MSE-213 Autumn 2024 – Python Practice Week 7

Run in the conda mse213 environment, use a Jupyter notebook. If you have trouble with that, you can use the online environment noto.epfl.ch

If you don't know how to proceed, stackoverflow.org, google.com, chatgpt.com can be very helpful. For a specific Numpy function, numpy.org has the documentation, or just type "np.function?" into Jupyter (where function is the name of your function of course).

Introduction : Python Function

Functions are isolated code blocks that can be executed at any point of your script. Functions are defined by their name, their parameters and their return value. The function can then be called by using it's name and providing the necessary parameters.

Use the following syntaxe to define and call your function :

```
def function_name(param1, param2, param3=9):
    # ---- Your function code
    value = (param1 + param2 + param3) / 3
    # ---
    return value

mean = function_name(2, 4, 5)
```

Parameters can be of any type if not implied otherwise. They can also be given a default value by writing $(\text{param} = \text{default})$. If not given a value when calling the function, this parameter will take its default value.

Use functions either to do repetitive actions, or make your code more readable by separating and naming relevant code blocks.

Task 1: Simple functions

Aim: Learn how to create and call functions. Use default parameters.

- Create a function that takes as parameters mass and velocity, checks if the mass is positive, and returns the kinetic energy. Recall : $E_{kin} = \frac{1}{2}mv^2$.
  Call this function for a different values.

- Consider a satellite of mass m orbiting the Earth at a distance r.
  Create a function that takes the distance from earth and returns their orbit speed (considering circular orbit). Set G and M as parmeters of your function with default values. What is the orbit speed of the ISS ? How would that speed change if the gravitationnal constant was doubled ? And if the ISS was 1 km higher ?

  Recall : $v = \sqrt{\frac{GM}{r}}$, where M is the mass of the earth and G the gravitational constant.

  $G = 6.67430 \times 10^{-11} \frac{Nm^2}{kg^2}$, $M_{earth} = 5.97 \times 10^{24}\ kg$, $r_{ISS} = 6.78 \times 10^{6}\ m$

Task 2 : Functions on numpy arrays

Aim: Learn how to apply functions to numpy arrays. Learn about numpy functions.

- Using special observations techniques, you were able to observe the velocity of a sample of space debris in the Earth's close orbit. Generate a sample of N=100 velocities following a gaussian distribution $\mathcal{N}(7.5\ \mathrm{km/s},\ (0.1\ \mathrm{km/s})^2)$.
- Create a function to relate a single velocity to the altitude (take into account the earth's radius). Call this function on the first element of the velocity list.
- Call this function, but pass the entire array of velocities instead of a single one. What do you see ? What is the difference between giving the function a velocity or an array of velocities ?
- You are now leading a small EPFL startup that plans on cleaning the Earth's orbit. With your current ressources, you can only go to 500km altitude.
    - Create a functions that takes an altitude from the previous step and returns either the unchanged value if you can reach this debris, or a Not A Number if not ($\mathrm{np.nan}$, this is the default way of eliminating a value that should not be taken into account).
    - Call this function with your array of altitudes. What do you see ? Is this expected ?

The np.vectorize(my_function) function return your function transformed into an element wise function, that is to say calling your function with an array now applies the function on every point of your array.

- Vectorize your last function and try again. What do you see ?
- To take the mean and std of an array while ignoring the NaNs, use np.nanmean() and np.nanstd().
  What is the mean and standard deviation of the reachable debris ?
- Repeat this experience with M = 1000 sample of gaussian velocities. What is the mean of the means and stds ?