Activities

(1) Use np.arrange in order to generate an array of numbers between 0 (incl.) and 10 (incl.). Then use np.full to generate an array of numbers 2 of the same size. Then use exponentiation to generate an array of powers of two.

(2) Use the same method to generate a numpy array of powers i^i for i between 1 and 20. You will see that there are overflow errors, unlike in traditional Python.

(3) Use numpy's function np.sqrt on a 4 by 4 matrix to create a matrix with values

$$\begin{pmatrix} \sqrt{1} & \sqrt{2} & \sqrt{3} & \sqrt{4} \\ \sqrt{5} & \sqrt{6} & \sqrt{7} & \sqrt{8} \\ \sqrt{9} & \sqrt{10} & \sqrt{11} & \sqrt{12} \\ \sqrt{13} & \sqrt{14} & \sqrt{15} & \sqrt{16} \end{pmatrix}$$

(4) Create an array of remainders modulo 7 of the squares $[1,4,9,16,\ldots,10000]$.

(5) Use a u-function to create the values:

$$[\sin(0), \sin(\frac{\pi}{16}), \sin(\frac{\pi}{8}), \sin(\frac{3\pi}{16}), \ldots, \sin(\frac{15\pi}{16}), \sin(1)]$$

We use np.pi for the value of π. Also, there are 17 values in the array. You will notice that the values are not completely precise.

(6) Create a normally distributed 3 by 4 matrix (with mean 10 and standard deviation 1). Then use .mean(0) to calculate the mean across the first dimension, i.e. the mean of all the columns.