Matlab/Octave basics - Exercises

Ágnes Baran, Csaba Noszály

Exercise 1

(a) Without typing the elements, create the following vectors:
  (1) $a = (0, 1, \cdots, 30)$
  (2) $b = (2, 4, 6, \ldots, 100)$,
  (3) $c = (2, 1.9, 1.8, \cdots, 0.1, 0)$
  (4) $d = (0, 3, 6, \ldots, 27, 30, -100, 30, 27, \cdots, 6, 3, 0)$
  (5) $e = \left(\frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{20}\right)$
  (6) $f = \left(\frac{1}{2}, \frac{2}{3}, \cdots, \frac{19}{20}\right)$

(b) You are given the vector $x = 1 : 100$. Define the vector
  (1) with elements of x, but in the reverse order
  (2) of the first 5 elements of x
  (3) with elements of x, except the 4th one.
  (4) with elements of x, except the 5.,72 and 93. ones.
  (5) of the odd indexed elements of x
  (6) of the 2., 5., 17. and 81. elements of x

Exercise 2

Your are given the vector $x = randi(10, 1, 10)$. Without using any loop construct, define the vector y for which

(1) $y(i) = x(i) + 2$

(2) $y(i) = x(i)^2$

(3) $y(i) = 1/x(i)$

(4) $y(i) = \sin(x(i)^3 - 1)$

(5) $y(i) = x(i) - i$

Verify the result when possible.

Exercise 3

Let

$$x = [-1 \quad 4 \quad 0],\ y = [3 \quad -2 \quad 5] \text{ and } A = [-3 \quad 1 \quad -4; 6 \quad 2 \quad -5]$$

For the commands given below describe the result or explain why it cannot be performed!

(1) $z = [x, y]$
(2) $z = [x; y]$
(3) $z = [x', y']$
(4) $z = [x'; y']$
(5) $z = [A; x]$
(6) $z = [A, x]$
(7) $z = [x; A; y]$
(8) $z = [A'; x]$
(9) $z = [A', x]$
(10) $z = [A', x']$
(11) $x + y$
(12) $x + y'$
(13) $A + y$
(14) $A + 2$
(15) $x./y$
(16) $A \hat{} 2$
(17) $A.\hat{} 2$

Exercise 4

Let

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

Construct the matrix B, which is made by

(1) deleting the 1-st row of A

(2) deleting the 2. and 4. columns row of A

(3) deleting the last row and column row of A

(4) putting A next to itself.

(5) transposing A

(6) exchanging the 2. and 4. columns of A

(7) squaring the elements of A

(8) adding 3 to each element of A

(9) taking the square root of the elements of A

(10) taking the sine of the elements of A

(11) by setting a_{12} to -2

(12) by setting the second row of A to $[-1 \quad 0 \quad -2 \quad 3]$

Exercise 5

- Define the matrix below:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 20 & 18 & 16 & 14 & 12 & 10 & 8 & 6 \\ 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 \end{pmatrix}$$

- Examine the results of the following commands:

(1) `sum(A)`
(2) `sum(A,2)`
(3) `reshape(A,6,4)`
(4) `max(A)`
(5) `max(A,[],2)`
(6) `max(A,2)`
(7) `flipud(A)`
(8) `fliplr(A)`
(9) `size(A)`
(10) `length(A)`

Exercise 6

Let $x = [0 \quad -1 \quad 2 \quad 0 \quad 4 \quad 4]$ and $y = [-1 \quad -2 \quad 3 \quad 1 \quad 0 \quad 4]$. What is the result of the evaluation of the following expressions?

(1) $x == y$
(2) $x <= y$
(3) $x > y$
(4) $x > 0$
(5) $y <= 3$
(6) $x|y$
(7) $x\&y$

Exercise 7

For the vectors of the previous exercise, what is the result of:

(1) `find` $(x == y)$
(2) `find` $(x <= y)$

Exercise 8

Let `a=rand(1,20)` Create the vector b containing the elements of with $a_i > 0.5$!

Exercise 9

Find an appropriate loopless expression that creates the following matrices in a given size:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 3 & 0 & 3 & 0 & 0 \\ 0 & 6 & 0 & 4 & 0 \\ 0 & 0 & 9 & 0 & 5 \end{pmatrix}, \quad B = \begin{pmatrix} -2 & 2 & 2 & 2 & 2 \\ 0 & -2 & 2 & 2 & 2 \\ 0 & 0 & -2 & 2 & 2 \\ 0 & 0 & 0 & -2 & 2 \\ 0 & 0 & 0 & 0 & -2 \end{pmatrix}$$

Exercise 10

Find an expression, which extends (at the end) the rows of a given matrix by the `mean` of the numbers in the row.

Exercise 11

Find an expression, which extends the columns (at the end) of a given matrix by the `sum` of the numbers in the row.

Exercise 12

Examine the following snippets:

```
clear all
N=100000; % a huge vector
x=rand(1,N);

disp('a naive way:')
tic
for i=1:N
   y(i)=x(i)+i;
end
toc
```

```
disp('a little bit smarter:')
tic
yy=zeros(1,N);
for i=1:N
   yy(i)=x(i)+i;
end
toc

disp('the expected way:')
tic
yyy=x+1;
toc
```

`Lesson:` Use the built-in features instead of loops! The resulting code is usually faster/shorter/more clean and easier to maintain/understand/modify and less error-prone.