Numerical optimization: Assignment 1

Deadline: the lab on 2024.03.04

The goal of this assignment is to introduce you to Python with NumPy and Matplotlib, or refresh your memory. You can get up to 5 points.

1. $\boxed{\text{1 point}}$ Using NumPy, write a script which defines the following variables:

   (a) `a = [1, 2, 3, ..., 100]` — a vector of integers from 1 to 100

   (b) `b = [1, 3, 5, ..., 99]` — a vector of odd integers from 1 to 99

   (c) `c =` $[-1.00\pi, -0.99\pi, \ldots, -0.01\pi, 0, 0.01\pi, \ldots, 0.99\pi, 1.00\pi]$

   (d) `d =` $[-1.00\pi, -0.99\pi, \ldots, -0.01\pi, 0.01\pi, \ldots, 0.99\pi, 1.00\pi]$

   (e) `e =` $[e_1, e_2, \ldots, e_{100}]$ — where $e_i = sin(i)$ if $sin(i) > 0$, else $e_i = 0$

   (f) `A` — a 10x10 matrix containing the integers from 1 to 100, the first row starting with 1, 2, ..., the second with 11, 12, ...etc. (hint: try `reshape`)

   (g) `B` — an upper diagonal matrix filled with ones

   (h) `C` — a 2x100 matrix which has the elements $c_{1i} = 1 + 2 + \ldots + i$ in its first row, and $c_{2j} = j!$ in its second row

2. $\boxed{\text{1 point}}$ Look into `numpy.random` and `matplotlib`.

   (a) Generate 10,000 numbers from a uniform distribution over [-1, 1]. Make a histogram with 100 intervals. Compare the histogram with the probability density function. Repeat the procedure for 100,000 numbers.

   (b) Generate 10,000 points $[x\ y]^T \in \mathbb{R}^2$, where x is from a normal distribution $N(5, 2)$ and y is from a normal distribution $N(3, 1)$. Plot the points and the probability density function. Repeat the procedure for 100,000 samples.

3. $\boxed{\text{1 point}}$ Let x, y, w be column vectors of set dimension d. Compute (without using `for` or `while`):

   (a) the Euclidean length of vector x

   (b) the weighted average of vector x with weights w

   (c) the Euclidean distance between vectors x and y

   (d) the dot product of vectors x and y (`do NOT use the built-in function`)

4. $\boxed{\text{2 points}}$

   (a) Load the IRIS dataset. You can do it easily with SciKit:
   `from sklearn import datasets`
   `iris = datasets.load_iris()`
   Examine what is contained in `iris.data`, `iris.target`, `iris.feature_names`, `iris.target_names`.

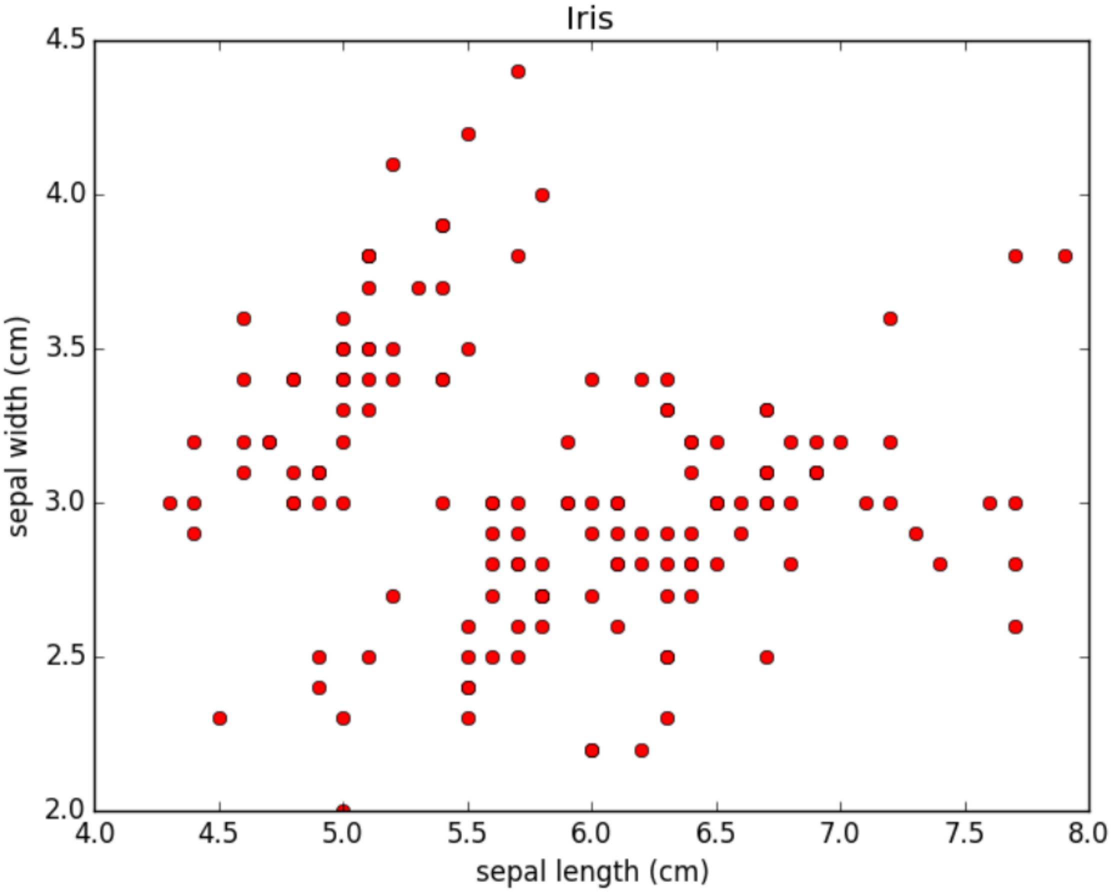
   (b) Plot the data in a similar way as below (color, type of markers, axes description, title).

   (c) Change the range of the axes: the X axis should show the values from 3 to 9, and the Y axis should show the values from 1 to 5.

   (d) Change the tickmarks on the axes so that they show only the integer numbers.

(e) Mark every iris species with a different color.

(f) Save the plot to the file `assignment1.png`.


Iris
sepal width (cm)
sepal length (cm)