Exercises 3: Numpy and matplotlib

Introduction to data science and Python programming for medical research (2025)

1 Array 1D

- Create an array, 1D, with three elements.
- Print the array itself and print its `shape`.

2 Array 2D

- Create an array, 2D, that contains 2×3 elements, two rows and three columns. The input argument should be a nested list with two rows.
- Print the array itself and print its `shape`.
- Create an array of random numbers if same shape as your 2D array. Use the `random` module in `numpy`.
- Add the random array to the original array. Print the resulting array.

3 Array creation and slicing

- Use numpys `arange` to create and array with values from 2 up to and including 8. Print the array.
- Use slicing to select the two elements from the second number in the array. Create a variable that is assigned this slice. Print the slice.
- Change the two numbers in the slice array by indexing them separately.
- Print the slice array as well as the original array. What happened to the values in the original array and why?

4 Array 2D creation, flattening and reshaping

- Create a 2D array with two rows and three columns filled with `ones`.
- Create another array, 1D, with a range of numbers that match the total size, *i.e.* number of elements, of the first array.
- Create a variable that stores the shape of the 2D array.
- Flatten the 2D array.
- Add the two flat arrays.
- `reshape` the resulting array such that it gets the shape of the original 2D array. Print the results.

5 Random and Boolean arrays

- Create and array of 100 random integers with values ranging from 0 to and including 9.
- Use a comparison to create a Boolean array that indicates which elements in the array are equal to 9.
- Use the Boolean array to determine how many elements in the original array are equal to 9. Print this number.

6 Plotting functions $y = f(x)$

- Create a numpy array, `x`, with linearly spaced values ranging from 0 to 4π. Use `np.linspace` and `np.pi`.
- Use `x` to compute values for the function

$$y_1(x) = 2\sin(x)$$

- Use matplotlib to create a figure, plot `x` and `y1` and `show` the figure.
- Use `x` to compute values for a second function

$$y_2 = 3\sin(x + \pi)$$

- Plot both `y1` and `y2`.
- Optional: Set the `linestyle` and `color` of the plots individually.

7 Plotting distributions using box or violin plots

- Create two arrays of 100 normally distributed numbers. The two arrays should have different means and standard deviations.
- Create a matplotlib figure and plot the two distributions using either `boxplot` or `violinplot`.
- `show` your plot and save it to file using either png or pdf format.

8 Optional

- Add text to your plots.
- Style your plots such that they get *e.g.* a dark background.