Universitatea Tehnica a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software și Automatică

# Raport

Lucrarea de laborator Nr.1

Programarea Declarativa

Tema:Introducere in R

A efectuat: St. gr. TI-216

Vlașițchi Ștefan

A verificat: Lect. Univ.

Chișinău 202**3**

**Scopul:**

**Exercițiul 1**

Rulați următorul cod și interpretați ceea ce este afișat:

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

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

> x[5]

> y[2:3]

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

> x[50]

> x[-5]

> x[3]

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

> x[6 :1]

> x[-(1 : 4)]

> x [-c(1,4)]

1. **x<-c(2:4,9:13)** - Se creează un vector numit "x" care conține elementele 2, 3, 4, 9, 10, 11, 12 și 13.

2. **y<-c("b", "c", "E")** - Se creează un vector de caractere numit "y" care conține literele "b", "c" și "E".

3. **x[5]** - Afișează al 5-lea element din vectorul "x". Rezultatul va fi 10.

4. **y[2:3]** - Afișează elementele 2 și 3 din vectorul "y". Rezultatul va fi un nou vector care conține "c" și "E".

5. **y[c(2,2,3)]** - Afișează elementele de la pozițiile 2, 2 și 3 din vectorul "y". De fapt, elementul de la poziția 2 este "c", așa că rezultatul va fi un nou vector care conține "c", "c" și "E".

6. **x[50]** - Afișează al 50-lea element din vectorul "x". Cum vectorul "x" are doar 8 elemente, iar indexul 50 este în afara intervalului, va returna "NA", semnificând "Not Available".

7. **x[-5]** - Afișează toate elementele din vectorul "x" exceptând elementul de la poziția 5. Rezultatul va fi un nou vector fără elementul 10.

8. **x[3]** - Afișează al 3-lea element din vectorul "x", care este 4.

9. **x[c(2,2,5:7)]** - Afișează elementele de la pozițiile 2, 2, 5, 6 și 7 din vectorul "x". Rezultatul va fi un nou vector cu elementele 3, 3, 11, 12 și 13.

10. **x[6:1]** - Afișează elementele de la pozițiile 6 la 1 din vectorul "x". Rezultatul va fi un nou vector cu elementele 11, 10, 9, 4 și 3, în această ordine.

11. **x[-(1:4)]** - Afișează toate elementele din vectorul "x" exceptând cele de la pozițiile 1 până la 4. Rezultatul va fi un nou vector fără primele patru elemente, adică 10, 11, 12 și 13.

12. **x[-c(1,4)]** - Afișează toate elementele din vectorul "x" exceptând cele de la pozițiile 1 și 4. Rezultatul va fi un nou vector fără elementele 2 și 9.

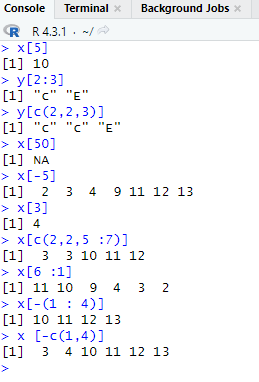


Figura 1 – Rezultatul exercitiului 1

Exercițiul 2

Scrieți o funcție pentru a calcula varianța unui vector.

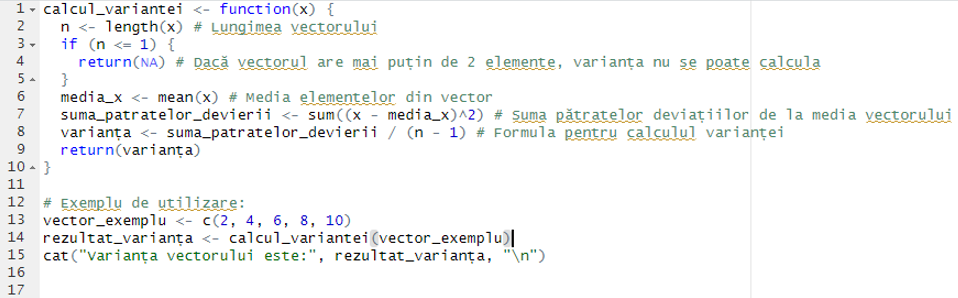


Figura 2 – Functia pentru calculul variatiei

Unde:

xi​ - reprezintă fiecare valoare individuală din setul de date.

μ - reprezintă media setului de date.

n - reprezintă numărul de valori din setul de date.

**Exercitiul 3**

Rulați următorul cod și interpretați ceea ce este afișat:

> vector=1:10

> matrice1=matrix(vector, ncol=2)

> matrice1

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

> matrice2

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

> m

> print(matrice2)

> dim(matrice1)

> ncol(matrice1)

> nrow(matrice1)

> resultat= matrice1 %\*% matrice2

1. **vector=1:10** - Se creează un vector numit "vector" care conține numerele de la 1 la 10.
2. **matrice1=matrix(vector, ncol=2)** - Se creează o matrice numită "matrice1" cu două coloane folosind vectorul "vector". Elementele vectorului sunt distribuite în matrice în ordine, umplând coloanele înainte de a trece la următoarea linie.
3. **matrice1** - Afișează matricea "matrice1". Rezultatul va arăta ca o matrice cu două coloane și cinci rânduri.
4. **matrice2=matrix(1:10,nrow=2,byrow=T)** - Se creează o altă matrice numită "matrice2" cu două rânduri și cinci coloane folosind vectorul 1 până la 10. Opțiunea **byrow=T** specifică faptul că valorile sunt umplute pe rânduri înainte de a trece la următoarea coloană.
5. **matrice2** - Afișează matricea "matrice2". Rezultatul va arăta ca o matrice cu două rânduri și cinci coloane, în care valorile sunt umplute pe rânduri.
6. **m=matrix(1:4,nrow=3,ncol=3)** - Se creează o altă matrice numită "m" cu trei rânduri și trei coloane folosind vectorul 1 până la 4. Deoarece vectorul are doar patru elemente, matricea va fi umplută cu aceste valori și va genera un avertisment că "lungimea datelor va fi replicată".
7. **m** - Afișează matricea "m". Rezultatul va arăta ca o matrice cu trei rânduri și trei coloane, în care valorile din vectorul 1 până la 4 sunt replicare pentru a umple matricea.
8. **print(matrice2)** - Afișează matricea "matrice2". Rezultatul va fi similar cu afișarea anterioară a matricei "matrice2".
9. **dim(matrice1)** - Afișează dimensiunea matricei "matrice1". Rezultatul va fi **[5, 2]**, indicând că matricea are 5 rânduri și 2 coloane.
10. **ncol(matrice1)** - Afișează numărul de coloane din matricea "matrice1". Rezultatul va fi 2.
11. **nrow(matrice1)** - Afișează numărul de rânduri din matricea "matrice1". Rezultatul va fi 5.
12. **resultat= matrice1 %\*% matrice2** - Calculează produsul matricial între "matrice1" și "matrice2" și îl stochează în "resultat". Pentru a face acest lucru, matricele trebuie să aibă dimensiuni compatibile, adică numărul de coloane al primei matrice să fie egal cu numărul de rânduri al celei de-a doua matrice. Rezultatul este o nouă matrice care reprezintă produsul matricial.

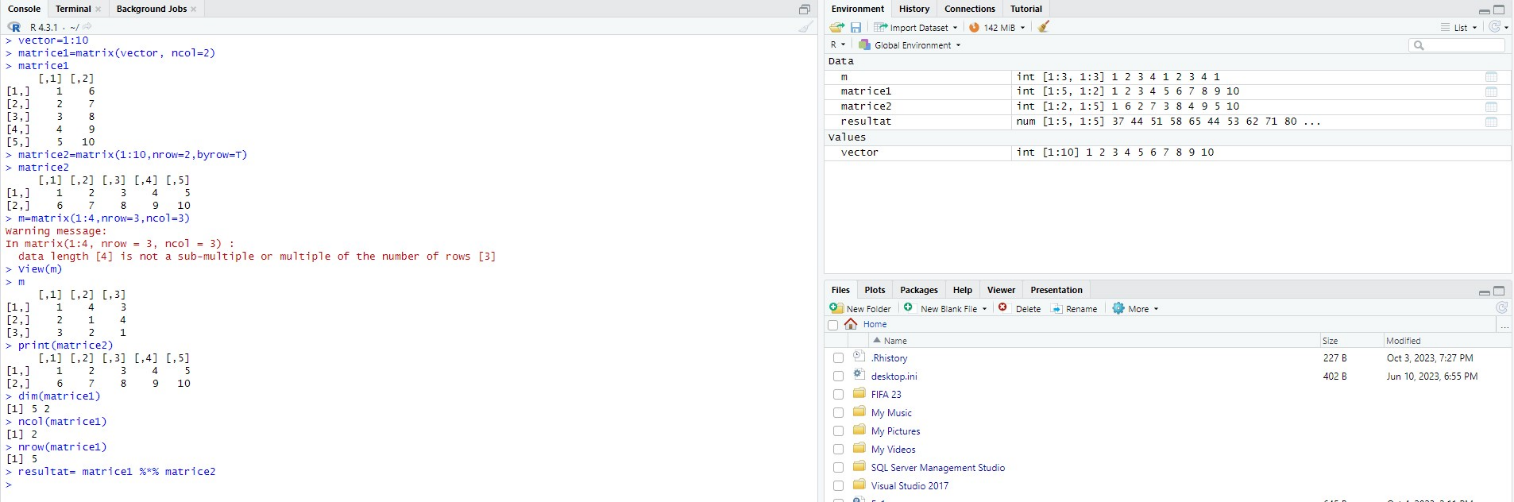


Figura 3 – Rezultatul pentru exercitiul 3

**Exercițiul 4**

Operații pe matrice:

Exemple - Funcțiile **dim(),** **ncol(),** **nrow()** arată dimensiunile unei matrice:

> dim(matrice1)

> ncol(matrice1)

> nrow(matrice1) - Produsul a două matrici se scrie cu operatorul **%\*%.**

> resultat= matrice1 %\*% matrice2 - Funcția **t**() transpune o matrice

> t (matrice1) - Funcția **diag**() permite preluarea diagonalei unei matrice pătrate sau construirea unei matrice diagonale dintr-un vector.

>diag (resultat)

>diag(c(3,2,4)) - Funcțiile **rbind**() și **cbind**() permit concatenarea pe rând sau coloană vectori sau matrice.

> vecteur1= (8,3,2)

>vecteur2=(23,6,9)

> res=rbind(vecteur1,vecteur2)

> vecteur3=c(2,4)

> cbind(res,vecteur3) - Pentru a diagonaliza o matrice pătrată, folosim funcția **eigen** () care returnează o listă de două componente: **$ values** care conțin valorile proprii, **$** **vectors** care conțin vectorii proprii.

> eigen(resultat)

1. Dimensiunea unei matrice și numărul de coloane și rânduri:

- **dim(matrice1)** - Această funcție returnează dimensiunea matricei "matrice1". De obicei, rezultatul va fi de forma `[nrânduri, ncoloane]`.

- **ncol(matrice1)** - Returnează numărul de coloane al matricei "matrice1".

- **nrow(matrice1)** - Returnează numărul de rânduri al matricei "matrice1".

2. Produsul a două matrici:

- **resultat = matrice1 %\*% matrice2** - Calculează produsul matricial între "matrice1" și "matrice2" și stochează rezultatul în "resultat".

3. Transpunerea unei matrice:

- **t(matrice1)** - Această funcție returnează matricea transpusă a lui "matrice1", adică coloanele devin rânduri și viceversa.

4. Diagonala unei matrice și construirea unei matrice diagonale:

- **diag(resultat)** - Această funcție returnează diagonala matricei "resultat". Dacă matricea nu este pătrată, va returna eroare.

- **diag(c(3,2,4))** - Această funcție creează o matrice diagonală cu valorile 3, 2 și 4 pe diagonala principală.

5. Concatenarea de matrice sau vectori:

- **rbind(vecteur1, vecteur2)** - Concatenează vectorii "vecteur1" și "vecteur2" pe verticală, rezultând o nouă matrice cu aceleași coloane și rânduri ca și "vecteur1" și "vecteur2".

- **cbind(res, vecteur3)** - Concatenează matricea "res" cu vectorul "vecteur3" pe orizontală, adăugând o nouă coloană la matrice.

6. Diagonalizarea unei matrice pătrate cu funcția `eigen()`:

- **eigen(resultat)** - Această funcție calculează valorile și vectorii proprii ai matricei pătrate "resultat" și returnează o listă cu două componente: "$values" care conține valorile proprii și "$vectors" care conține vectorii proprii.

**Exercițiul 5**

Rulați următorul cod și interpretați ceea ce este afișat:

> resultat[1,]

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

> resultat[-1, ]

> resultat [1 :2,-1]

> resultat [resultat>51]

>matrix(vector,nrow=2)

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

1. **resultat[1,]** - Afișează prima linie a matricei "resultat". Se selectează toate coloanele din prima linie. Rezultatul va fi o matrice cu o singură linie și toate coloanele din prima linie a matricei "resultat".

2. **resultat[, c(2,2,1)]** - Afișează toate rândurile din matricea "resultat", dar selectează coloanele 2, 2 și 1. Rezultatul va fi o matrice care conține toate rândurile, dar coloanele în această ordine: 2, 2, 1.

3. **resultat[-1, ]** - Afișează toate rândurile din matricea "resultat", exceptând prima linie. Rezultatul va fi o matrice care exclude prima linie a matricei "resultat".

4. **resultat [1 :2,-1]** - Afișează primele două linii din matricea "resultat", dar exclude prima coloană. Rezultatul va fi o matrice cu primele două linii ale matricei "resultat", dar fără prima coloană.

5. **resultat [resultat>51]** - Afișează toate elementele din matricea "resultat" care sunt mai mari decât 51. Rezultatul va fi un vector cu acele elemente din matrice care îndeplinesc condiția dată.

6. **matrix(vector, nrow=2)** - Creează o nouă matrice cu două rânduri, folosind vectorul "vector". Vectorul este umplut în matrice pe coloane în ordine.

7. **matrix(vector, nrow=2, byrow=T**) - Creează o altă matrice cu două rânduri, folosind vectorul "vector". De data aceasta, opțiunea `byrow=T` specifică faptul că valorile din vector sunt umplute pe rânduri înainte de a trece la următoarea coloană.

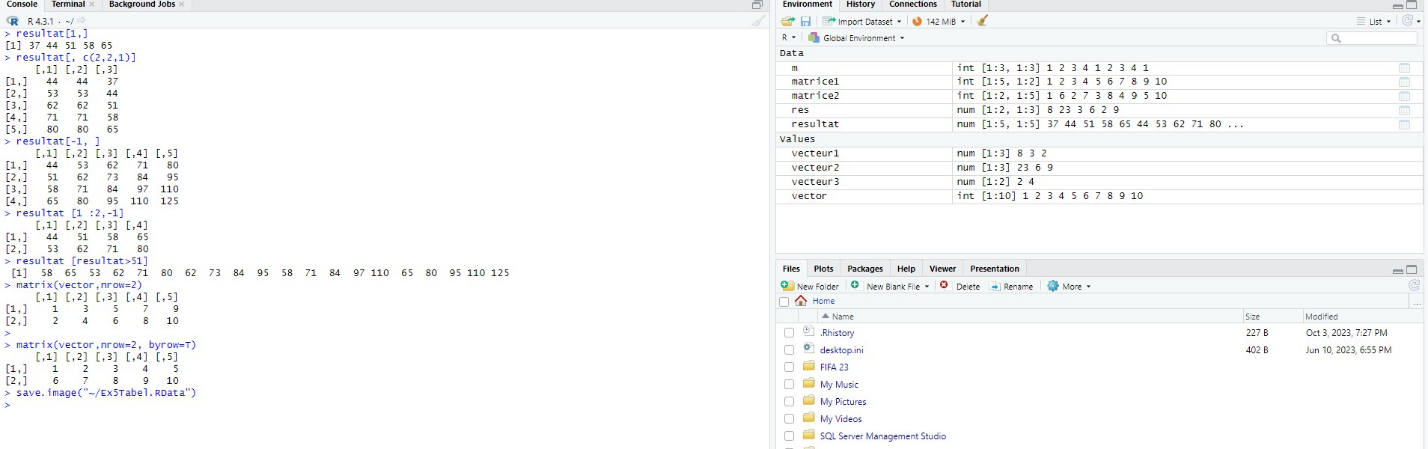
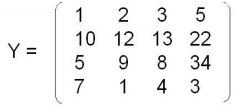


Figura 4 – Rezultatul pentru exercitiul 5

**Exercițiul 6**

1. Creați următoarea matrice Y:
2. Vizualizați elementul Y conținut în:

- Al treilea rând și a doua coloană

- Al doilea rând de Y

- A patra coloană a lui Y

- Matricea obținută după îndepărtarea primului rând și a celei de-a doua coloane

1. Exportați matricea Y într-un fișier .txt numit data.txt

1)

Y <- matrix(c(1, 2, 3, 5,

10, 12, 13, 22,

5, 9, 8, 34,

7, 1, 4, 3), nrow = 4, ncol = 4, byrow = TRUE)

2)

- Al treilea rând și a doua coloană

element\_32 <- Y[3, 2]

element\_32

- Al doilea rând de Y

al\_doilea\_rand <- Y[2, ]

al\_doilea\_rand

- A patra coloană a lui Y

a\_patra\_coloana <- Y[, 4]

a\_patra\_coloana

- Matricea obținută după îndepărtarea primului rând și a celei de-a doua coloane

matrice\_noua <- Y[-1, -2]

matrice\_noua

3)  
Pentru a exporta matricea "Y" într-un fișier .txt numit "data.txt", puteți folosi funcția **write.table()**:

write.table(Y, file = "data.txt", sep = "\t", row.names = FALSE)

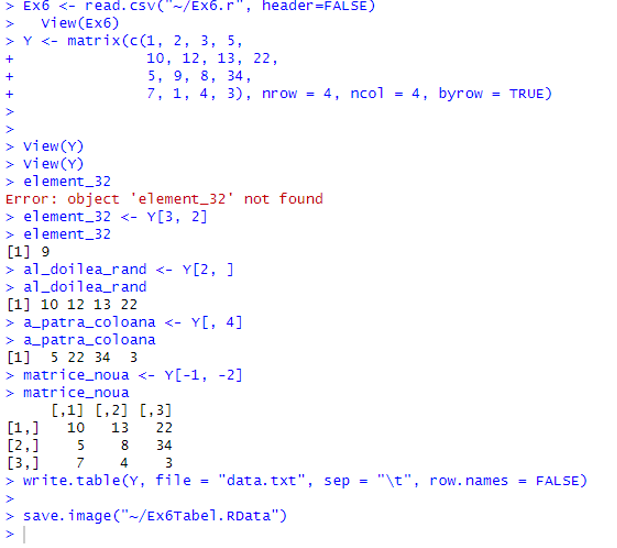


Figura 5 – Rezultatul pentru exercitiul 6

**Exercițiul 7**

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