

Introduction to Data Science in Accounting – Practice Set 1

We will try to do the following three activities:

Step 1: Setting up your environment

Step 2: Printing “Hello World”, your first Python instruction

Step 3: Carrying out some basic coding exercises in Python

By the end of the set, you should have a working Python environment, and you should be able to carry out some very basic coding exercises.

Difficulty level – Easy

Required: A Google account (if using online version)

Step 1: Setting up your environment

Our Python environment is called Colab – it will be entirely online, and is supplied by Google. The correct URL to go to is:

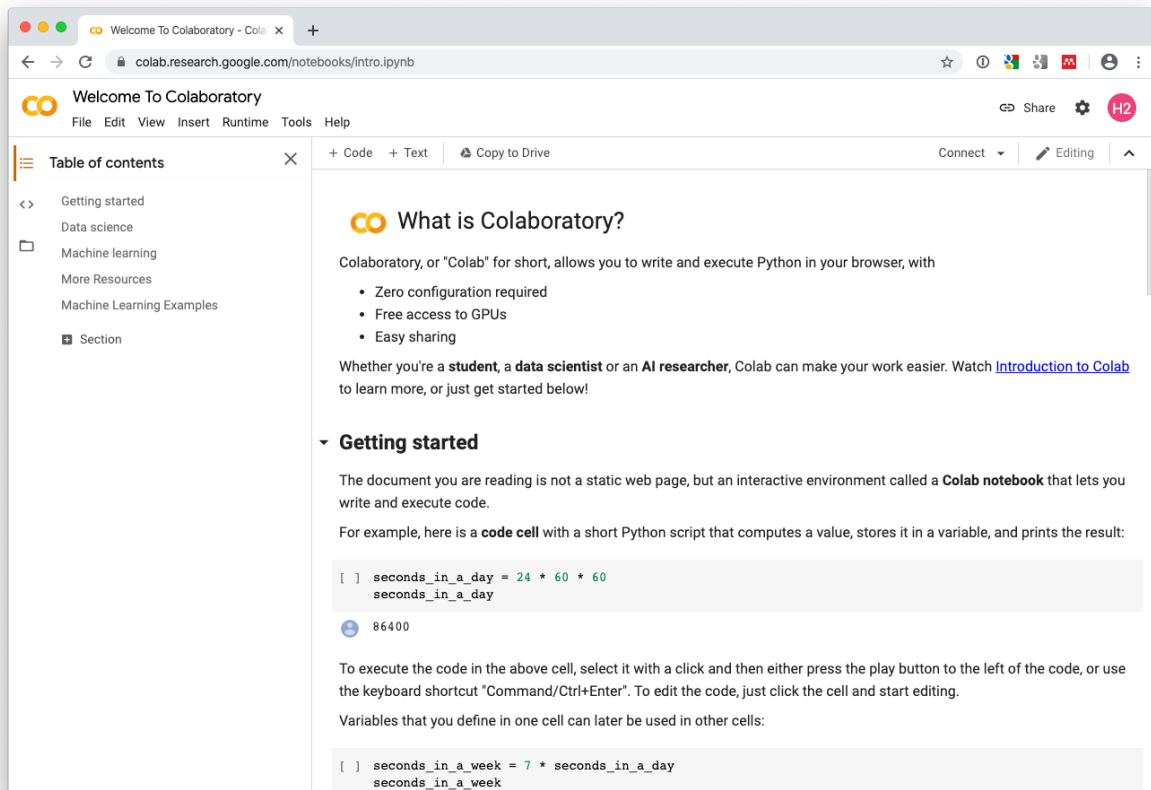
<https://colab.research.google.com>

The advantages of using an online environment such as Colab are that

- 1) you will be running software on Google’s hardware, which is probably a bit faster than your own
- 2) your progress is saved in the cloud, so you can access from any device, and do not have to worry about losing your data
- 3) you will always use the latest version of the libraries (packages of code created by others). Keeping libraries up to date locally can be problematic, and using libraries that are not up to date may lead to unexpected errors.

For those who are cannot access Colab or prefer an offline Python environment, I recommend downloading JupyterLab at <https://www.jupyter.org/> or <https://www.anaconda.com/>. There are some small differences between Colab, JupyterLab or Anaconda, but none are substantial for this exercise.

Colab will generate an intro workbook for you, which will look a bit like this:

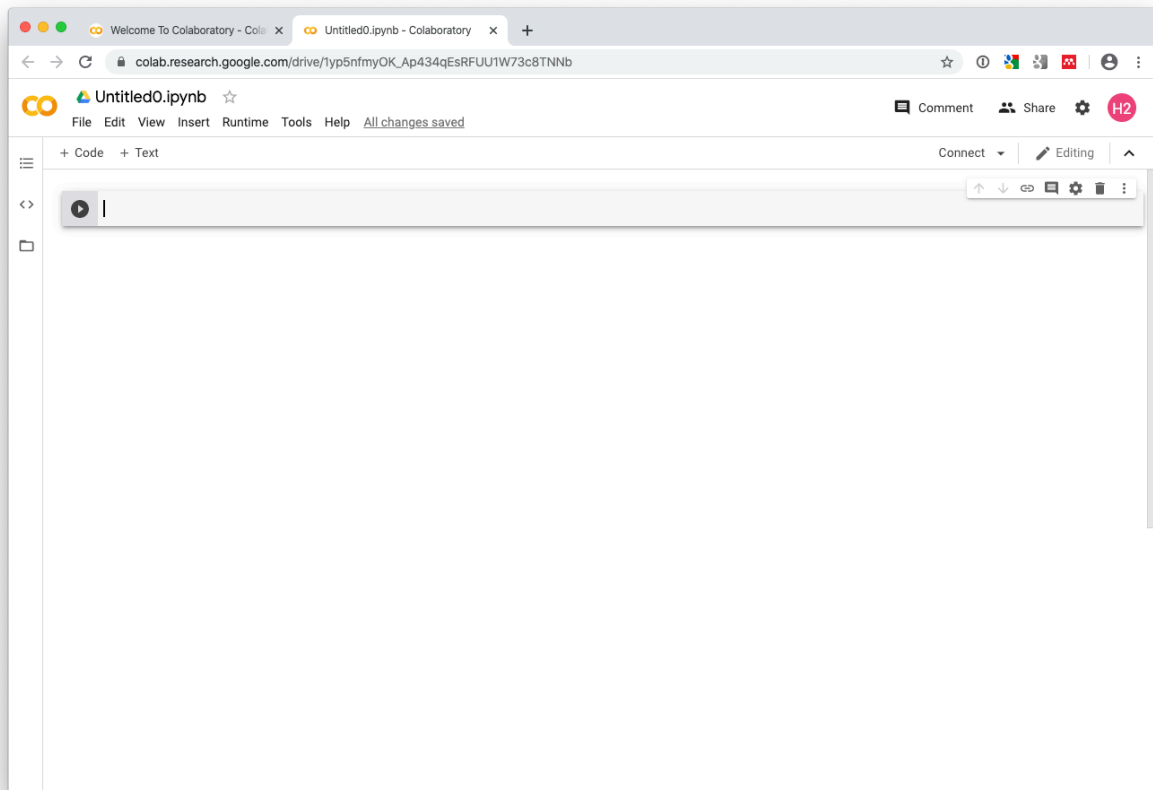


If you do not see the side bar with 'Table of contents', you can click on the icon with the three lines to make it appear (and disappear).

It's worth reading the text under 'What is Colaboratory'.

Colab works with Jupyter **notebooks**, which is the common way to use Python in data science. Both Jupyter's JupyterLab and Anaconda's JupyterLab work with the same type of notebooks.

To work with data, we will first create a new notebook. Click on 'File' and then 'New Notebook'.



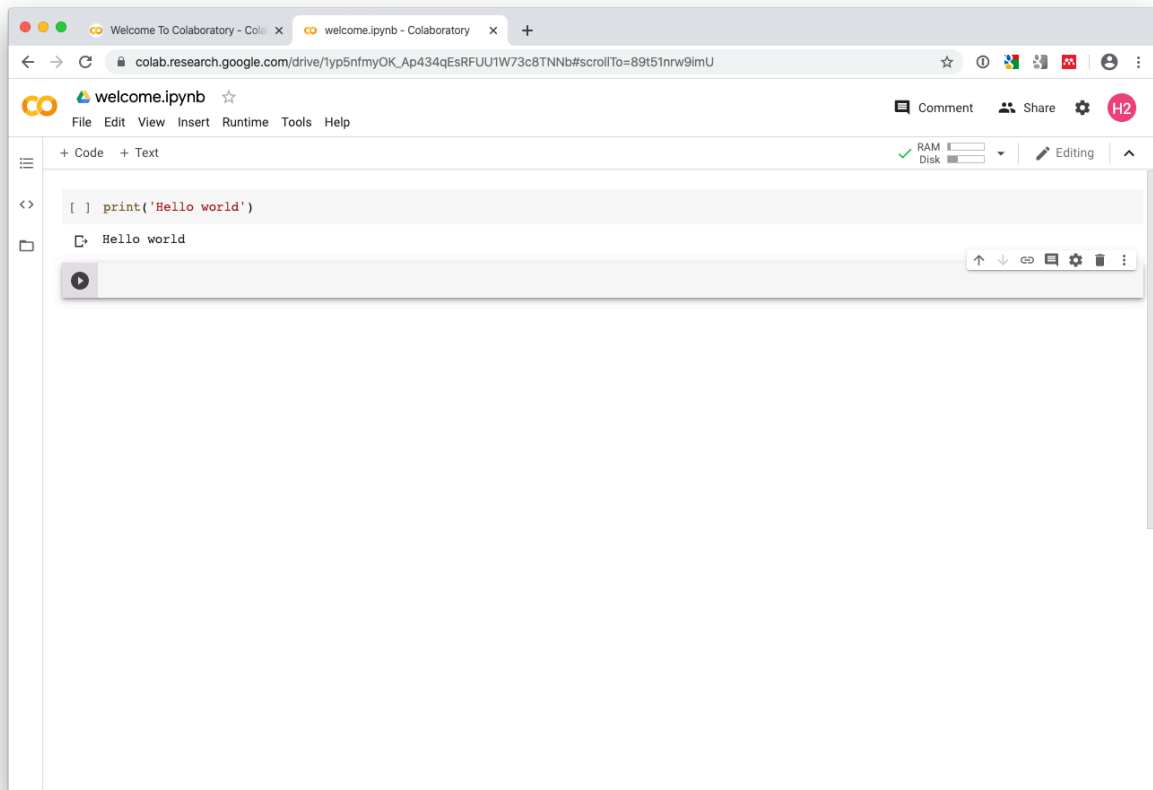
Click on the word 'Untitled' to give your notebook a name. Rename your notebook to 'welcome' or any other name you would like to give it.

The notebook will auto-save from time to time. To force a save, click 'File', then 'Save'.

Step 2: "Hello World"

Printing a 'hello world' message is often the first thing people learn to do when learning a computer language. In Python, this is very simple. Type: `print('hello world')` and then press Shift-Enter to run.

This might take a while, initially, as Google will allocate a Python runtime to you and connect to it. Everything will be faster when Python is up and running. You will see a green tick, and the RAM and Disk space that you can use.



The words Hello World are encapsulated by single quotes to indicate that is a String value. It appears in red in Colab. Try to print your own name now: `print('hello <your name here>')`

Step 3: Basic coding exercises in Python

Try out the following commands, each in a different cell (use Enter to add a new line in the cell, and use Shift-Enter to run the code in a cell), and verify the output.

Command	Verify output	Notes
<code>2 + 2</code>	4	Don't forget to press Shift-Enter to process the cell, not Enter
<code>a = 1</code> <code>a</code>	1	Assign a constant value to a variable (a). Press 'Enter' for the new line.
<code>b = a + 1</code> <code>b</code>	2	Work with variables
<code>c = a + b</code>	Nothing	This is an assignment, you are not asking the cell to output something

<code>print(c)</code>	3	This works the same as just writing 'c' as the last line of the cell.
<code>p = 4 * 5</code>		Multiplication. Print "p" to see the outcome.
<code>d, e = 1, 2</code>		Multiple assignments on one line. To see if this works, now print "d" and "e".
<code>2 == 3</code>	False	Work with Booleans, which are True or False. The == symbol means 'is equal to'
<code>2 == 2</code>	True	
<code>is_same = c == (a + b)</code>	Nothing	"is_same" is an arbitrary variable name, just like "a" and "b"
<code>is_same</code>	True	Output the variable

Try re-running a previously created command. You can put the cursor in a cell, and then press Shift-Enter again.

Select Edit -> Clear all outputs, to clear all the outputs from the cells.

Select Runtime -> Restart and Run-All to restart the Python runtime (you will lose all data in the variables), and run the notebook one cell after another.

Python has several built-in **functions**, that will be useful as we look at data. For example, there is the `min` function, the `sum` function, and the `round` function. These functions take variables as *input*, and then produce an *output*.

Command	Output	Notes
<code>min(1, 2)</code>	1	Min function
<code>a = min(1, 2)</code> <code>a</code>	1	
<code>list = (1, 2, 3, 4, 5)</code>	Nothing	Assignment only
<code>min(list)</code>	1	
<code>sum(list)</code>	15	
<code>round(3.233, 2)</code>	3.23	The second argument of this function is the number of decimals. Experiment with other values.

We can also make our own functions, by using the `def` command. The instruction below builds a value function, which takes a quantity and a price, and then returns the total value. Note the indent on the second line, this is necessary otherwise Python doesn't know that the second line is part of the function that we started in the first line.

Command	Output	Notes
<code>def value(price, quantity): return price * quantity</code>	Nothing	Note the indent on the second line (press tab)
<code>value(10, 100)</code>	1000	

Try out different prices and different quantities to verify that the function works.

This completes your initial exercises in Python.

You are welcome to experiment a bit further, with different values and functions etc.