Google Colab

Ramiro Logares

Intro

- Google Colab, short for Google Colaboratory, is a cloud-based Jupyter Notebook environment provided by Google for free
- It allows users to create, edit, and share Python notebooks without the need to install any software on their local machines (runs bash and R too!)
- Colab is designed to facilitate collaboration, making it easy for multiple users to work on a project simultaneously
- It also provides access to powerful hardware, such as GPUs and TPUs, for accelerated computing.

Key Features

- Real-time collaboration: work with your peers simultaneously
- Version control: track changes and revert to previous versions
- Rich text and markdown support: create well-documented notebooks
- Integration with Google Drive: store and share notebooks easily

Getting Started

- Visit https://colab.research.google.com
- Sign in with your Google account
- Create a new Python notebook or open an existing one
- Start coding and running cells

colab.research.google.com/#scrollTo=Nma_JWh-W-IF

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share

+ Code + Text Copy to Drive

Connect

Table of contents

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

colab.research.google.com/#scrollTo=Nma_JWh-W-IF

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

New notebook
Open notebook ⌘/Ctrl+O
Upload notebook
Rename
Save a copy in Drive
Save a copy as a GitHub Gist
Save a copy in GitHub
Save ⌘/Ctrl+S
Revision history
Download
Print ⌘/Ctrl+P

Code + Text Copy to Drive

Share Connect

Welcome to Colab!

f you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command alette.

What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

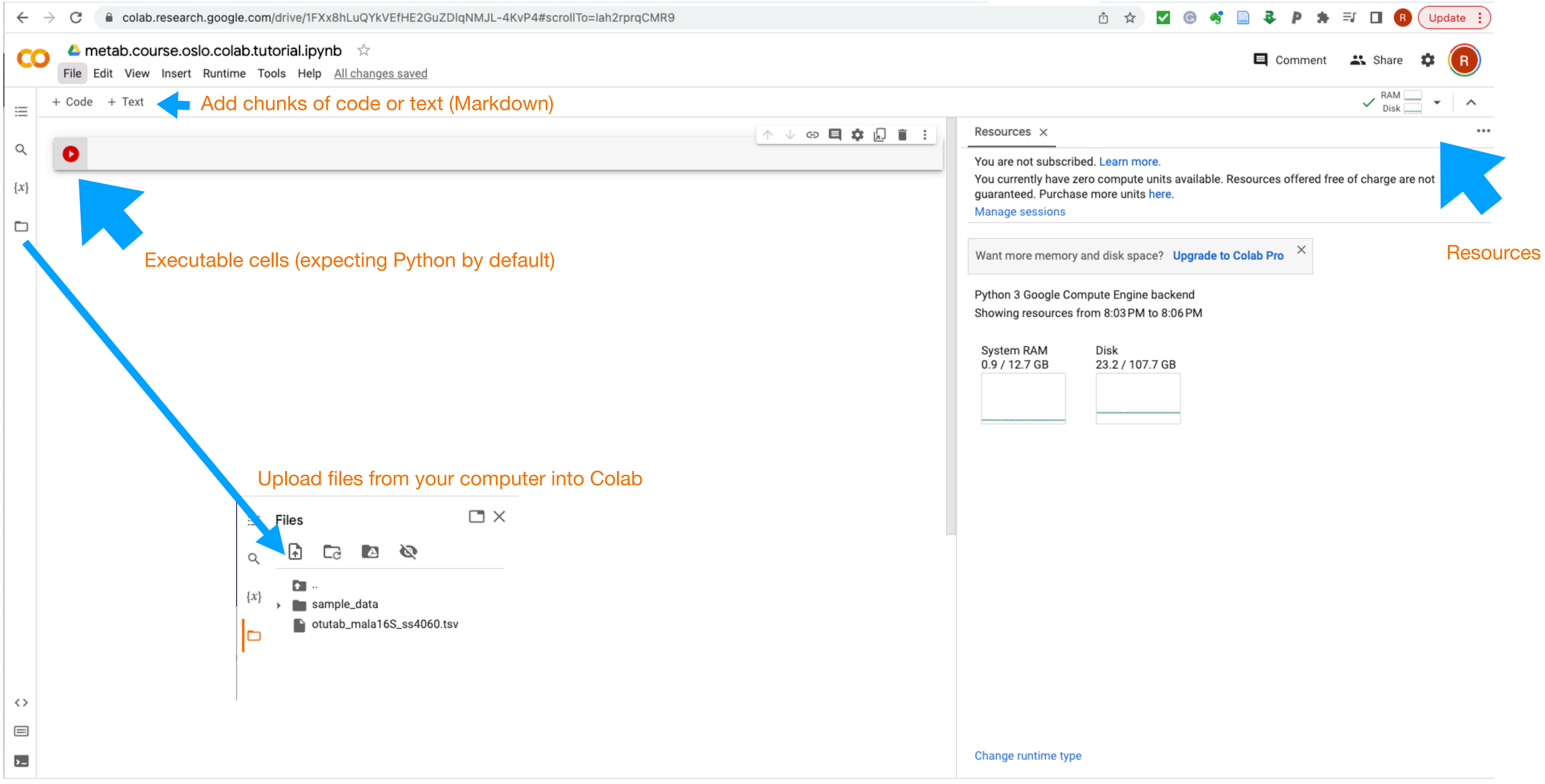
- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:


colab.research.google.com/drive/1FXx8hLuQYkVEfHE2GuZDlqNMJL-4KvP4#scrollTo=lah2rprqCMR9
Update
metab.course.oslo.colab.tutorial.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Add chunks of code or text (Markdown)
RAM
Disk
Resources
You are not subscribed. Learn more.
You currently have zero compute units available. Resources offered free of charge are not guaranteed. Purchase more units here.
Manage sessions
Want more memory and disk space? Upgrade to Colab Pro
Resources
Executable cells (expecting Python by default)
Python 3 Google Compute Engine backend
Showing resources from 8:03 PM to 8:06 PM
System RAM
0.9 / 12.7 GB
Disk
23.2 / 107.7 GB
Upload files from your computer into Colab
Files
..
sample_data
otutab_mala16S_ss4060.tsv
Change runtime type

Let’s run one example in Python

Testing google colab with Python

Step 1. Import python libraries (NumPy, Pandas, and Matplotlib)

(Press the "play" button Shift + Enter to run the cell and import the libraries)

NB: This notebook is only shared in "viewer" mode. To execute the code in the cells and make changes, follow these instructions:

To execute the code cells, make a copy of the notebook on your own Google Drive by clicking "File" > "Save a copy in Drive". This creates a separate copy of the notebook, which allows you to execute code cells and make changes without affecting the original notebook.

[5]
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Step 2. Generate sample data

[6]
```
# Set the seed for reproducibility
np.random.seed(42)

# Generate a Pandas DataFrame with random numbers
data = pd.DataFrame(np.random.randn(100, 4), columns=['A', 'B', 'C', 'D'])

# Display the first 10 rows of the DataFrame
data.head(10)
```

	A	B	C	D
0	0.496714	-0.138264	0.647689	1.523030
1	-0.234153	-0.234137	1.579213	0.767435

Step 3: Calculate summary statistics

```
[7] # Calculate summary statistics for the dataset
    summary = data.describe()

    # Display the summary statistics
    summary
```

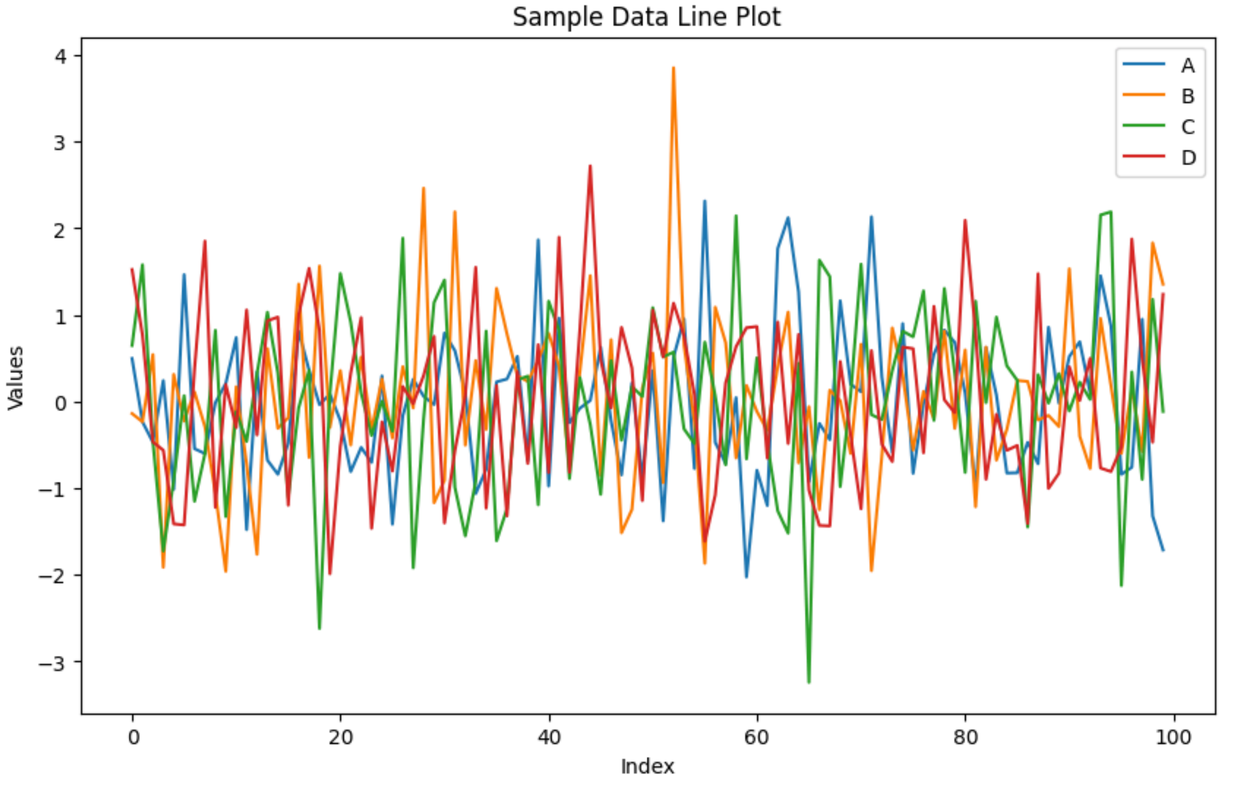
	A	B	C	D
count	100.000000	100.000000	100.000000	100.000000
mean	-0.009811	0.033746	0.022496	0.043764
std	0.868065	0.952234	1.044014	0.982240
min	-2.025143	-1.959670	-3.241267	-1.987569
25%	-0.716089	-0.564362	-0.616727	-0.727600
50%	-0.000248	-0.024646	0.068665	0.075219
75%	0.528231	0.547116	0.701519	0.778891
max	2.314659	3.852731	2.189803	2.720169

Step 4: Create a visualization

```
[8] # Plot the dataset as a line plot
    data.plot(kind='line', figsize=(10, 6))

    # Add title and labels
    plt.title('Sample Data Line Plot')
    plt.xlabel('Index')
    plt.ylabel('Values')

    # Display the plot
    plt.show()
```



Running Bash in Colab

To execute a Bash command in a code cell, prepend the command with an exclamation mark (!)

Step 1. list the files and directories in the current working directory

```
[9] !ls
```

```
sample_data
```

Let's check what remote machine we are running

```
[36] !uname -a
```

```
Linux f27d269d05b6 5.10.147+ #1 SMP Sat Dec 10 16:00:40 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
```

Step 2. Generate a sample directory

```
[10] !mkdir sample_directory
```

Step 3. Change de current working directory

```
[11] %cd sample_directory
```

```
/content/sample_directory
```

Note: We use %cd instead of !cd in this case because %cd changes the working directory for the entire notebook, while !cd would only change it for the specific cell.

Running Bash in Colab

Step 4. Let's check the working directory

```
[12] !pwd
```

```
/content/sample_directory
```

Step 5. We generate an empty file

```
[16] !touch sample_file.txt
     !ls
```

```
sample_file.txt
```

Step 6: Write content to the file

We use the echo command along with the > operator to write content to "sample_file.txt":

```
[17] !echo "This is the metab course of Oslo v2023!" > sample_file.txt
```

Step 7: Read the content of the file

Use the cat command to read and display the content of "sample_file.txt":

```
[18] !cat sample_file.txt
```

```
This is the metab course of Oslo v2023!
```

Running Bash in Colab

Step 8: Remove the file and directory

First, change the working directory to the parent directory:

```
[19] %cd ..
```

```
/content
```

Now, use the rm command to remove the "sample_file.txt" and "sample_directory":

```
[21] !rm -r sample_directory
     !ls
```

Uploading files from your computer into Colab

Step 1. let's import the necessary module from the google.colab package:

```
[22] from google.colab import files
```

Step 2: Use the upload() function to upload files

```
[23] uploaded = files.upload()
```

Choose files otutab_mal...S_ss4060.tsv

- **otutab_mala16S_ss4060.tsv**(text/tab-separated-values) - 1713307 bytes, last modified: 12/09/2021 - 100% done

```
Saving otutab_mala16S_ss4060.tsv to otutab_mala16S_ss4060.tsv
```

Click "Choose Files" and select one or more files from your computer to upload. The uploaded files will be saved in the current working directory

Uploading files from your computer into Colab

Step 3: Verify the uploaded files

Use the !ls command to list the files in the current working directory:

```
[29] !ls -lh
```

```
total 1.7M
-rw-r--r-- 1 root root 1.7M Apr  8 19:04 otutab_mala16S_ss4060.tsv
drwxr-xr-x 1 root root 4.0K Apr  6 13:39 sample_data
```

Lets check how the data looks like

```
[30] !head otutab_mala16S_ss4060.tsv
```

```
OTU_1    OTU_10   OTU_100 OTU_1000         OTU_10001        OTU_10002        OTU_10003
ST_1_MD28         234     0        0        0        0        0        5        0
ST_2_MD40         385     0        0        0        0        0        1        0
ST_3_MD52         1829    0        0        0        0        0        4        0
ST_4_MD60         1897    0        0        0        0        0        1        0
ST_7_MD98         622     1        0        0        0        0        0        0
ST_9_MD111        2064    1        0        0        0        0        0        0
ST_10_MD141       2428    0        0        0        0        0        0        0
ST_12_MD196       1714    0        0        0        0        0        1        0
ST_13_MD202       1478    0        0        0        0        0        0        0
```