

## **LAB No. 1**

# **Introduction to VS Code and Google Colab for Data Analysis Using Python**

### **LAB Description**

In this lab, students will learn how to work with Python programming environments using Visual Studio Code (VS Code) and Google Colab. VS Code is a lightweight and powerful source-code editor that runs on a local system and supports Python development through extensions. It allows users to write, run, and debug Python programs efficiently on their own computer.

Google Colab is a cloud-based Python notebook environment provided by Google that runs in a web browser. It does not require any local installation and provides free access to computing resources. Colab is especially useful for data analysis and visualization, as it supports interactive notebooks, built-in libraries, and easy file uploads.

Using either VS Code or Google Colab, students will create a dataset in Python, upload or load the data, and perform basic data analysis operations. The lab focuses on understanding how datasets are handled, how simple statistics are calculated, and how graphical representations help in interpreting data.

### **Lab Objective**

The objectives of this lab are:

- To understand the basic functioning of VS Code and Google Colab
- To learn how to create and upload a dataset using Python
- To perform basic statistical analysis (such as mean, median, and count)
- To visualize data using simple graphs (line charts, bar charts, or histograms)
- To develop foundational skills in data analysis and visualization using Python

## **How to use python in VS code?**

### **1. Create environment named 'venv'**

python -m venv

### **2. Activate environment**

venv\Scripts\activate

### **3. Select Environment in VS Code** After creating the environment:

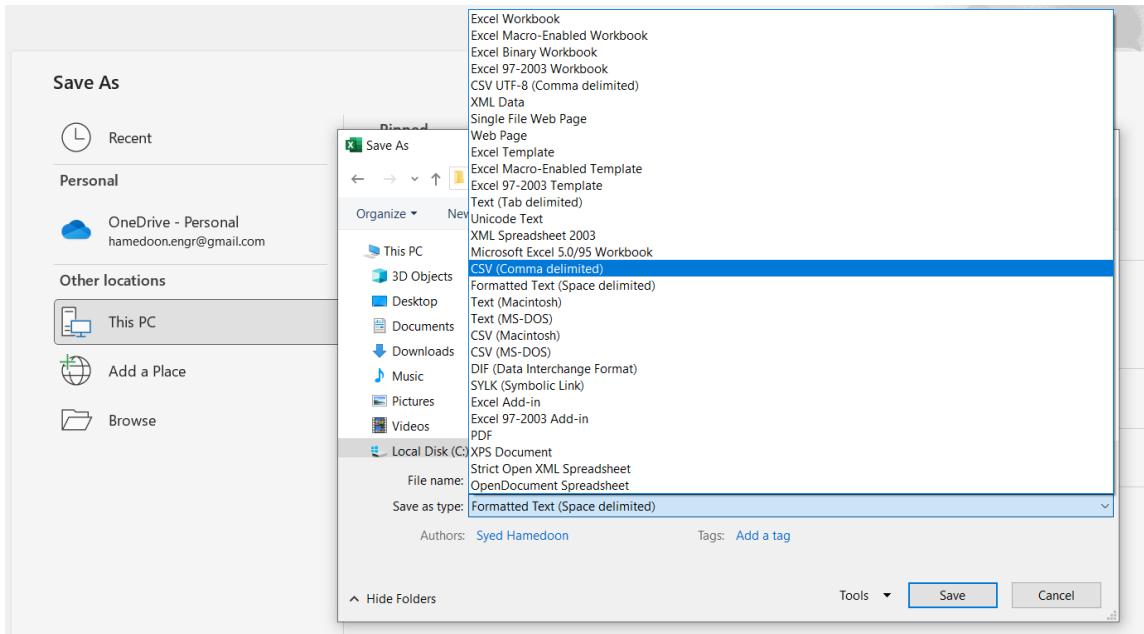
- 1) Open your project folder in VS Code.
- 2) Press Ctrl + Shift + P → search for Python:
- 3) Select Interpreter.
- 4) Choose your newly created environment (venv or myenv).

### **4. Install Machine Learning and Deep Learning Libraries**

- pip install –upgrade pip
- pip install pandas
- pip install matplotlib
- pip install seaborn
- pip install scikit-learn
- pip install scipy
- pip install numpy
- pip install xgboost
- pip install lightgbm
- pip install catboost
- pip install tensorflow
- pip install keras
- pip install torch

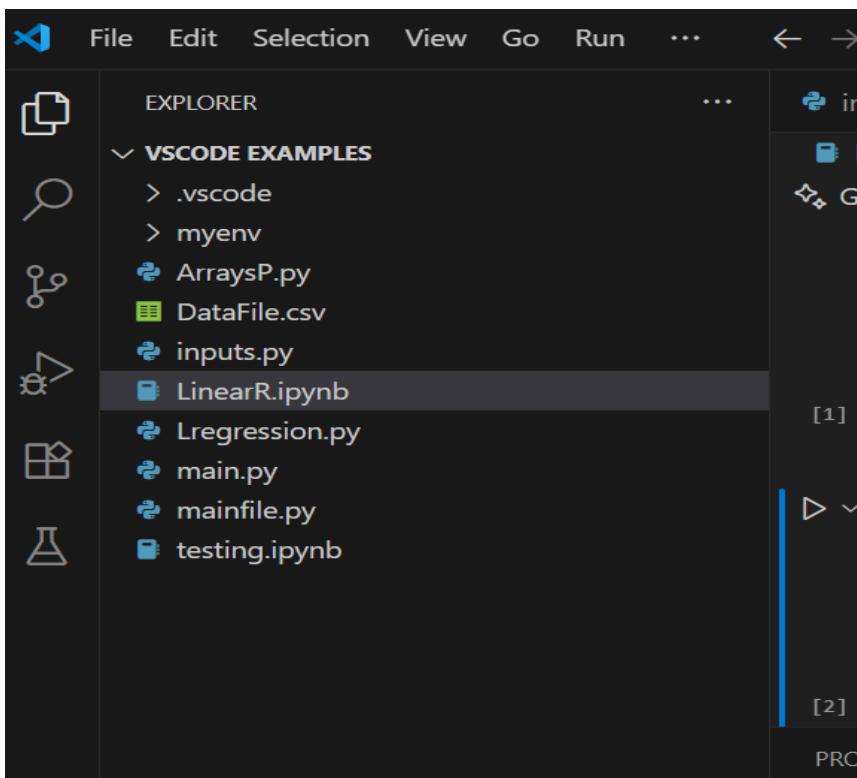
## **Task 1:**

### **Save Excel file as .csv (Comma delimited)**



## Task 2:

**Save .csv file in directory and folder where we created a python environment**



**Here we can see our file successfully**

The screenshot shows a Jupyter Notebook interface with several tabs at the top: inputs.py, Lregression.py, LinearR.ipynb, Untitled-1.ipynb (selected), and test. Below the tabs are buttons for Generate, Code, Markdown, Run All, Restart, Clear All Outputs, and Out. The notebook contains three code cells:

- [2] `import pandas as pd` ✓ 0.7s
- [3] `data=pd.read_csv("./DataFile.csv")` ✓ 0.1s
- [4] `data.head()` ✓ 0.1s

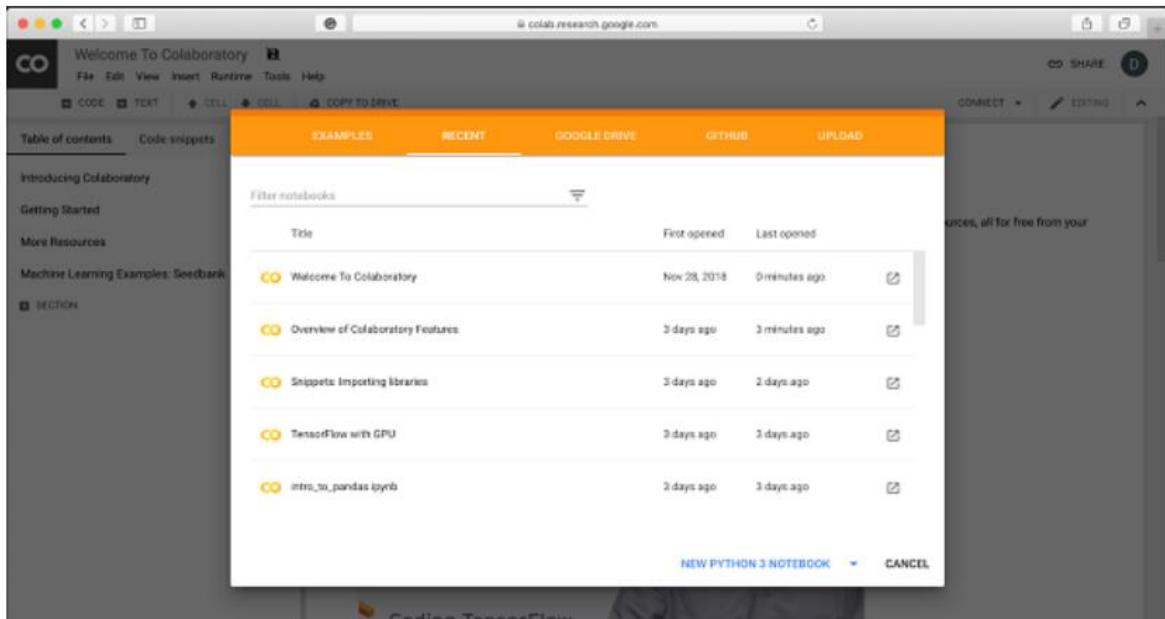
The output of the third cell is a table:

Sr No	SAP_ID	Name	Subject	University
0	1	MUHAMMAD HASSAN MADNI	AI	UOL
1	2	QAZI ZARYAB SAJJAD	AI	UOL
2	3	MOHTISHIM FAREED	AI	UOL
3	4	SHIZA ISHAQ	AI	UOL
4	5	MALAIKAH KHALID	AI	UOL

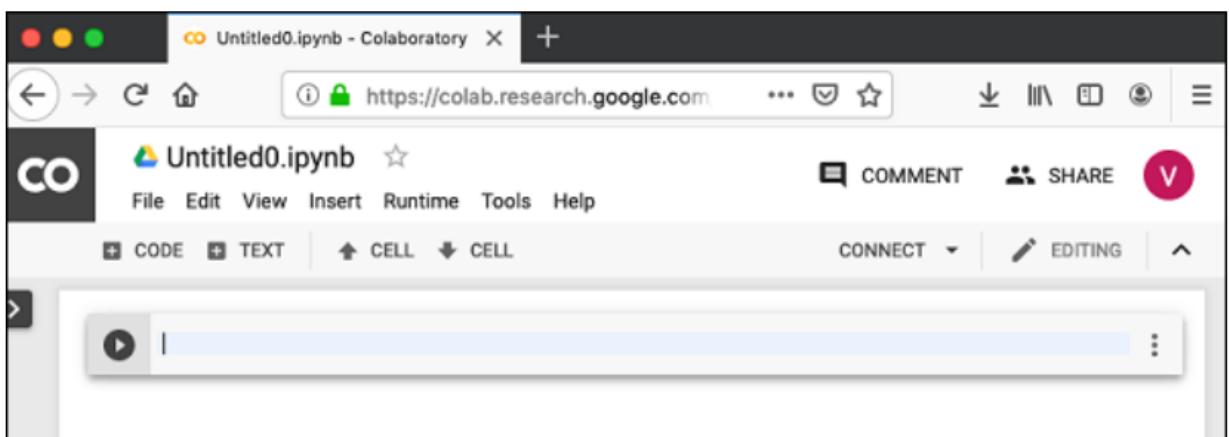
## Introduction of Google Colab

- Colab is a free notebook environment that runs entirely in the cloud. It lets you and your team members edit documents, the way you work with Google Docs.
- Colab supports many popular machine learning libraries which can be easily loaded in your notebook.
- Another attractive feature that Google offers to the developers is the use of GPU. Colab supports GPU and it is totally free. The reasons for making it free for public could be to make its software a standard in the academics for teaching machine learning and data science.
- Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook

- **Step 1** – Open the following URL in your browser  
– <https://colab.research.google.com> Your browser would display the following screen (assuming that you are logged into your Google Drive) –

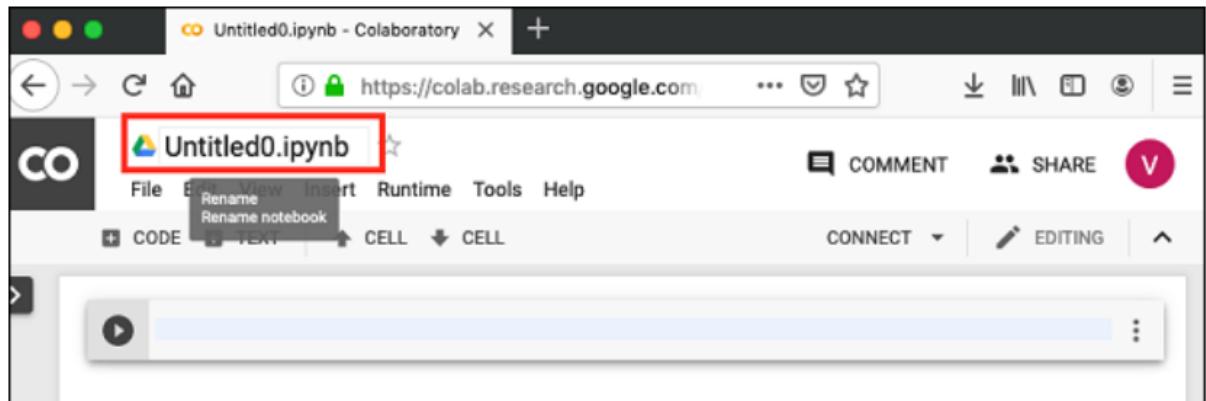


- **Step 2** – Click on the **NEW PYTHON 3 NOTEBOOK** link at the bottom of the screen. A new notebook would open up as shown in the screen below.

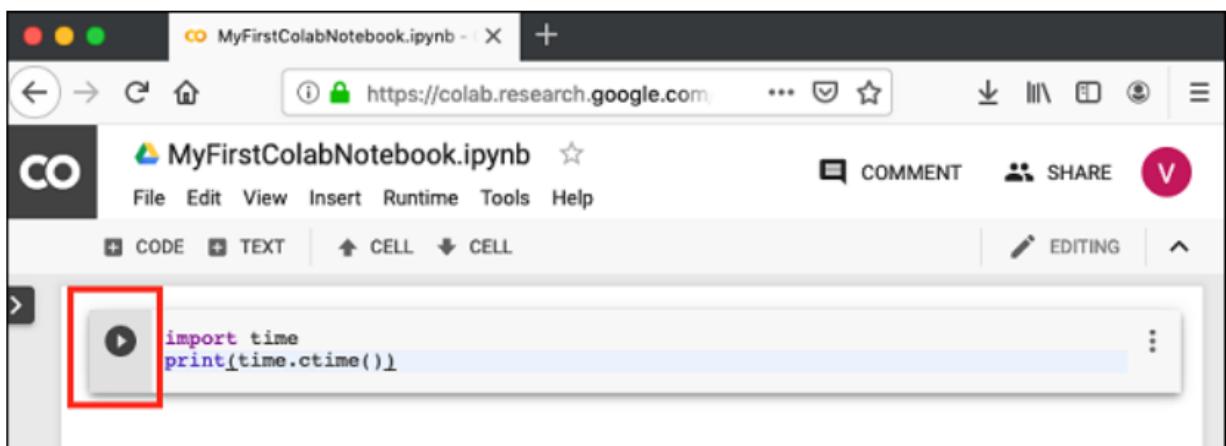


- **Setting Notebook Name**

By default, the notebook uses the naming convention UntitledXX.ipynb. To rename the notebook, click on this name and type in the desired name in the edit box as shown here –



- Entering Code



- Adding Code Cells
- To add more code to your notebook, select the following **menu** options –

A screenshot of a Google Colab notebook titled "MyFirstColabNotebook.ipynb". The notebook interface includes a toolbar with file operations, a menu bar with File, Edit, View, Insert, Runtime, Tools, and Help, and a toolbar below the menu with CODE, TEXT, CELL, CONNECT, and EDITING buttons. A code cell contains the following Python code:

```
import time
print(time.ctime())
```

The "CODE" tab in the cell toolbar is highlighted with a red box.

- **Run All**

To run the entire code in your notebook without an interruption, execute the following menu options –

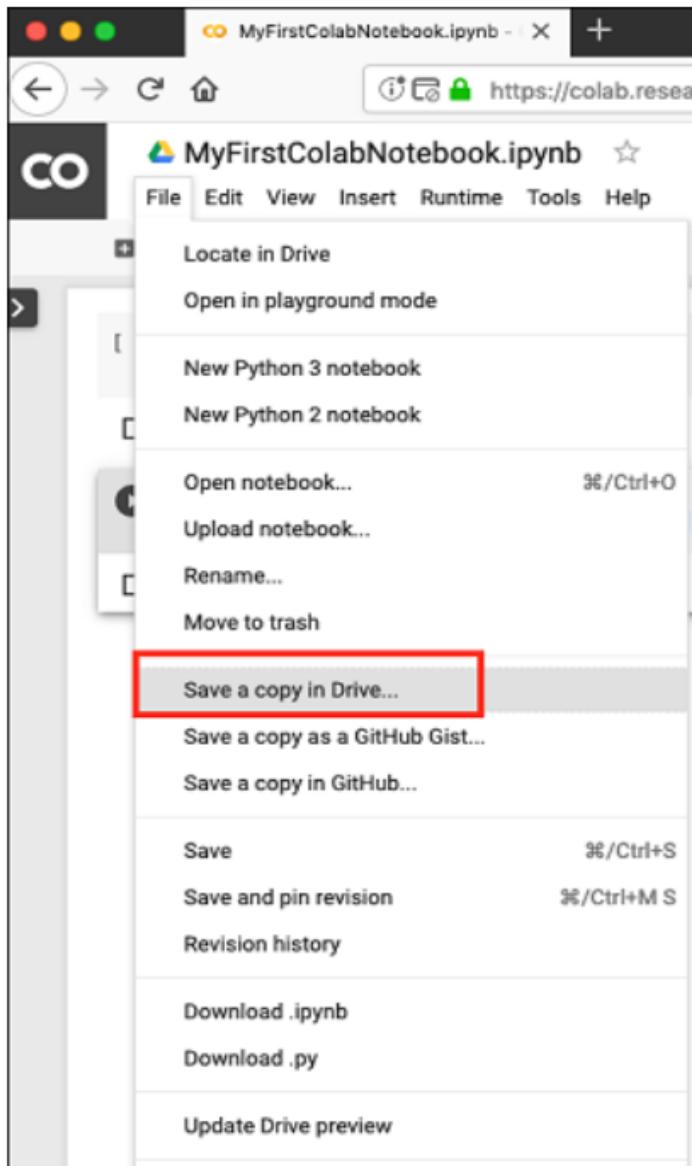
A screenshot of the same Google Colab notebook after executing the "Run All" command. The notebook shows the execution results:

```
[1]: import time
      print(time.ctime())
      Mon Jun 17 11:02:39 2019
```

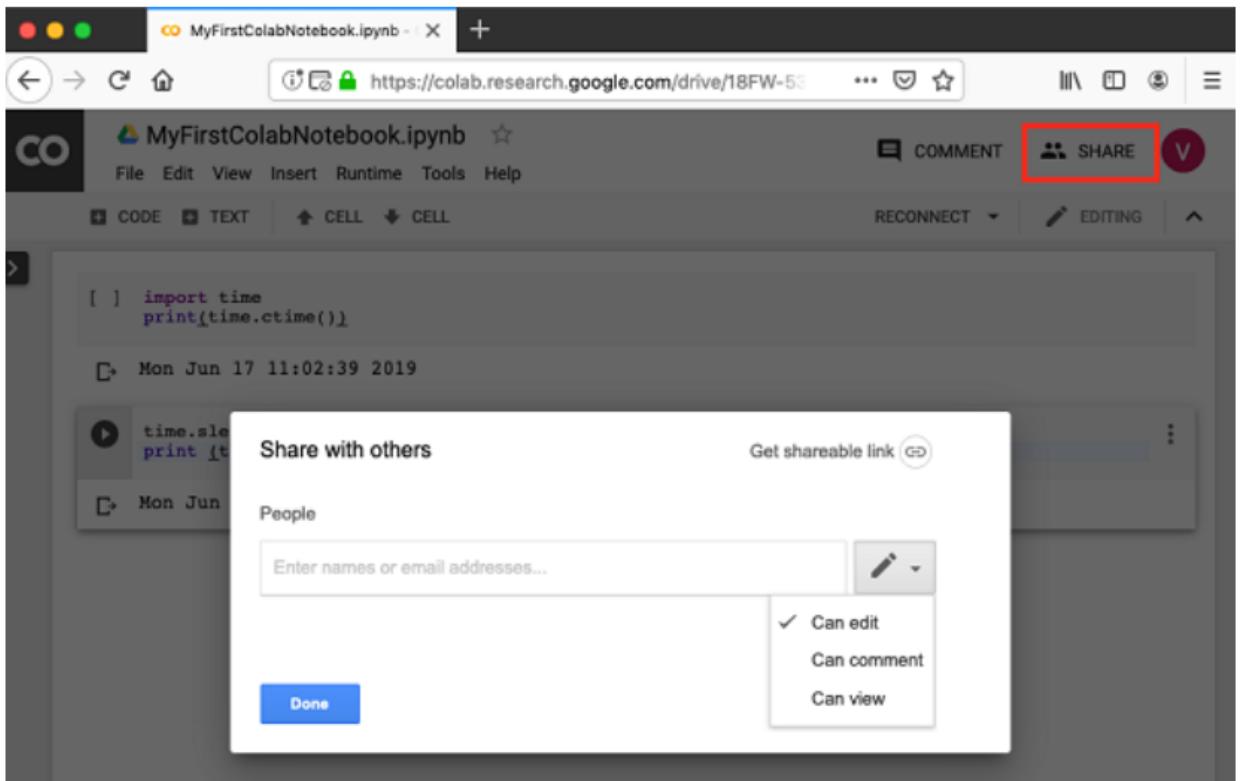
```
[2]: time.sleep(5)
      print(time.ctime())
      Mon Jun 17 11:03:09 2019
```

The "CODE" tab in the cell toolbar is now grayed out, indicating the cell has been run. The "RAM" and "Disk" status indicators are visible in the top right.

## Saving to Google Drive



## Google Colab - Sharing Notebook



## Lab Assignment No. 1

### Objective

To learn how to create and upload a dataset in Python, perform basic statistical analysis, and visualize data using graphs.

### Tasks

#### ◆ Q1: Create a Dataset Manually

- Create a dataset of at least **10 students** with the following columns:
  - Student\_ID,
  - Name,
  - Age,
  - Marks\_Math,
  - Marks\_Science.
- Store the dataset in a **CSV file** named students.csv.

#### Q2: Upload Dataset in Python

- Use **Pandas** to load the dataset.

#### Q3: Observe Dataset Information

Run the following commands and explain the output:

1. `data.info()` → Dataset structure
2. `data.describe()` → Summary statistics (mean, std, min, max, etc.)
3. `data['Marks_Math'].mean()` → Mean of Math marks
4. `data['Marks_Science'].max()` → Maximum Science marks

#### Q4: Perform Some Data Analysis

- Find how many students have `Marks_Math > 50`.
- Find the student with the **highest Science marks**.
- Calculate the **correlation** between `Marks_Math` and `Marks_Science`.

---

#### ◆ Q5: Data Visualization

Use **Matplotlib/Seaborn** to create graphs:

1. A bar chart of Student\_ID vs Marks\_Math.
  2. A histogram of Age.
  3. A scatter plot of Marks\_Math vs Marks\_Science.
- 

#### ◆ Q6: Save Your Work

- Save your notebook as Lab1\_Assignment.ipynb.
- Submit the notebook file along with screenshots of graphs and outputs.

### LAB Assessment

<b>Student Name</b>		<b>LAB Rubrics</b>	CLO3 , P5, PLO5
		<b>Total Marks</b>	10
<b>Registration No</b>		<b>Obtained Marks</b>	
		<b>Teacher Name</b>	Dr. Syed M Hamedoon
<b>Date</b>		<b>Signature</b>	