**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii: Tehnologia informației**

RAPORT

# LUCRARE DE LABORATOR NR. 1

# la Programarea Declarativă

**Tema: Introducere în R***.*

A efectuat:

st. gr. TI-211 Popa Cătălin

A verificat: lect.dr. Mariana Rusu

UTM, Chișinău 2023

**Exercițiul 1**

Rulați următorul cod și interpretați ceea ce este afișat.

**1.**

x <- *c*(2:4, 9:13)

**Rezultat : x = 2 3 4 9 10 11 12 13**

Aceasta linie creează un vector numit X care conține elementele de la 2 până la 4 și de la 9 până la 13.

**2.**

y <- *c*("b", "c", "E")

**Rezultat : y = "b" "c" "E"**

Aici se crează un vector y care conține trei caractere: "b" "c" "E".

**3.**

x[5]

**Rezultat : 10**

Returnează al 5-lea element al vectorului x, care este 10.

**4.**

y[2:3]

**Rezultat : "c" "E"**

Returnează al 2-lea și al trei-lea element ale vectorului y.

**5.**

y[*c*(2,2,3)]

**Rezultat : "c" "c" "E"**

Returnează elementele 2,2 și 3 ale vectorului y.

**6.**

x[50]

**Rezultat : NA**

Încearcă să acceseze al 50-lea element al vectorului x. Din motiv că vectorul x are doar 8 elemente, se va returna un rezultat gol sau NA(nu este valabil).

**7.**

x[-5]

**Rezultat : 2 3 4 9 11 12 13**

Returnează vectorul x, cu al 5-lea element eliminat.

**8.**

x[3]

**Rezultat : 4**

Returnează al 3-lea element al vectorului x, care este 4.

**9.**

x[*c*(2,2,5:7)]

**Rezultat : 3 3 10 11 12**

Returnează elementele 2,2,5,6 și 7 ale vectorului x.

**10.**

x[6:1]

**Rezultat : 11 10 9 4 3 2**

Returnează elementele de la 6 până la 1 ale vectorului x, în ordine inversă.

**11.**

x[-(1:4)]

**Rezultat : 10 11 12 13**

Acesta va elimina primele 4 elemente ale vectorului x și va afișa restul.

**12.**

x[-*c*(1,4)]

**Rezultat : 10 11 12 13**

Returneaza elementele vectorului x, fără elementul 1 și 4.

**Exercițiul 2**

Scrieți o funcție pentru a calcula varianța unui vector.

varianta <- function(vector) {  
 varianta <- *var*(vector)  
 *return*(varianta)  
}  
# Folosirea funcției  
rez <- *varianta*(x)  
*cat*("Varianta vectorului =", rez)

**Rezultat : Varianta vectorului = 18.85714**

Mai întâi am creat funcția **varianta**, care foloseste funcția predefinită var() și ca parametru primește un vector. Mai jos am implementat un caz când se folosește funcția respectivă cu vectorul x.

**Exercițiul 3**

Rulați următorul cod și interpretați ceea ce este afișat.

**1.**

vector=1:10

**Rezultat : vector = 1 2 3 4 5 6 7 8 9 10**

Aceasta linie creează un vector numit vector cu valorile de la 1 la 10.

**2.**

matrice1=*matrix*(vector, ncol=2)

Se crează matricea cu numele matrice1, care folosește vectorul precedent, pentru a prea lua valorile de la 1 la 10. **ncol = 2,** specifică numărul de coloane.

**3.**

matrice1

**Rezultat :**

**[1,] 1 6**

**[2,] 2 7**

**[3,] 3 8**

**[4,] 4 9**

**[5,] 5 10**

Returnează matricea creată la punctul 3.

**4.**

matrice2=*matrix*(1:10,nrow=2,byrow=T)

Crează matricea 2, care ia valorile de la 1 la 10, pe două rânduri și argumentul **byrow = T**, indicând că valorile sunt așezate pe rând.

**5.**

matrice2

**Rezultat :**

**[,1] [,2] [,3] [,4] [,5]**

**[1,] 1 2 3 4 5**

**[2,] 6 7 8 9 10**

Afișează matricea 2.

**6.**

m=*matrix*(1:4,nrow=3,ncol=3)

**Rezultat : Warning in matrix(1:4, nrow = 3, ncol = 3) :**

**data length [4] is not a sub-multiple or multiple of the number of rows [3]**

Încearcă să creze matricea **m** cu 3 rânduri și 3 coloane, dar deoarece numărul total de elemente nu se potrivește cu dimensiunea specifică a matricei, am primit o avertizare și matricea a luat valorile în ordine secvențială de la stânga la dreapta și de sus în jos.

**7.**

m

**Rezultat :**

**[,1] [,2] [,3]**

**[1,] 1 4 3**

**[2,] 2 1 4**

**[3,] 3 2 1**

Afișează matricea m.

**8.**

*print*(matrice2)

**Rezultat :**

**[,1] [,2] [,3] [,4] [,5]**

**[1,] 1 2 3 4 5**

**[2,] 6 7 8 9 10**

La fel afișează matricea 2.

**9.**

*dim*(matrice1)

**Rezultat : 5 2**

Returnează mărimea matricei, care este de 5 rânduri și 2 coloane.

**10.**

*ncol*(matrice1)

**Rezultat : 2**

Returnează numărul de coloane a matricei.

**11.**

*nrow*(matrice1)

**Rezultat : 5**

Returnează numărul de rânduri.

**12.**

resultat= matrice1 %\*% matrice2

**Rezultat : --**

Este o încercare de a face o înmulțire între două matrice, dar este o eroare deoarece numărul de coloane ale matricei 1 nu se potrivește cu numărul de rânduri ale matricei 2.

**Exercițiul 4**

Operații pe matrice: Exemple.

- Funcțiile dim(), ncol(), nrow() arată dimensiunile unei matrice.

**1.**

*dim*(matrice1)

**Rezultat : 5 2**

Numărul de rânduri și coloane a matricei.

**2.**

*ncol*(matrice1)

**Rezultat : 2**

Afișează numărul de coloane.

**3.**

*nrow*(matrice1)

**Rezultat : 5**

Afișează numărul de coloane.

**4.**

resultat= matrice1 %\*% matrice2

**Rezultat :**

**1 37.0000 44.0000 51.0000 58.0000 65.00002 44.0000 53.0000 62.0000 71.0000 80.00003 51.0000 62.0000 73.0000 84.0000 95.00004 58.0000 71.0000 84.0000 97.0000 110.0005 65.0000 80.0000 95.0000 110.000 125.000**

Produsul a două matrici se scrie cu operatorul %\*%.

**5.**

*t* (matrice1)

**Rezultat :**

**[,1] [,2] [,3] [,4] [,5]**

**[1,] 1 2 3 4 5**

**[2,] 6 7 8 9 10**

Funcția **t()** traspune o matrice. Ea transpune o matrice, schimbând rândurile cu coloanele și coloanele cu rândurile.

**6.**

*diag* (resultat)

**Rezultat : [1] 37 53 73 97 125**

Funcția **diag()** permite preluarea diagonalei unei matrice pătrate.

**7.**

*diag*(*c*(3,2,4))

**Rezultat :**

**[,1] [,2] [,3]**

**[1,] 3 0 0**

**[2,] 0 2 0**

**[3,] 0 0 4**

Construiește o matrice diagonală dintr-un vector.

**8.**

vecteur1= *c*(8,3,2)

**Rezultat : S-a create vectorul cu valorile 8,3,2.**

Funcțiile **rbind()** și **cbind()** permit concatinarea pe rând sau coloană vectori sau matrice.

**9.**

vecteur2 = *c*(23,6,9)

**Rezultat : S-a create vectorul cu valorile 23,6,9.**

**10.**

res=*rbind*(vecteur1,vecteur2)

**Rezultat X1 X2 X3**

**vecteur1 8.00000 3.00000 2.00000vecteur2 23.0000 6.00000 9.00000**

Concatinează vectorii pe rând.

**11.**

vecteur3 = *c*(2,4)

**Rezultat : Crează un vector nou cu valorile 2,4**.

**12.**

*cbind*(res,vecteur3)

**Rezultat :**

**vecteur3**

**vecteur1 8 3 2 2**

**vecteur2 23 6 9 4**

Are loc concatinarea la vectorului **vecteur3** pe coloană la matricea **res**.

**13.**

*eigen*(resultat)

**Rezultat:**

**eigen() decomposition**

**$values**

**[1] 3.817254e+02 3.274605e+00 5.901666e-15 -4.680421e-15 -4.741161e-14**

**$vectors**

**[,1] [,2] [,3] [,4] [,5]**

**[1,] -0.3042621 -0.71233741 0.0000000 0.0000000 6.324555e-01**

**[2,] -0.3707409 -0.40317635 -0.1964879 0.5112656 -6.324555e-01**

Pentru a diagonalize o matrice pătrată, folosim funcția **eigen()** care returnează o listă de două componente: **$ values** care conțin valorile proprii, **$ vectors** care conțin vectorii proprii.

**Exercițiul 5**

Rulați următorul cod și interpretați ceea ce este afișat.

**1.**

resultat[1,]

**Rezultat : 37 44 51 58 65**

Afișează rândul 1 din matricea **resultat**.

**2.**

resultat[, *c*(2,2,1)]

**Rezultat :**

**[,1] [,2] [,3]**

**[1,] 44 44 37**

**[2,] 53 53 44**

**[3,] 62 62 51**

**[4,] 71 71 58**

**[5,] 80 80 65**

Se extrag coloanele 2, 2, 1 din matricea **resultat**.

**3.**

resultat[-1, ]

**Rezultat :**

**[,1] [,2] [,3] [,4] [,5]**

**[1,] 44 53 62 71 80**

**[2,] 51 62 73 84 95**

**[3,] 58 71 84 97 110**

**[4,] 65 80 95 110 125**

Exclude primul rând din matricea **resultat**.

**4.**

resultat [1 :2,-1]

**Rezultat :**

**[,1] [,2] [,3] [,4]**

**[1,] 44 51 58 65**

**[2,] 53 62 71 80**

Selectarea rândurilor 1 și 2 și toate coloanele, cu excepția primei coloane.

**5.**

resultat [resultat>51]

**Rezultat : 58 65 53 62 71 80 62 73 84 95 58 71 84 97 110 65 80 95 110 125**

Returnează toate elementele din matricea **resultat** care sunt mai mare decât 51.

**6.**

*matrix*(vector,nrow=2)

**Rezultat :**

**[,1] [,2] [,3] [,4] [,5]**

**[1,] 1 3 5 7 9**

**[2,] 2 4 6 8 10**

Creează o matrice cu 2 rânduri folosind vectorul **vector**  cu 2 rânduri.

**7.**

*matrix*(vector,nrow=2, byrow=T)

**Rezultat :**

**[,1] [,2] [,3] [,4] [,5]**

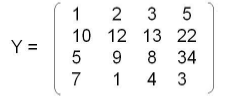
**[1,] 1 2 3 4 5**

**[2,] 6 7 8 9 10**

Aici, o altă matrice cu 2 rânduri este creată folosind vectorul "vector". Argumentul byrow=T indică faptul că elementele vor fi așezate pe rând. Aceasta poate afecta modul în care sunt așezate valorile în matrice, în funcție de numărul de elemente din vector. Dacă vectorul are suficiente elemente pentru a umple cele două rânduri.

**Exercițiul 6**

**1.** Creați următoarea matrice:



new\_vector <- *c*(1, 10, 5, 7, 2, 12, 9, 1, 3, 13, 8, 4, 5, 22, 34, 3)  
new\_matrix <- *matrix*(new\_vector, nrow = 4, ncol = 4)

**2.** Vizualizați elementul Y conținut în:

**a.** Al treilea rând și a doua coloană.

element <- new\_matrix[3,2]  
*print*(element)

**b.** Al doilea rând de Y

new\_matrix[2, ]

**c.** A patra coloană a lui Y

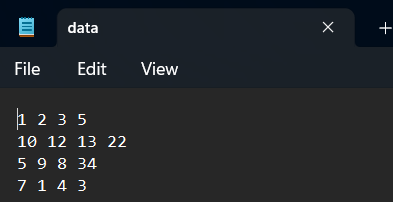
new\_matrix[, 4]

**d.** Matricea obținută după înlăturarea primului rând și a celei de-a doua coloane

new\_matrix[-1, -2]

**3.** Exportați matricea Y într-un fișier .txt numit data.txt.

*write.table*(new\_matrix, file="data.txt", row.names=FALSE, col.names=FALSE)



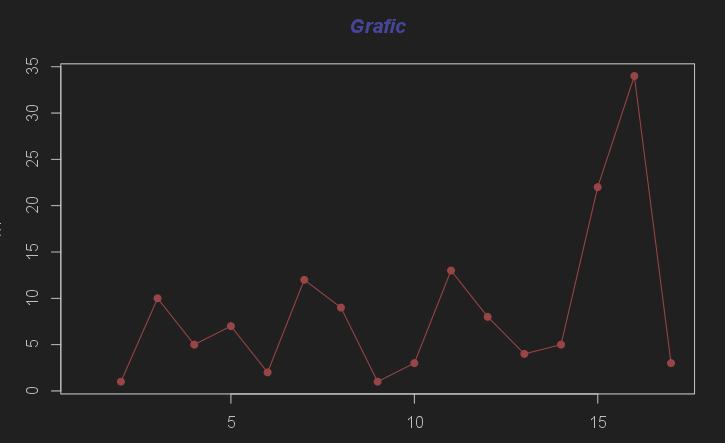
**Figura 1. Matricea exportată în fișier de tip txt**.

**Exercițiul 7**

Scrieți o funcție care permite citirea datelor dintr-un fișier, urmată de un grafic. Utilizați setul de date data.txt pentru a test funcția creată.

creare\_grafic <-function(file){  
 x <- *read.table*(file)  
 y <- *plot*(x, col="red", type="o", pch=19)  
 *title*(main="Grafic", col.main="Blue", font.main=4)  
 *return* (y)  
}  
*creare\_grafic*("data.txt")

**Răspuns:**



**Figura 2. Grafic.**

**Exercițiul 8**

1. Încărcați datele **airquality**.

*data*(airquality)

1. Explicați cele șase variabile.

**Ozone**: Concentrația de ozon în aer (unitatea de măsură nu este specificată, dar poate fi ppm - părți per milion).

**Solar.R**: Radiația solară în lumeni pe metru pătrat.

**Wind**: Viteza vântului în mile pe oră.

**Temp**: Temperatura în grade Fahrenheit.

**Month**: Luna (valori de la 5 la 9 reprezentând lunile mai - septembrie).

**Day**: Ziua (valori de la 1 la 31).

1. Calculați statisticile principale ale bazei de date folosind funcția **summary**.

*summary*(airquality)

Ozone Solar.R Wind Temp Month

Min. : 1.00 Min. : 7.0 Min. : 1.700 Min. :56.00 Min. :5.000

1st Qu.: 18.00 1st Qu.:115.8 1st Qu.: 7.400 1st Qu.:72.00 1st Qu.:6.000

Median : 31.50 Median :205.0 Median : 9.700 Median :79.00 Median :7.000

Mean : 42.13 Mean :185.9 Mean : 9.958 Mean :77.88 Mean :6.993

3rd Qu.: 63.25 3rd Qu.:258.8 3rd Qu.:11.500 3rd Qu.:85.00 3rd Qu.:8.000

Max. :168.00 Max. :334.0 Max. :20.700 Max. :97.00 Max. :9.000

NA's :37 NA's :7

Day

Min. : 1.0

1st Qu.: 8.0

Median :16.0

Mean :15.8

3rd Qu.:23.0

Max. :31.0

1. Calculați separate media, mediana și abaterea standart a variabilei Temp folosind comenzile corespunzătoare.

*mean*(airquality$Temp) --- media  
*median*(airquality$Temp) --- mediana  
*sd*(airquality$Temp) --- abaterea

**Răspuns**

77.88235

79

9.46527

5. Calculați varianța și scrieți o funcție pentru a calcula abaterea standart.

varianta <- *var*(airquality$Temp)

**Răspuns:**

varianta = 89.59133

**Cu funcție:**

abaterea\_standart <- function(data) {  
 varianta <- *var*(data)  
 ab <- *sqrt*(varianta)  
 *return*(ab)  
}  
abaterea <- *abaterea\_standart*(airquality$Temp)

**Răspuns:**

abaterea = 9.46527

6. Extras:

a) A doua linie :

airquality[2, ]

b) A treia coloană :

airquality[ , 3]

c) Liniile 1,2 și 4 cu o singură comandă c() :

airquality[*c*(1, 2, 4), ]

d) Liniile 2-6 cu comanda ‘:’ :

airquality[2:6, ]

e) Toate, cu excepția coloanelor 1 și 2 :

airquality[, -*c*(1, 2)]

f) Toate liniile cu temperature mai mare de 900:

airquality[airquality$Temp > 90, ]

**Concluzie:**

Realizarea acestei lucrări ma ajutat la introducerea inițială în limbajul de programare R. Am însușit funcțiile principale de lucru și metodele de creare a unei matrice sau vector. La fel am folosit vectori pentru a crea matrice. Am creat o matrice și am introdus datele intr-un fișier, iar apoi le-am extras pentru a crea o diagramă.