

Google Gemini in Colab

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?

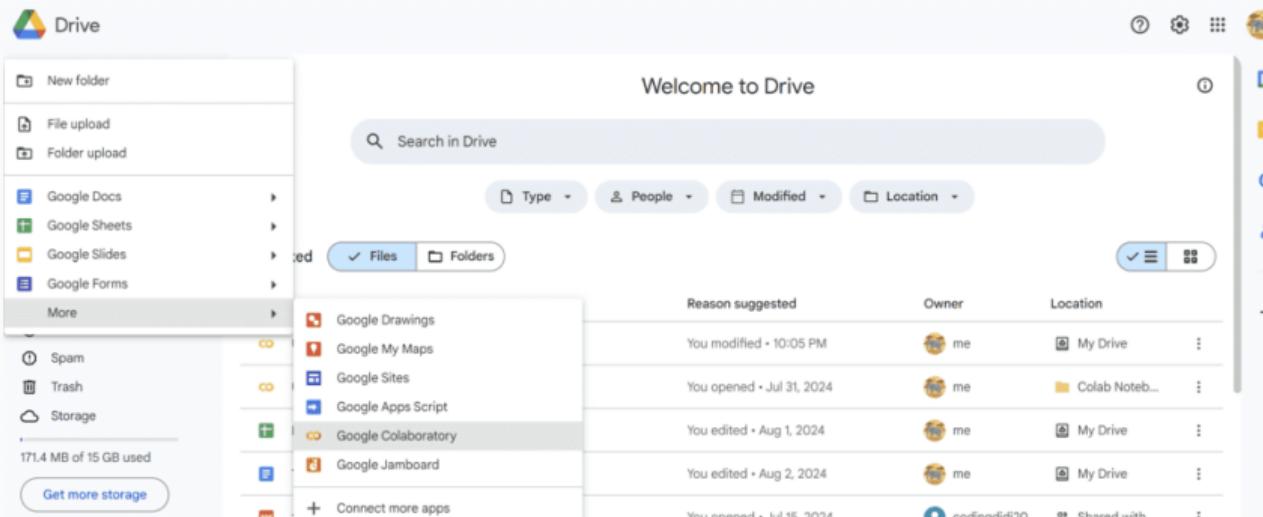
- A free tool from Google for running code in your browser
- No software installation required
- Works on any computer with internet access
- All your work saves automatically to Google Drive

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

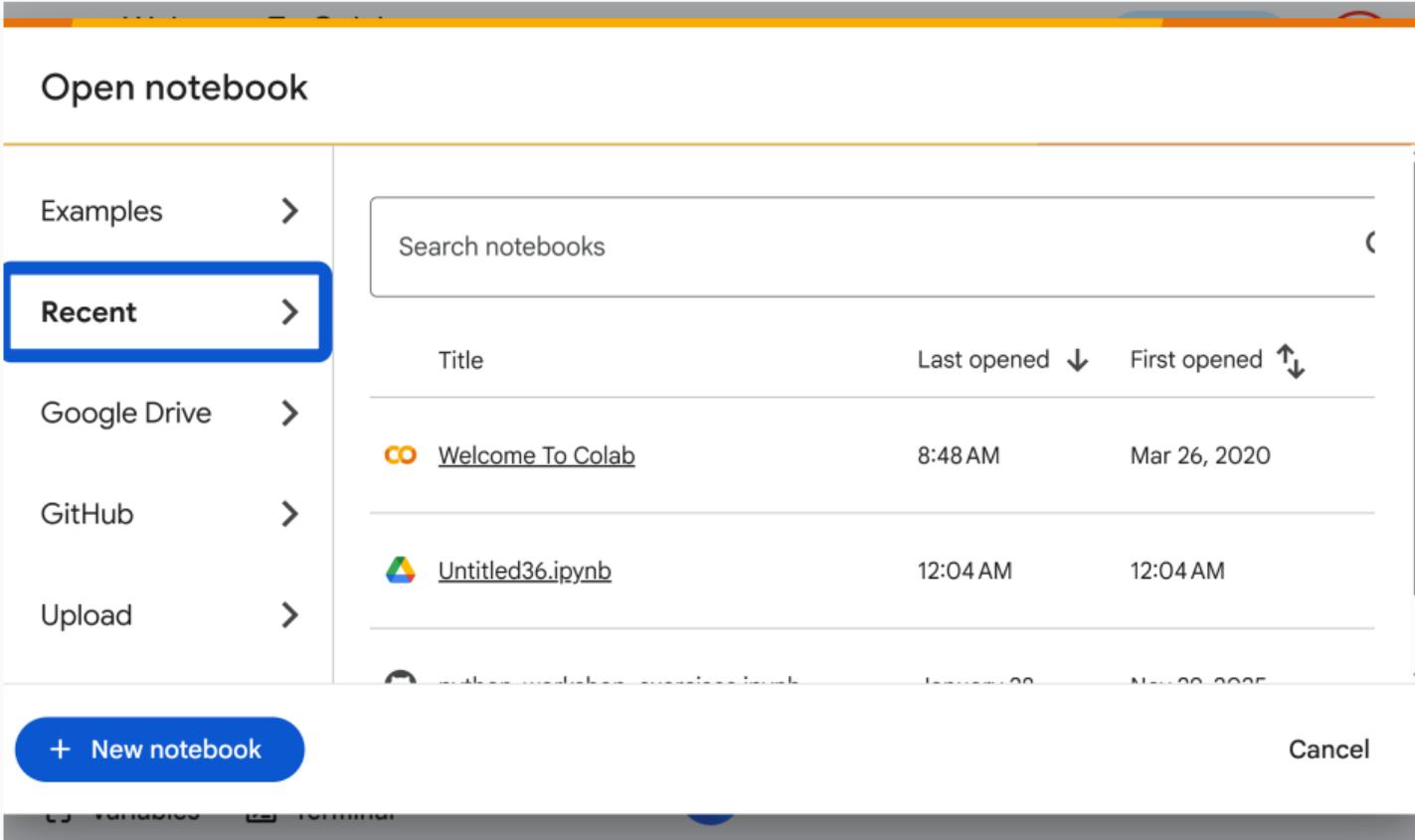


Accessing Colab: Direct

Go directly to:

colab.research.google.com

The Open Notebook Dialog



Open notebook

Examples >

Recent > Recent

Google Drive >

GitHub >

Upload >

Search notebooks

Title	Last opened	First opened
 Welcome To Colab	8:48 AM	Mar 26, 2020
 Untitled36.ipynb	12:04 AM	12:04 AM
...

+ New notebook

Cancel

Variables Terminal

Search

6

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

The image shows the Google Colab interface, which integrates the Notebook and Gemini features. The top navigation bar includes the Colab logo, the notebook title "Untitled37.ipynb", a star icon, and standard menu options: File, Edit, View, Insert, Runtime, Tools, and Help. To the right of the menu are icons for Help, Settings, and Share. Below the menu is a search bar with "Commands" and buttons for "Code" and "Text", followed by "Run all" and "Connect".

The main workspace features a sidebar on the left with icons for file operations (New, Open, Save, Find, Copy, Paste, Find and Replace, and a refresh icon). The central area contains a code cell with a play button and the placeholder text "Start coding or generate with AI." Below the cell are standard edit controls (up, down, edit, delete, and more). A large blue box highlights this cell.

At the bottom of the workspace, there are three buttons: "How can I install Python libraries?", "Load data from Google Drive", and "Show an example of training a".

The bottom right corner of the workspace contains the text "Gemini 2.5 Flash" with a dropdown arrow and a right-pointing arrow, indicating the AI integration feature.

At the very bottom of the interface are two navigation links: "Variables" and "Terminal".

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

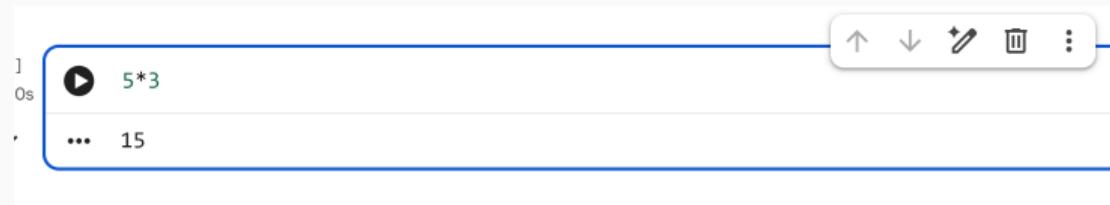
- + Code: Add a new code cell
- + Text: Add a text/markdown cell
- Connect: Connect to Google's servers
- Files (folder icon): View and upload files

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



A screenshot of a code editor interface. The code `5*3` is typed into the editor. The editor has a toolbar with icons for up, down, edit, delete, and more. The code is highlighted with a blue border. The output of the code, `15`, is shown below the code line. The entire code editor window is also highlighted with a blue border.

Your First Code: Hello World

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



The image shows a Jupyter Notebook cell with the following content:

```
[2] 0s
▶  print('hello world')
...  hello world
```

The cell has a status bar showing [2] and 0s. The code `print('hello world')` is preceded by a play button icon. The output `... hello world` is shown in a collapsed state, indicated by a downward arrow icon. To the right of the cell is a toolbar with icons for up, down, edit, delete, and more.

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

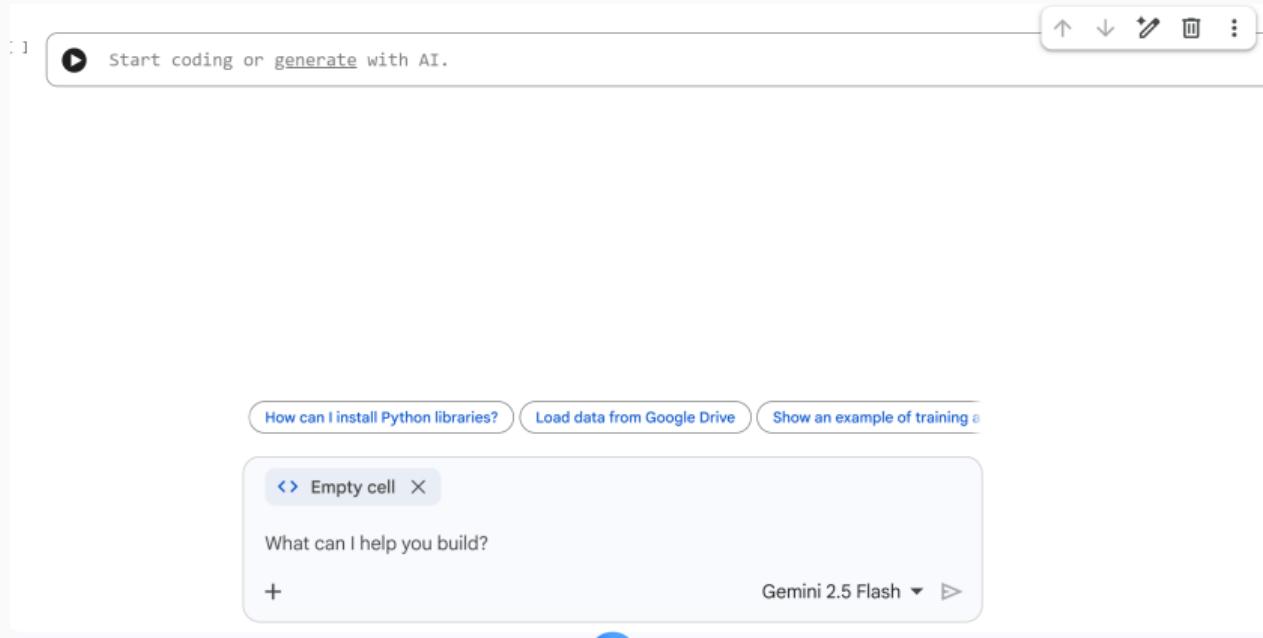
- From the toolbar:
- Click + Code for a code cell
- Click + Text for a text cell
- Using keyboard:
- $\text{Ctrl} + \text{M}, \text{B}$ → Add cell below
- $\text{Ctrl} + \text{M}, \text{A}$ → Add cell above

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 ~90 minutes idle
- Maximum ~12 hours continuous use
- Limited GPU/TPU hours per week

Get Started

1. Ask Gemini to get stock price data from Yahoo Finance and compute daily returns.
2. Ask Gemini to generate a boxplot of the daily returns.
3. Ask Gemini how you can save the boxplot.
4. Ask Gemini how you can save the return data.
5. Ask Gemini how you can save the notebook

Exercise: Computing Returns

- Download the Excel file containing NOV price and dividend history
- Compute daily returns including dividends
- Calculate annualized mean return and volatility

[Download returns.xlsx](#)

Exercise: Estimating Betas

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

[Download betas.xlsx](#)

Exercise: Mean-Variance Analysis

- Download the Excel file with expected returns and covariance matrix
- Find the tangency portfolio
- Plot the efficient frontier

[Download meanvariance.xlsx](#)