**ECOX5003 Applied Econometrics**

**2nd May 2025**

**Professor Iain Fraser**

**Using Python via Google Colab (Version 1)**

1. **Introduction**

This document provides a basic introduction to Google Colab which a relatively easy to use environment in which to employ Python to do basic data analysis and econometrics

***What is Google Colab?***

*“Google Colab is a free, cloud-based service that allows users to write and execute Python code in a browser-based environment, essentially a hosted Jupyter Notebook. It's designed to be accessible and user-friendly, especially for machine learning, data science, and education. You can share your Colab notebooks with others and access them from any device with a web browser.”* (Google AI)

You find Colab here: <https://colab.research.google.com/>

It states in the quote that Colab is a hosted Jupyter Notebook.

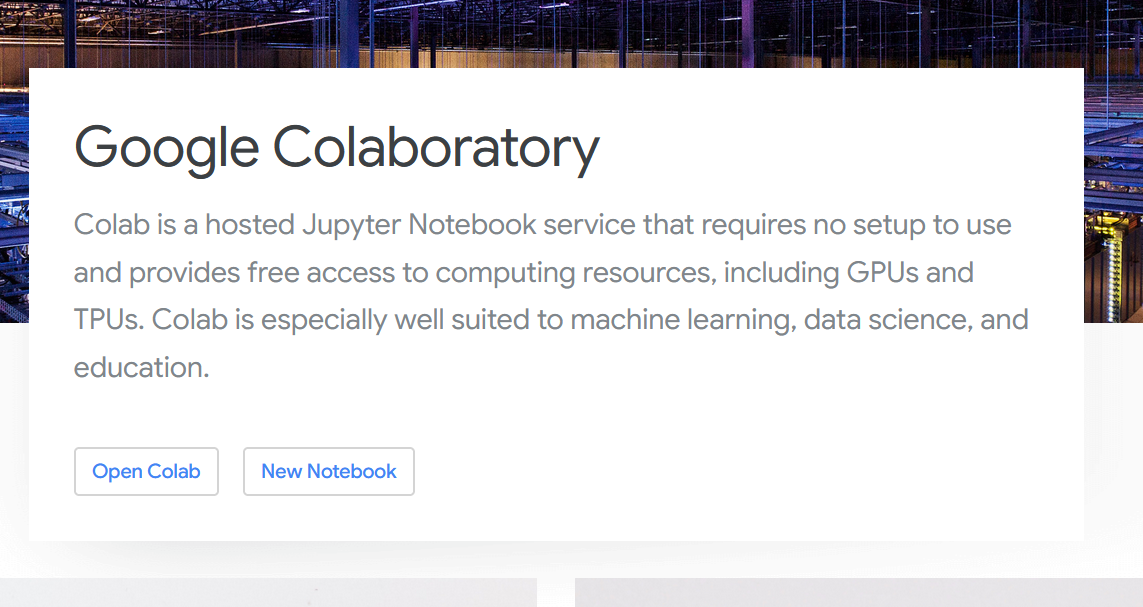
A Jupyter Notebook is essentially the online environment in which you enter Python code and run it. There are other notebooks that are used by Python users but for now we will keep things simple and use what is available in Colab.

The advantage of Colab: It requires relatively little effort to set up and to start running Python code.

The disadvantage of Colab: No Persistent Environment. What does this mean? At then of each session, unless you save it the environment resets. Also, at the start of each session, when you open a “new notebook”, you have to reinstall all the custom libraries you need to run specific programmes. Of course, if you save your notebook file you call it up again and re-run all the code that you have previously written. Once, we get going you will understand!

1. **Open Google Colab**

Call up Google Colab. You will see this (plus lots of other stuff…)



To begin a session, you simply click on “New Notebook".

A screenshot of a computer

AI-generated content may be incorrect.

This the code entry line

This will be the icon you

Click on to import data.

This is a new Juypter Notebook. All the action happens here.

Before we get going, I briefly need to introduce pandas.

***What is pandas?***

"Pandas is a widely used Python library primarily for data manipulation and analysis. It provides powerful data structures, like DataFrames, and functions to handle tabular data efficiently. These data structures allow for easy cleaning, transforming, and exploring data, making it a go-to tool for data scientists and analysts" (AI Google search)

Pandas provides many of the basic handling functions you need. In Colab there is a helpful area called: "10 minutes to pandas"

<https://colab.research.google.com/drive/1_n1RZx2maSeC9w2iy2CsMb8hq2arG6kW#scrollTo=GTIkhd1e7JYH>

This content is a shorter version of the pandas outline which is given here: <https://pandas.pydata.org/docs/user_guide/visualization.html#visualization>

We will always need pandas to be “called” for all sessions.

***What is NumPy?***

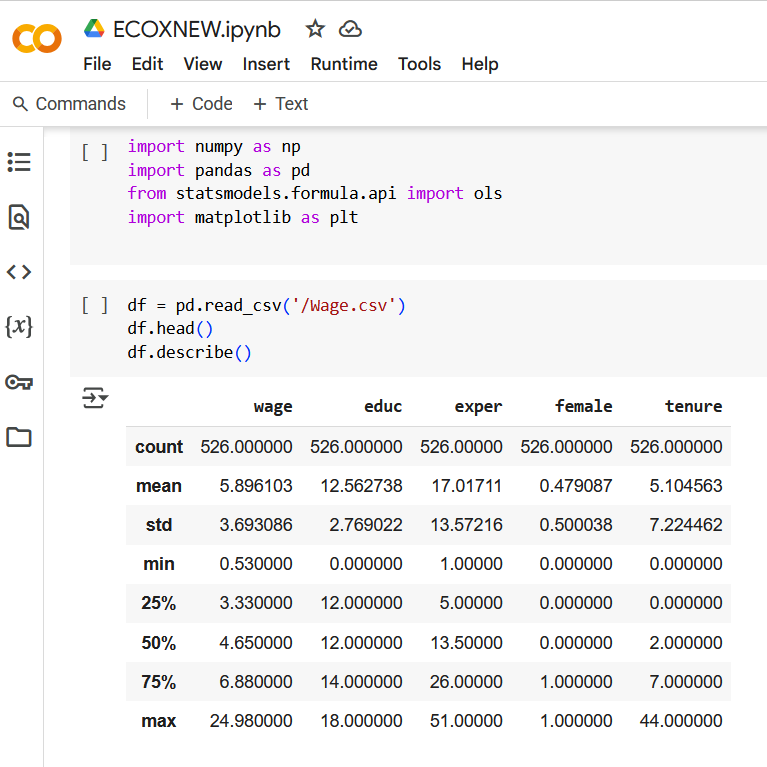
Another important package we call at the start of each session in NumPy. This is a key package enabling scientific computing in Python.

<https://numpy.org/doc/stable/user/whatisnumpy.html>

1. **Getting started in your notebook**

In your “Notebook” you will generally start each file by calling up the required packages. At a minimum it might look like this below. To the right of the first command prompt [ ] there are 4 lines of codes. These lines open the various packages we require to undertake basic OLS.

Below this there is a second command prompt [ ] and this bit of code open a data file the top part of which is displayed.



Here is the code so that you copy and paste:

import numpy as np

import pandas as pd

from statsmodels.formula.api import ols

import matplotlib as plt

1. **Importing Data**

There are various ways to import data into the Notebook. For example, much like with Gretl you can import data from Wooldridge. All you need to know is the file name as shown in the example here (note you need the “! pip install Wooldridge”)

! pip install wooldridge

import wooldridge as woo

import pandas as pd

from statsmodels.formula.api import ols

import matplotlib as plt

import numpy as np

Alternatively, you can import a file directly from one of your own directories (e.g., csv or xlsx).

To do this there are several steps required:

Step 1: Go to the bottom icon on the left-hand side of the screen

A black and white folder

AI-generated content may be incorrect.

Click on it and it will turn "orange"

Step 2: Next click on

`

This will reveal the following:

A screenshot of a computer

AI-generated content may be incorrect.

Step 3: Now click on the icon that has an arrow pointing up. This should take you to your own pc file structure. I see this:

A screenshot of a computer

AI-generated content may be incorrect.

Step 4: Locate the file you want to download. Tip – if downloading Excel data save as a csv file format. Click on the file you want. This file is now within Colab.

Step 5: Now scroll to the bottom of the black file structure and you will see this file.

Right click and several file options will appear including "Copy Path" - click on this.

Once the file is within the Colab file area can then call up as a data frame (df)

You copy the Path into the command below - if a csv file then use csv, if xlsx use xlsx

In the bracket between ('') copy the path

df = pd.read\_csv('/Wage.csv') ---- put your own file path here

df.head()

df.describe()

Now run the command lines and your data set should load and you will the names of the data by column and all the basic descriptive statistics. Finally, another way to load data is as follows:

X=pd.read\_csv('https://raw.githubusercontent.com/KelvinBalcombe/Econometrics/main/covidseverity.csv')

This data has been placed on an open-source data area called GitHub.

1. **Saving your session**

Given how Colab works it makes sense to save your session Notebook file. I have set up the option to save my Colab files directly to my Google drive. All, I then need to do, is go to my Google drive and I can reopen any files I have previously saved.

Alternatively, I can also store on an open-source web based repository such as GitHub.

I will upload the Python code as the module progresses onto GitHub at this address:

<https://github.com/imfraser/ECOX5003-Public.git>

1. **Some useful Python commands**

Over the next few weeks, I will gradually introduce new Python commands as the tasks we undertake become more involved. However, some useful commands to help you get started are provided here:

How to calculate the correlation between two variables:

df[['educ','exper']].corr()

Run and print a simple OLS regression

mod=ols('educ~exper',data=df).fit()

print(mod.summary())

For a multiple regression you add variables as shown here

mod=ols('educ~exper+female',data=df).fit()

print(mod.summary())

To generate a histrogram you can use code such as this:

plt.rcParams['figure.figsize'] = [10, 5]

df['educ'].plot(kind='hist',y='educ',bins=20,title='Educ',grid=True)

plt.show

Scatter diagram

df.plot(kind='scatter',y='educ',x='exper',grid=True)

Run a basic regression and show as a best fit line

import seaborn as sns

sns.regplot(data=df,y='educ',x='exper')

OLS with robust standard errors

#With Robust Standard errors (Note the # allows me to write explanatory text in the code)

mod=ols('educ~exper',df).fit(cov\_type = "HC0")

print(mod.summary())

To create a new variable:

df['expeduc']=df['educ']\*df['exper']

To examine the new variable:

df.head()

df.describe()

Other Model Types

To call up other model types can use the command

from statsmodels.formula.api import ols,probit,logit