Python exercise: Random numbers

Part I: Generation of random numbers

- **RandomNumberGeneration.py** creates 10^7 random numbers and stores them in a list. Run it, and measure the time it takes to run the code with **time.time().**

Explanation: A list is defined with **ListOfRandomNumbers= []**. Numbers are appended to the list with **ListOfRandomNumbers.append(random.random())** inside a loop (**for i in range(10000000)**)

A note on time.time() and UNIX time: On most operating systems, the number of seconds since January 1st 1970 is returned by time.time() – this is called UNIX time.

Part II: Generation of random numbers with numpy

- Run **RandomNumberGeneration_numpy.py**, which is a NumPy implementation of the same program. We now use the functions **numpy.random.random**, **numpy.max**, **numpy.min** and **numpy.mean**.
- Modify the program, so it uses **time.time()** to measure the time it takes to draw the random numbers and calculate the min, max and mean.

Part III: Calculation of π using random numbers

- **Pi.py** calculates π with python using lists and the random package. Run the program and make sure you understand what is going on.
- Exercise: Write the same program using NumPy. The following functions and code fragments will be useful: **x=numpy.random.random(10000000)**, **r = numpy.sqrt((x-0.5)**2+(y-0.5)**2)** and **NInsideCircle = x[numpy.where(r<0.5)].size**
- Compare the speed of the two programs. The NumPy program should be around 10 times faster than the python-list program.
- Explain in text why the program using numpy arrays is faster in comparison to the program using python lists. *Bonus exercise: Try to get ChatGPT to explain this, and check whether you agree with it and understand the response.*