Laboratory # 1

Starting with Jupyter Notebook

1 Prerequisite

In this course, we will work with Anaconda 2023.09 (64-bit) and Python 3.12. Anaconda installation includes Jupyter Notebook local environment. You can install Anaconda on your laptop and use Jupyter Notebook locally.

Alternatively, Jupyter Notebook is available online via https://ucalgary.syzygy.ca/. You can also access Google Colab by going to https://colab.research.google.com/ and use it online as well.

The Anaconda distribution is available on https://www.anaconda.com/download/ for either Windows, Mac or Linux.

To start the Jupyter Notebook, go to the main menu and find the entry `Anaconda3(64-bit)/Jupyter Notebook`. It will open a console window and activate your browser with the Jupyter dashboard as shown in Fig. 1.

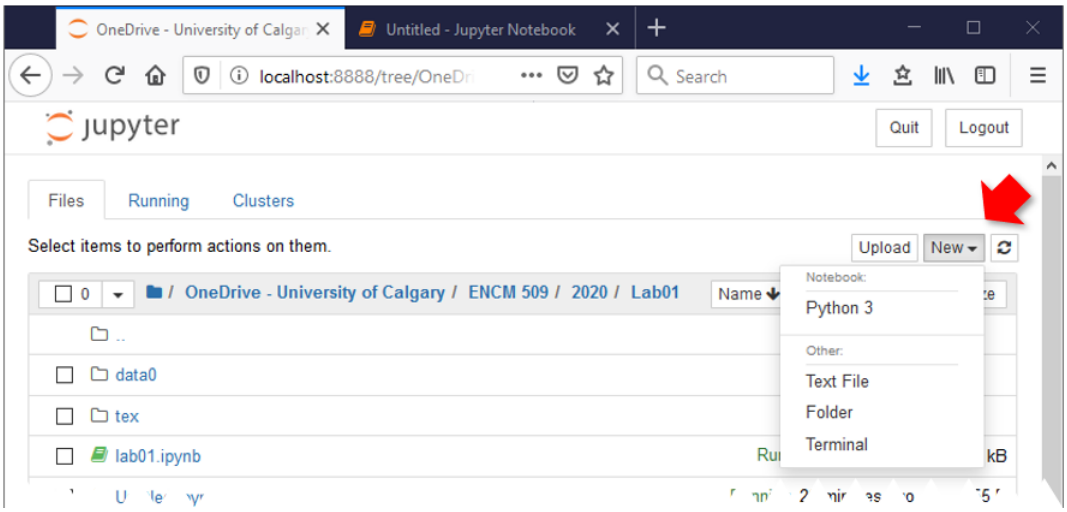


Figure 1: Creating a notebook on Jupyter Notebook.

To start, read the document `Tutorial_Jupyter_Markdown.pdf` which is the tutorial on how to use Anaconda with Jupyter Notebook. Run `Lab01-Markdown.ipynb` to practice, before proceeding with this Lab. Both files can be found on D2L.

We provide a template notebook `Lab01-template.ipynb` that should be used for your report. Open it in the Jupyter Notebook. Rename the template notebook to have your name on it, for example: `lab01_YourName`.

2 Introduction

Jupyter is a web-based interactive environment that work with Python and some other programming languages. Jupyter Notebook will allow us to create the reproducible **analysis results and your**

lab report (text) in a **single document**.

In this lab, we will show how to generate some synthetic data and plot it using 2D and 3D graphs. For these tasks, two Python libraries will be used, Matplotlib and NumPy.

To make the code readable, we recommend you use comments like this when necessary:

```
# this is a comment sign in Python
```

2.1 Matplotlib

Matplotlib (https://matplotlib.org/) is the library used to plot data and show images. Its command syntax is very similar to MATLAB.

Before plotting, follow these steps:

1. Import the Matplotlib `pyplot` module:

   ```
   import matplotlib.pyplot as plt
   ```

2. To plot inside the Notebook, it is necessary to put the command `%matplotlib inline` after importing the Matplotlib:

   ```
   import matplotlib.pyplot as plt
   %matplotlib inline
   ```

2.2 NumPy

NumPy (https://numpy.org/) stands for *Numerical Python*. This library is used for scientific calculations. Similarly to Matplotlib, NumPy was developed to facilitate the transition from MATLAB to Python.

Since NumPy is a library, it needs to be imported using the command:

```
import numpy as np
```

The next section will explain how to use this library.

3 The laboratory procedure

The provided template notebook `Lab01_template.ipynb` shall be used for your report. Open it with Jupyter Notebook and rename adding your name at the end, for example: `lab01_YourName`.

3.1 Generating synthetic data

As an example, we will generate a vector using the NumPy's function `arange`, which syntax is as follows

`arange(start, stop, step)`

where:

`start`: is an optional parameter which specifies the starting value of your vector. If not specified, `start` is 0;

`stop`: is the last value of the vector;

`step`: is an optional parameter which specifies the step between to values of the vector. If not specified, the default `step` is 1.

The code sample below shows how `arange` should be used to create an array of numbers from 0 to 2 with a step of 0.1. Pay attention to the mandatory `import` before the function call:

```
import numpy as np
x = np.arange(0, 2, 0.1)
```

The return value of the `arange` function is the data in the format `ndarray` which is a *n-dimensional array*, a basic data type in NumPy. To see the created vector, use the `print` command with the argument `x`:

```
print(x)
# the result will be printed as follows:
[0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9]
```

Another popular function to generate data is `linspace`. Its syntax is similar to the `arange`: `linspace(start, stop, num=50, endpoint=True, ...)`. The difference is the optional parameter `num` (by default, `num=50`). It is used to specify how many values inside the interval (`start`, `stop`) you want to generate. The steps between the values will be inferred based on the amount of points.

As an example, let us create a vector consisting of 10 points between 0 and 1:

```
# we assume that NumPy was already imported
x = np.linspace(0, 1, 10)
```

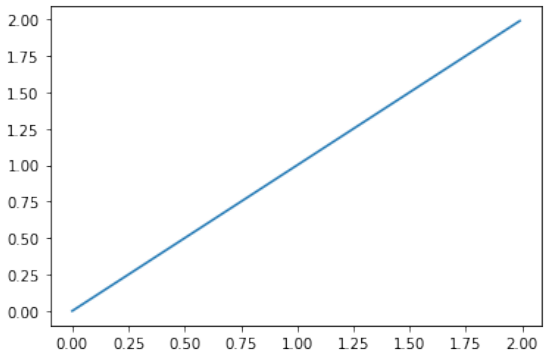
Again, using the `print` command, we can see the result:

```
print(x)
[0.         0.11111111 0.22222222 0.33333333 0.44444444 0.55555556
 0.66666667 0.77777778 0.88888889 1.        ]
```

3.2 Plotting data with Matplotlib

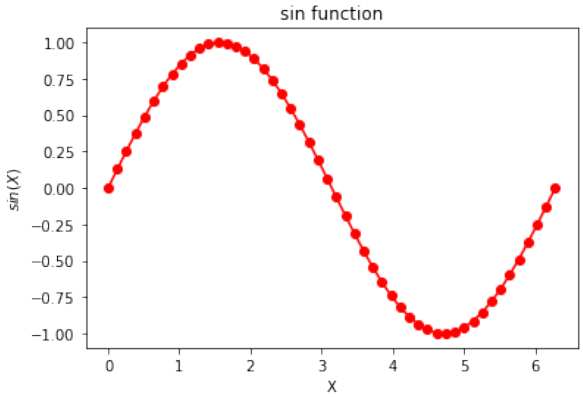
To visualize data, we will use Matplotlib. For more advanced plotting functions, you may refer to Matplotlib's Tutorial[1] and Examples[2]. In our example, we will use the previously generated `x` variable, and plot it as a line composed of its points, as shown below:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 2, 0.01)
plt.plot(x, x)
```



Another example is plotting a *sine* function using constants[3] and built-in functions from NumPy and Matplotlib:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 2*np.pi)
y = np.sin(x)
plt.plot(x,y, '—ro')
plt.title('sin function')
plt.xlabel('X')
plt.ylabel('$sin(X)$')
```



3.3 Lab 1 Report

The report to be uploaded to D2L by the deadline (the following Thursday) must to be submitted as one file, a Jupyter notebook file `Lab01_YourName.ipynb`. You can save it using the Jupyter menu: `File -> Download as`.

It includes the following graded components (10 marks total):

- (1 marks) Exercise 1: Markdown code and text (picture).
- (3 marks) Exercise 2: the code and the generated output, including a brief description (use Markdown), equations, illustrations/graphs and analysis of the results.
- (3 marks) Exercise 3: the code and the generated output, including a brief description, equations, illustrations/graphs and analysis of the results.
- (3 marks) Exercise 4: the code and the generated output, including a brief description, equations, graphs and conclusion.

[1] https://matplotlib.org/tutorials/
[2] https://matplotlib.org/gallery/
[3] https://numpy.org/doc/stable/reference/constants.html

Exercise 1

Create a title of Lab 1 report and insert the Dino picture (see the image below, which is also on D2L as a dino.png file) in your Jupyter notebook report using Markdown language.

Exercise 2

Consider a variable and its interval: $x \in (-6, 6)$. Let us create a function $f(x)$:

$$f(x) = \sin x + \frac{cos(4x)}{4} + \frac{sin(5x)}{5} + \frac{cos(6x)}{6}$$

- Plot $f(x)$ including the plot title, labels and grid.
- Change the markers (to `"o"`, `"^"` and `"1"`), line styles (to `"--"`, `":"` and `"-."`) and color (red, gray and magenta).
- **Select only three combinations of markers, line styles, and colors based on your preferences; the result shall be the three plots, in addition to the original, default one.**
- Description of changes made to the previous examples using Markdown.

Exercise 3

Consider the data below. It is a demographic data (such as age and gender) of the passenger and crew population of the Diamond Princess Cruise ship circa February 2020. Note the demographic data is often used in biometric-enabled decision support systems.

```
# percentage of male and female
gender = [55, 45]
# counting of people at each age interval shown in age_labels
age = [16, 23, 347, 428, 334, 398, 923, 1015, 216, 11]
age_labels = ['0—9','10—19','20—29','30—39',
              '40—49','50—59','60—69','70—79','80—89','90—99']
```

You are tasked with reproducing the plots of distribution of such data shown in Fig. 2. Use the functions `bar(...)` and `pie(...)`. Consult the Matplotlib documentation (https://matplotlib.org/contents.html) to understand the parameters of the plots.

Exercise 4

Consider the ship population's age data from Exercise 3.

The intent of this exercise is to analyze the statistics from the provided *age* data. We wish to calculate the mean and standard deviation of the age of the ship population using the NumPy functions `np.mean(..)` and `np.std(..)`, respectively. Those two values are sufficient to generate and plot a normal distribution.

Let us initially expand the `age` vector by indexing it.

As example, consider the age intervals (see `age_labels`) as indexes:

- the age interval [0-9] will correspond to index 0,
- the interval [10-19] will correspond to index 1, ...
- and so on, as interval [90-99] will have index 9.

Next, create a new vector with 16 numbers 0, 23 numbers 1 and so on. These values correspond to the first two numbers in the `age` vector. The new vector is the one you will use to calculate the *mean* and *standard deviation* values. Next, generate and plot the corresponding normal distribution.

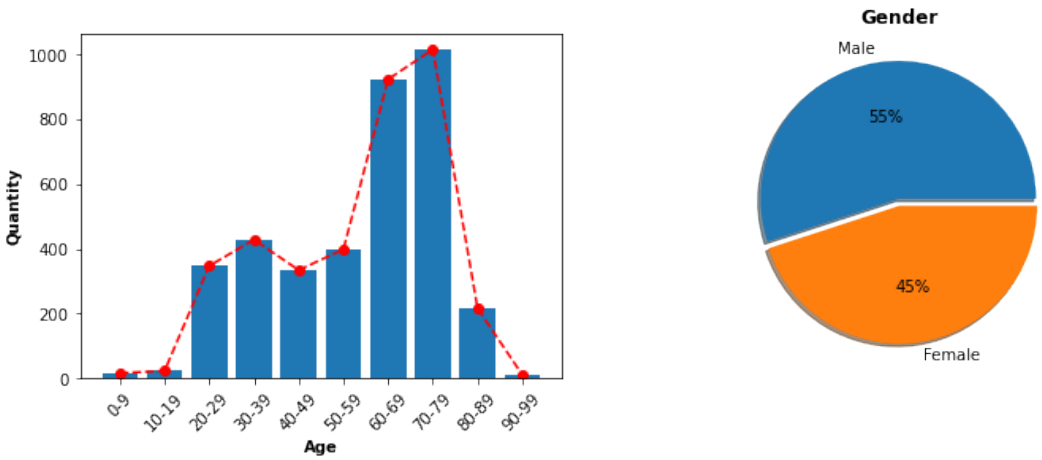


Figure 2: Visualization of the age and gender data distribution for the cruise ship.

Acknowledgment

The template for this Lab and initial plotting exercise were developed by Dr. H. C. R. Oliveira (postdoc in the Biometric Technologies Laboratory in 2020-2022). We also acknowledge this course TAs, O. Shaposhnyk, I. Yankovyi and M. Zakir, for and verifying this lab code.

Dr. S. Yanushkevich
Biometric Technologies Laboratory
January 12, 2024.