LAB 5

Python: Lists and arrays

Learning goals:
Creating and manipulating lists.
Creating and manipulating vector and matrix arrays.
Using `range()`, `arange()`, and `linspace()`.

5.1 Basic list and array operations

1. Create a vector that has the elements: 6, 8*3, 2.5, ‘labtime’, and $\sqrt{2}$.
In general for this lab, don’t just create vectors, but also print out their contents for verification.

2. Create a vector in which the first element is 0, and the last element is 42, with an increment of 3 between the elements (0, 3, 6, ... , 42).
Starting with this question, create elements using “shortcut” techniques such as the colon operator, rather than manual entry of all the values.

3. Create a vector in which the first element is 18, the elements decrease with increments of -4.5, and the last element is -18.

4. Create a list with 16 equally spaced elements in which the first element is 5 and the last element is 61.

5. a) Create two character strings, ‘*name1*’ and ‘*name2*’, containing your first and last names.
b) Create a third string variable by concatenating the first two strings, along with an appropriate space.

6. a) Create a vector, named ‘`Afirst`’, that has 13 elements in which the first is 3, the increment is 4 and the last element is 51.
b) Create a new vector, named ‘`Asecond`’, that has seven elements (*do this step first.*) The first four elements of `Asecond` are the first four elements of the vector `Afirst` and the last three elements of `Asecond` are the last three elements of the vector `Afirst`.

5.2 More vectors, 2D arrays, and vectorization

1. Create an array `w` with values 0, 0.1, 0.2, ..., 3.
Print the following: `w[:]`, `w[:−2]`, `w[::5]`, `w[2:−2:6]`.
Explain in each case the elements of the array that are being printed.

2. Create a 5-element vector, and a 2×3 array, with names and values of your choosing. Inspect their properties using `type()`, `whos`, `len()`, `shape()`, and `size()`. Report the results and comment on what they mean.

3. Create a vector, try out the following built-in functions on it, and comment on what they do: `len()`, `min()`, `max()`, `sum()`, `sorted()`, `sort()`

4. Create a vector, named ‘`same`’ with eleven elements that are all 4.

5. Create the matrix shown below by using vector notation for creating vectors with constant spacing and/or the `linspace` command when entering the rows. (Do not worry if the number formatting is different from what is shown here.)

$$B = \begin{pmatrix} 0 & 4 & 8 & 12 & 16 & 20 & 24 & 28 \\ 69 & 68 & 67 & 66 & 65 & 64 & 63 & 62 \\ 1.4 & 1.1 & 0.8 & 0.5 & 0.2 & -0.1 & -0.4 & -0.7 \end{pmatrix}$$

6. Replace the fourth column in matrix B with the numbers 1 to 3.

7. a) Using the colon operator, create a 2×5 matrix, named `msame` in which all the elements are the number 7.
b) Add a third row of zeros, and then a fourth row with the numbers 1 to 5.
c) Add a fifth row that multiplies the second row by the number 6.
d) Concatenate (= combine horizontally) the matrix B above with the middle three rows of `msame`.

8. Evaluate the dot product of two vectors, using vectorization to do so in just **one** line of code, where the dot product is: $\mathbf{a} \cdot \mathbf{b} = \sum_i a_i b_i$.