Numerical Methods (S210007)   NM − 2022/23
I. Roko

Assignment #1   *The answers should be submitted until Wednesday October 12, 16:00*

Exercise 1

By using the `for` loop perform the following operations and compare your results with those obtained by using matrix operators

1. Let's consider the vector $x = (1\ 8\ 3\ 9\ 5\ 1)$

   - Calculate the vector $v = x^2$ (verify with `x^2`).
   - Calculate the sum of the elements of x (verify with `sum(x)`).
   - Calculate the cumulative sum of the elements of x (verify with `cumsum(x)`).
   - Calculate the product of the elements of x (verify with `prod(x)`).

2. Create a random matrix A of dimensions 3×4 (use `matrix`, `runif`). By going through each element of the matrix (double `for` loop), replace by 0 the values lower than 0.2, and by 1 those greater or equal to 0.2 (verify with `1*(A >= 0.2)`).

3. We have $x = (4\ 1\ 6)$ and $y = (6\ 2\ 7)$. Compute the following matrices or vectors using a `for` loop:

   - $a_{ij} = x_i y_j$ (verify with `outer(x,y)`).
   - $b_{ij} = x_i/y_j$ (verify with `outer(x,y,'/')`).
   - $d_i = x_i y_i$ and sum the elements of d (verify with `sum(x*y)`).
   - e_i equal the smallest value between x_i and y_i (verify with `pmin(x,y)`).

Exercise 2

Using R:

- Let's consider the vector $a = (3\ 8\ 9\ 7\ 6\ 5)$. Following the previous exercise, and with the help of a `for` loop, compute:

$$y = a_1^2 + a_2^2 + a_3^2 + \ldots + a_6^2 \quad .$$

  How can we do the same computation without using a `for` loop ?

- With and without the use of a `for` loop, compute:

$$y = \sum_{i=1}^{6} a_i^{6-i} i \quad .$$

Exercise 3

This exercise aims to give a general idea about logical operators (operators returning values `TRUE` or `FALSE`), and also their use for indexation.

1. Create the following vectors :

   $x = \begin{pmatrix} -4, & -2, & 0, & \ldots, & 12, & 14 \end{pmatrix}$

   and

   $y = \begin{pmatrix} 9, & -3, & 15, & 6, & -8, & 12, & 4, & 0, & -7, & 0 \end{pmatrix}.$

2. Execute the following commands and comment on the results (add a comment at each line):

   ```
   (x > -1) & (x < 4)
   x[x > 2]
   y[x <= 4]
   x[(x < -1) | (x >= 4)]
   y[(x < -1) | (x >= 4)]
   x[y < 0]
   ```

3. Replicate the results of the following commands, by using `for` loops and the instructions `if`, `else`, ... (create separated `for` loops for each of them) :

   ```
   I1 <- x > 2
   I2 <- (x < -1) | (x >= 4)
   I3 <- (x > -1) & (x < 4)
   ```