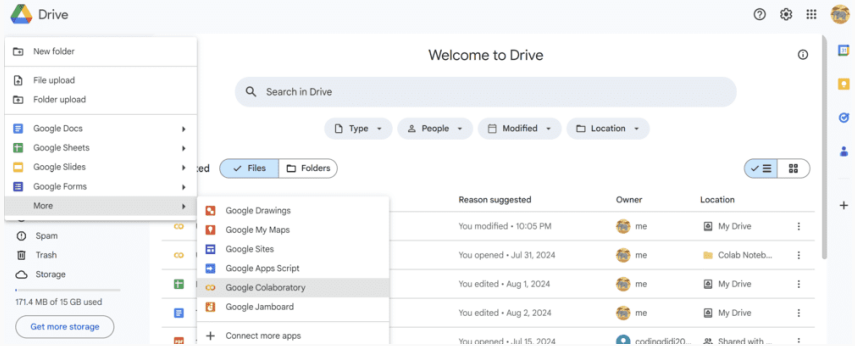
- A free tool from Google for running code in your browser
- No software installation required
- Works on any computer with internet access
- All your work saves automatically to Google Drive

What You Need

- Just two things:
- A Google account (Gmail works)
- A web browser (Chrome recommended)
- That's it!

Accessing Colab: From Google Drive

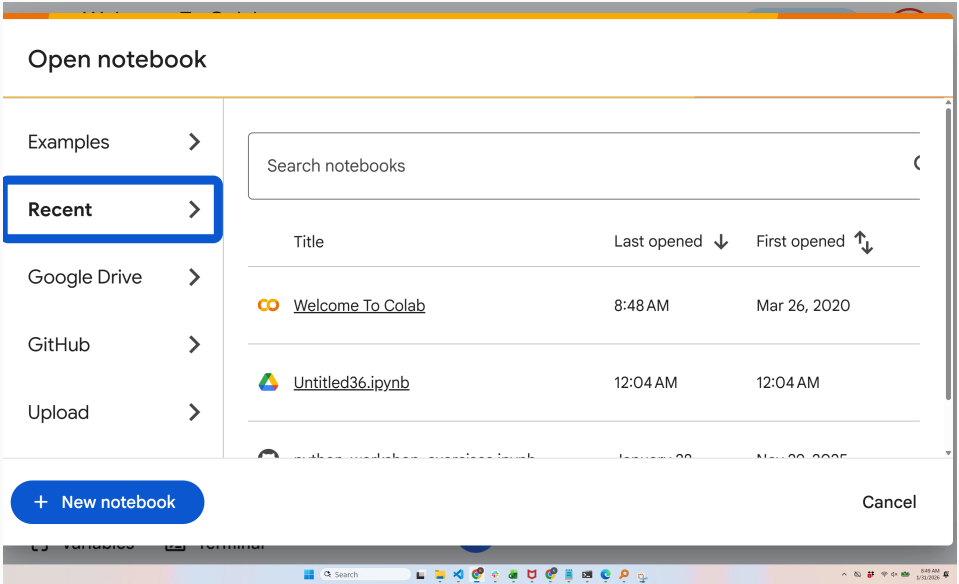
Click New → More → Google Colaboratory



Accessing Colab: Direct

colab.research.google.com

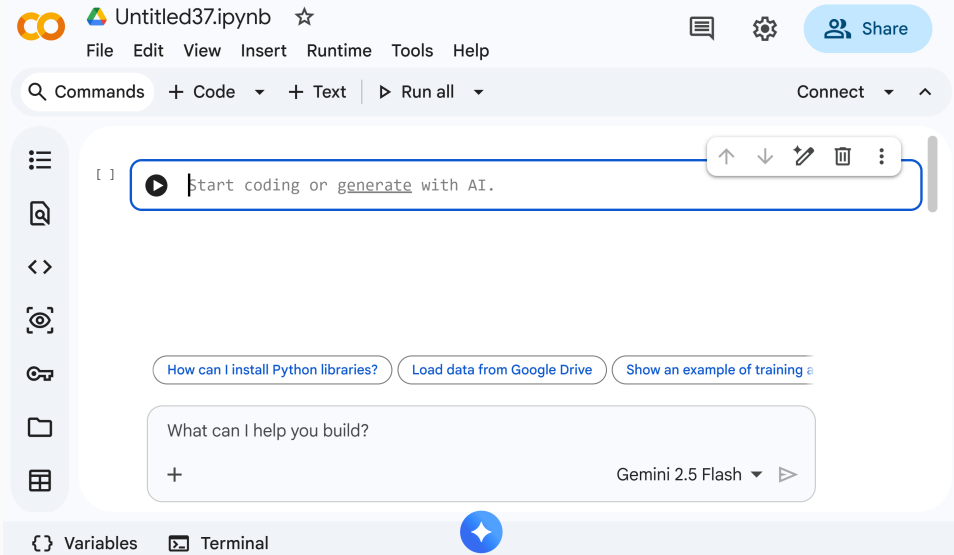
The Open Notebook Dialog



Opening Notebooks

- Examples: Google's tutorial notebooks
- Recent: Your recently opened notebooks
- Google Drive: Notebooks saved in your Drive
- GitHub: Open notebooks from GitHub repos
- Upload: Upload a .ipynb file
- Click + New notebook to start a fresh notebook

The Colab Interface: Notebook + Gemini



How Notebooks Work

- Three elements: notebook, notebook interface (Colab or other), and Python runtime environment
- A notebook (.ipynb file) is just a text file
- The interface renders the file to create what you see and handles communication with the runtime environment
    1. When you run a cell, the code is transmitted to a runtime environment (called a kernel).
    2. The runtime processes and executes your code.
    3. Results flow back to the notebook interface.
    4. The interface renders outputs, visualizations, and any error messages.

Navigating a Notebook

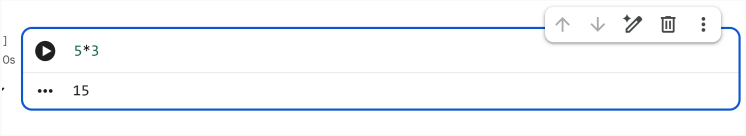
- + Code: Add a new code cell
- + Text: Add a text/markdown cell
- Connect: Connect to Google's servers
- Files (folder icon): View and upload files

What is a Cell?

- A cell is a box where you write code or text.
- Two types:
- Code cells: Run Python code
- Text cells: Write notes and explanations
- You can have as many cells as you need.

Your First Code: Simple Math

Type 5*3 and press Shift + Enter → Result: 15



Your First Code: Hello World

Type print('hello world') and press Shift + Enter

Running Code: Three Ways

- Click the play button (>) on the left of the cell
- Press Shift + Enter (runs and moves to next cell)
- Press Ctrl + Enter (runs and stays in cell)
- Tip: Shift + Enter is the most common method

Understanding the Play Button

- Before running:
- Circle with play icon (>) - Cell is ready
- While running:
- Spinning circle - Code is executing
- After running:
- Checkmark - Output appears below

Cell Numbers

- Notice the [1] or [2] next to cells?
- Shows the order cells were run
- Empty [ ] means not yet run
* means currently running
- Important: Can run cells in any order but top to bottom avoids confusion.

Adding New Cells

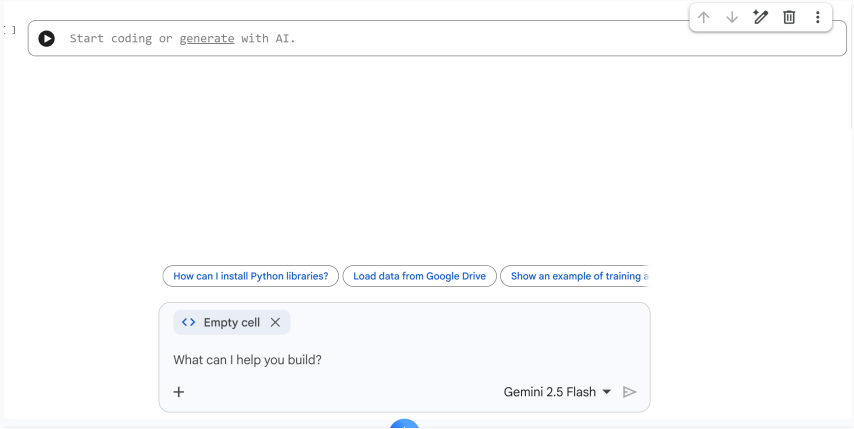
- From the toolbar:
- Click + Code for a code cell
- Click + Text for a text cell
- Using keyboard:
- Ctrl + M, B → Add cell below
- Ctrl + M, A → Add cell above

Deleting and Moving Cells

- To delete a cell:
- Click the trash icon in the cell toolbar
- Or: Ctrl + M, D
- To move a cell:
- Click the up/down arrows in the cell toolbar
- Or drag and drop the cell

Meet Gemini: Your AI Assistant

Gemini is built into Colab to help you write code



What Gemini Can Do

- Generate code from plain English descriptions
- Explain what existing code does
- Fix errors in your code
- Suggest improvements
- Answer Python questions

Runtime: What Powers Your Code

- When you click Connect, Colab gives you a virtual computer:
- CPU (standard processing)
- RAM (memory)
- Disk space
- And optionally: GPU or TPU for machine learning

Restarting the Runtime

If your code isn't working as expected:

Runtime → Restart runtime

This clears all variables and starts fresh.

Note: You'll need to re-run your cells after restarting

Session Limits

- Sessions disconnect after $\sim$90 minutes idle
- Maximum $\sim$12 hours continuous use
- Limited GPU/TPU hours per week

Get Started

1. Ask Gemini to get monthly GDP data from FRED using the `pandas-datareader` library.
2. Ask Gemini to plot GDP over time as a line chart.
3. Ask Gemini how you can save the chart.
4. Ask Gemini how you can save the GDP data as a CSV file.
5. Ask Gemini how you can save the notebook.

Reading Data Files in Colab

- Download the file to your **Google Drive** (e.g., to My Drive or a subfolder)
- In your Colab notebook, ask Gemini to **mount your Google Drive**
- Gemini will generate code like:
  - `from google.colab import drive`
  - `drive.mount('/content/drive')`
- Run the cell and authorize access when prompted
- Your Drive files are now at `/content/drive/MyDrive/...`
- Ask Gemini to load the file with pandas — it will use the correct path and reader (`read_excel`, `read_csv`, etc.)

When unsure about anything in Colab, ask Gemini.

Exercise: Computing Returns

- Download prices-dividends.xlsx to your Google Drive
- Ask Gemini to mount your Google Drive
- Ask Gemini to compute daily returns including dividends
- Ask Gemini to calculate annualized mean return and volatility

Exercise: Estimating Betas

- Download exercise10-betas.xlsx to your Google Drive
- Ask Gemini to mount your Google Drive
- Ask Gemini to estimate betas for each stock using regression
- Ask Gemini to interpret the results

Exercise: Pairs Trading Signals

- Download pairs-ko-pep.xlsx (or pairs-f-gm.xlsx) to your Google Drive
- Ask Gemini to mount your Google Drive and load the price data
- Ask Gemini to compute the price ratio of the two stocks and plot it over time
- Ask Gemini to compute the rolling 30-day z-score of the ratio and plot it, with horizontal lines at ± 1 and ± 2
- Ask Gemini to highlight dates where the z-score crosses ± 2 (potential entry signals) and where it returns to 0 (potential exit signals)