**Ministerul Educației și Cercetării al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**RAPORT**

Lucrare de laborator nr.1

la cursul „Programarea Declarativă”

**Tema 1:** „ Introducerea in R”

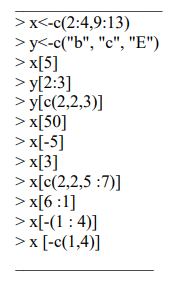
A efectuat : **st. gr. TI-214 Arnaut Roxana**

dA verificat: **M. Rusu**

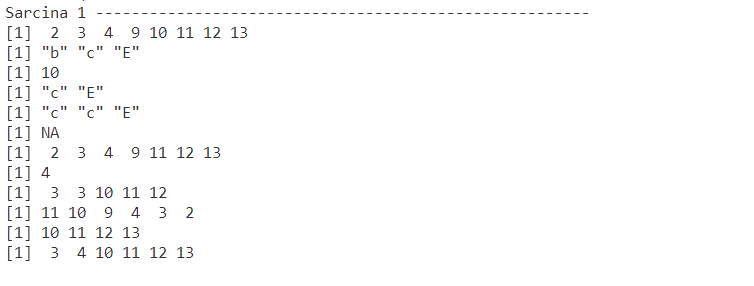
**Chișinău 2023**

**Sarcina 1**

Rulați următorul cod și interpretați ceea ce este afișat:



1. x <- c(2:4, 9:13) a definit un vector x cu elementele 2, 3, 4, 9, 10, 11, 12 și 13.
2. y <- c("b", "c", "E") a definit un vector y cu caracterele "b", "c" și "E".
3. x[5] a accesat al 5-lea element din vectorul x (rezultat: 10).
4. y[2:3] a accesat elementele 2 și 3 din vectorul y (rezultat: "c" și "E").
5. y[c(2, 2, 3)] a accesat elementele de pe pozițiile 2, 2 și 3 din vectorul y (rezultat: "c", "c" și "E").
6. x[50] a încercat să acceseze al 50-lea element din vectorul x, dar vectorul x are doar 8 elemente, deci a returnat NA (valoare lipsă).
7. x[-5] a eliminat al 5-lea element din vectorul x (rezultat: vectorul x fără valoarea 10).
8. x[3] a accesat al 3-lea element din vectorul x (rezultat: 4).
9. x[c(2, 2, 5:7)] a accesat elementele de pe pozițiile 2, 2, 5, 6 și 7 din vectorul x (rezultat: 3, 3, 11, 12 și 13).
10. x[6:1] a accesat elementele de la al 6-lea la primul element din vectorul x, în această ordine inversă (rezultat: 11, 10, 9, 4 și 3).
11. x[-(1:4)] a eliminat primele patru elemente din vectorul x (rezultat: vectorul x fără valorile 2, 3, 4 și 9).
12. x[-c(1, 4)] a eliminat elementele de pe pozițiile 1 și 4 din vectorul x (rezultat: vectorul x fără valorile 2 și 10).



**Sarcina 2**

Scrieți o funcție pentru a calcula varianța unui vector.

calcul\_variante <- function(vector) {

n <- length(vector) # Lungimea vectorului

media <- mean(vector) # Calcularea mediei

varianta <- sum((vector - media) ^ 2) / (n - 1) # Calcularea varianței

return(varianta) # Returnarea valorii varianței

}

vector\_test <- c(1, 2, 3, 4, 5) # Definirea unui vector de test

variance\_result <- calcul\_variante(vector\_test) # Apelarea funcției pentru calcularea varianței

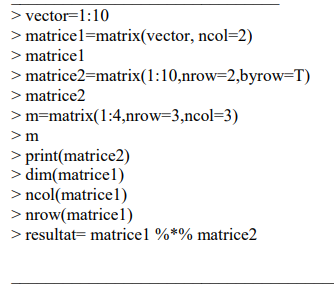
print(variance\_result)

Funcția creată calcul\_variante, calculează varianța unui vector numeric. Varianța reprezintă măsura dispersiei datelor în raport cu media lor. Mai întâi, se determină lungimea vectorului, apoi se calculează media elementelor. Se obține varianța ca suma pătratelor diferențelor dintre fiecare element și media, împărțită la lungimea vectorului minus 1. Funcția returnează valoarea varianței.

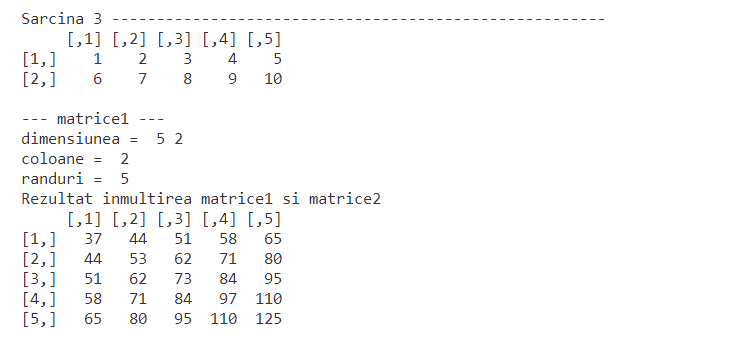


**Sarcina 3**

Rulați următorul cod și interpretați ceea ce este afișat:



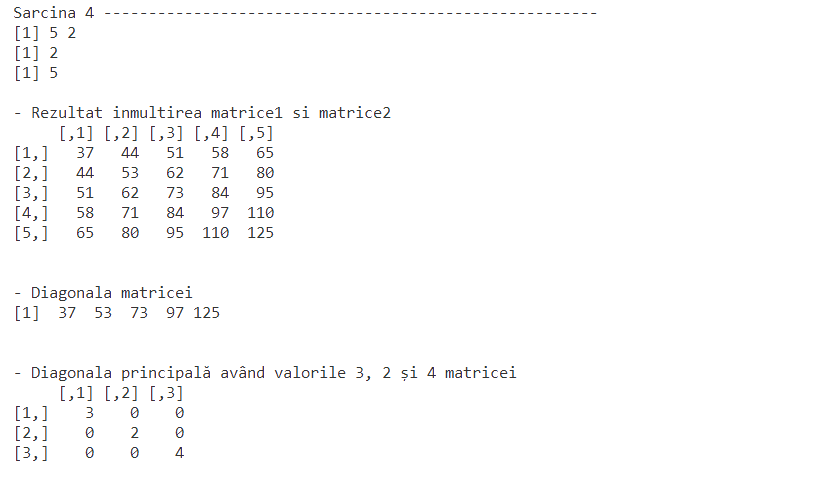
1. vector = 1:10 - Se creează un vector numit "vector" care conține numere de la 1 la 10.
2. matrice1 = matrix(vector, ncol = 2) - Se construiește o matrice numită "matrice1" având 5 rânduri și 2 coloane, folosind elementele din "vector".
3. matrice2 = matrix(1:10, nrow = 2, byrow = T) - Se creează o matrice "matrice2" de dimensiuni 2x5, introducând elementele de la 1 la 10 pe rânduri.
4. m = matrix(1:4, nrow = 3, ncol = 3) - Acest cod încearcă să creeze o matrice 3x3, dar are prea puține elemente (doar 4), așa că completează cu NA (valoare lipsă).
5. resultat = matrice1 %\*% matrice2 - Se efectuează înmulțirea matricială dintre "matrice1" și "matrice2", iar rezultatul este stocat în variabila "resultat".
6. print(matrice2) - Afișează conținutul matricei "matrice2".
7. cat("\n--- matrice1 ---") - Afișează un text indicând începutul secțiunii despre "matrice1".
8. cat("\ndimensiunea = ", dim(matrice1)) - Afișează dimensiunile matricei "matrice1" (5 rânduri, 2 coloane).
9. cat("\ncoloane = ", ncol(matrice1)) - Afișează numărul de coloane al matricei "matrice1" (2).
10. cat("\nranduri = ", nrow(matrice1)) - Afișează numărul de rânduri al matricei "matrice1" (5).
11. cat("\nRezultat inmultirea matrice1 si matrice2\n") - Afișează un text indicând începutul secțiunii despre rezultatul înmulțirii.
12. print(resultat) - Afișează rezultatul înmulțirii matriciale dintre "matrice1" și "matrice2".

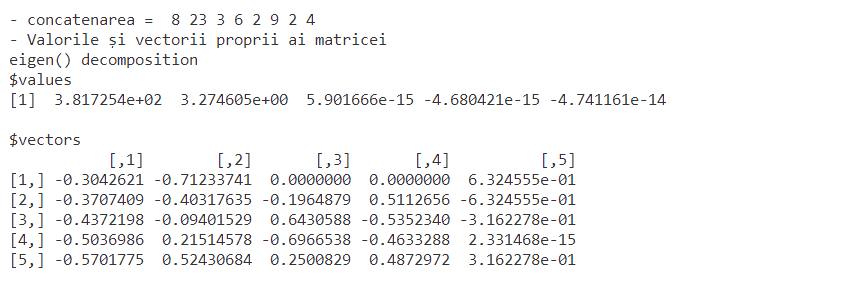


**Sarcina 4**

Operații pe matrice:

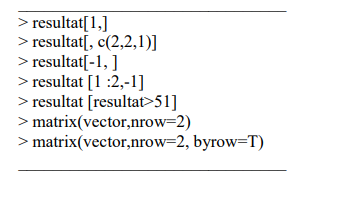
1. print(dim(matrice1)) - Afișează dimensiunile matricei matrice1, adică numărul de rânduri și numărul de coloane.
2. print(ncol(matrice1)) - Afișează numărul de coloane al matricei matrice1.
3. print(nrow(matrice1)) - Afișează numărul de rânduri al matricei matrice1.
4. resultat = matrice1 %\*% matrice2 - Efectuează înmulțirea matricială dintre matrice1 și matrice2, iar rezultatul este stocat în variabila resultat.
5. cat("\n- Rezultat inmultirea matrice1 si matrice2\n") - Afișează un mesaj pentru a indica următoarea secțiune care va afișa rezultatul înmulțirii.
6. print(resultat) - Afișează rezultatul înmulțirii matriciale.
7. cat("\n\n- Diagonala matricei\n") - Afișează un mesaj pentru a indica următoarea secțiune care va trata diagonala matricei.
8. print(diag(resultat)) - Afișează diagonala matricei rezultat. Dacă resultat este o matrice pătratică, aceasta va returna elementele de pe diagonala principală.
9. cat("\n\n- Diagonala principală având valorile 3, 2 și 4 matricei\n") - Afișează un mesaj pentru a indica următoarea secțiune care va trata crearea unei noi matrice pătratice.
10. print(diag(c(3, 2, 4))) - Creează o nouă matrice pătratică cu diagonala principală având valorile 3, 2 și 4. Celelalte elemente vor fi 0.
11. vecteur1 <- c(8, 3, 2) - Atribuie vectorului vecteur1 valorile 8, 3 și 2.
12. vecteur2 <- c(23, 6, 9) - Atribuie vectorului vecteur2 valorile 23, 6 și 9.
13. res = rbind(vecteur1, vecteur2) - Creează o matrice în care vecteur1 este prima linie și vecteur2 este a doua linie.
14. vecteur3 = c(2, 4) - Atribuie vectorului vecteur3 valorile 2 și 4.
15. cat("\n- concatenarea = ", cbind(res, vecteur3)) - Concatenează vectorul vecteur3 la dreapta matricei res și afișează rezultatul.
16. cat("\n- Valorile și vectorii proprii ai matricei\n") - Afișează un mesaj pentru a indica următoarea secțiune care va trata calculul valorilor și vectorilor proprii.
17. print(eigen(resultat)) - Calculează valorile și vectorii proprii ai matricei resultat folosind funcția eigen() și afișează rezultatul.



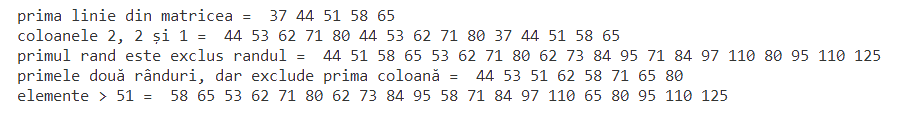


**Sarcina 5**

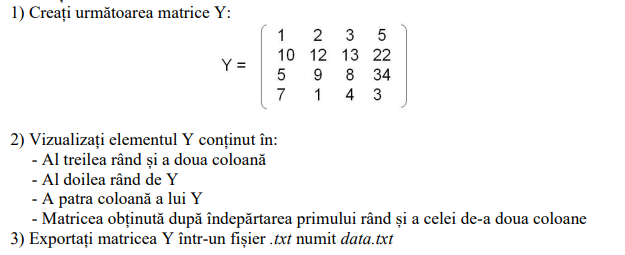
Rulați următorul cod și interpretați ceea ce este afișat:



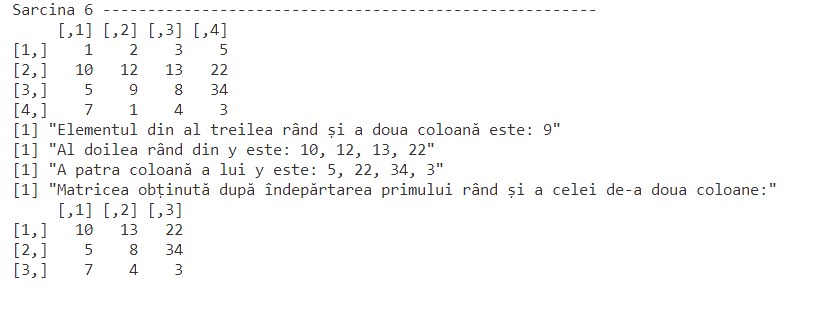
1. cat("\nprima linie din matricea = ", resultat[1,]) - Afișează prima linie din matricea resultat.
2. cat("\ncoloanele 2, 2 și 1 = ", resultat[, c(2, 2, 1)]) - Afișează coloanele 2, 2 și 1 din matricea resultat. Coloana 2 este afișată de două ori consecutiv.
3. cat("\nprimul rand este exclus randul = ", resultat[-1,]) - Afișează toate rândurile, cu excepția primului, din matricea resultat.
4. cat("\nprimele două rânduri, dar exclude prima coloană = ", resultat[1:2, -1]) - Afișează primele două rânduri, dar exclude prima coloană din matricea resultat.
5. cat("\nelemente > 51 = ", resultat[resultat > 51]) - Afișează toate elementele din matricea resultat care sunt mai mari decât 51.
6. matrix(vector, nrow = 2) - Creează o matrice cu 2 rânduri, completată cu valorile din vector. Valorile vor fi introduse pe coloane.
7. matrix(vector, nrow = 2, byrow = T) - Creează o matrice cu 2 rânduri, iar valorile din vector vor fi introduse pe rânduri.



**Sarcina 6**



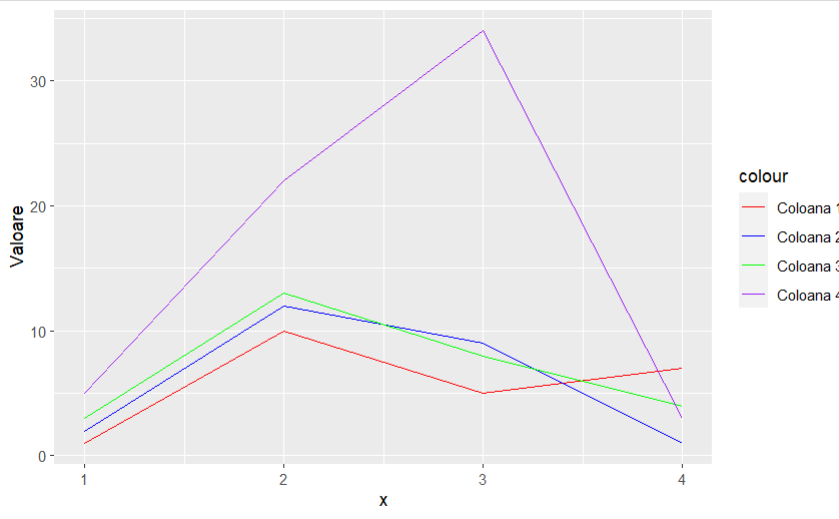
1. y <- matrix(c(1, 2, 3, 5, 10, 12, 13, 22, 5, 9, 8, 34, 7, 1, 4, 3), nrow = 4, byrow = TRUE) - Se creează o matrice y cu 4 rânduri și 4 coloane, folosind valorile specificate.
2. print(y) - Afișează matricea y.
3. element\_1 <- y[3, 2] - Se extrage elementul de pe rândul 3 și coloana 2 și se atribuie variabilei element\_1.
4. print(paste("Elementul din al treilea rând și a doua coloană este:", element\_1)) - Afișează un mesaj care conține valoarea element\_1.
5. rand\_2 <- y[2,] - Se extrage al doilea rând al matricei y și se atribuie variabilei rand\_2.
6. print(paste("Al doilea rând din y este:", paste(rand\_2, collapse = ", "))) - Afișează un mesaj care conține elementele din rand\_2, separate prin virgulă.
7. coloana\_4 <- y[, 4] - Se extrage a patra coloană a matricei y și se atribuie variabilei coloana\_4.
8. print(paste("A patra coloană a lui y este:", paste(coloana\_4, collapse = ", "))) - Afișează un mesaj care conține elementele din coloana\_4, separate prin virgulă.
9. matrice\_noua <- y[-1, -2] - Se creează o nouă matrice eliminând primul rând și a doua coloană a matricei y, și se atribuie variabilei matrice\_noua.
10. print("Matricea obținută după îndepărtarea primului rând și a celei de-a doua coloane:") - Afișează un mesaj.
11. print(matrice\_noua) - Afișează noua matrice.
12. fisier <- file("data\_sar\_6.txt", "w") - Se creează un fișier text numit "data\_sar\_6.txt" pentru scriere.
13. write.table(y, fisier, sep = "\t", row.names = FALSE, col.names = FALSE) - Se scrie matricea y în fișierul creat, folosind tab ca separator și fără a include numele de rânduri și coloane.
14. close(fisier) - Se închide fișierul.



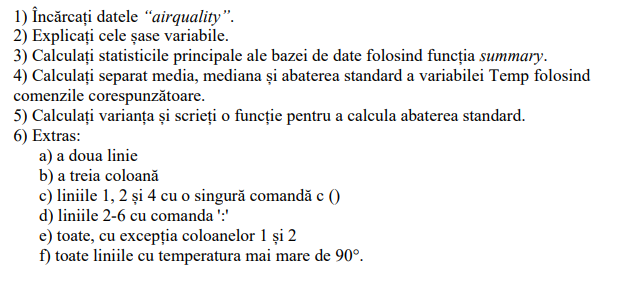
**Sarcina 7**

1. Verifică dacă pachetul ggplot2 este deja instalat. Dacă nu este instalat, îl instalează.
2. Încarcă pachetul ggplot2 pentru a putea folosi funcționalitățile sale.
3. Definește o funcție numită citeste\_si\_afiseaza\_grafic care primește ca argument un nume de fișier (nume\_fisier).
4. În interiorul funcției:
   1. date <- read.table(nume\_fisier, header = FALSE): Se citește conținutul fișierului specificat în variabila date, fără a considera prima linie ca fiind un antet (header).
   2. print(head(date)): Afișează primele câteva rânduri din setul de date pentru a le vizualiza.
   3. grafic <- ggplot(data = date, aes(x = V1, y = V2)) + ...: Se definește un obiect grafic folosind pachetul ggplot2. Acesta este un grafic de dispersie (scatter plot) cu datele din coloanele V1 și V2.
   4. labs(x = "X", y = "Y", title = "Grafic de dispersie"): Adaugă etichete pentru axele x și y, precum și un titlu pentru grafic.
   5. print(grafic): Afișează graficul creat.

În final, funcția citeste\_si\_afiseaza\_grafic este apelată cu argumentul "data.txt", ceea ce va duce la citirea datelor din fișierul data.txt și la afișarea graficului de dispersie utilizând pachetul ggplot2.



**Sarcina 8**

cat("\n\nSarcina 8 -------------------------------------------------------\n")

# 1) Încărcarea datelor "airquality".

data(airquality)

# 2) Explicarea celor șase variabile.

# Variabilele sunt:

#   - Ozone: Concentrația de ozon în aer (în unități de 10 micrograme pe picior cub).

#   - Solar.R: Radiația solară în Langleys (unitate de măsură a radiației solare).

#   - Wind: Viteza vântului (în mile pe oră).

#   - Temp: Temperatura (în grade Fahrenheit).

#   - Month: Lună (1-12).

#   - Day: Zi (1-31).

# 3) Calcularea statisticilor principale.

cat("\n3) Calcularea statisticilor principale\n")

print(summary(airquality))

# 4) Calcularea separată a mediei, medianei și deviației standard a variabilei Temp.

cat("\n\n4) Calcularea separată a mediei, medianei și deviației standard a variabilei Temp.")

mean\_temp <- mean(airquality$Temp)

median\_temp <- median(airquality$Temp)

sd\_temp <- sd(airquality$Temp)

cat("\nmedia = ",mean\_temp)

cat("\nmediana = ",median\_temp)

cat("\ndeviatia standart = ",sd\_temp)

# 5) Calcularea varianței și scrierea unei funcții pentru deviația standard.

cat("\n\n5) Calcularea varianței și scrierea unei funcții pentru deviația standard.")

calcul\_dev\_standard <- function(vector) {

  dev\_standard <- sqrt(var(vector))

  return(dev\_standard)

}

variance\_temp <- var(airquality$Temp)

dev\_standard\_temp <- calcul\_dev\_standard(airquality$Temp)

cat("\nvarianta = ",variance\_temp)

cat("\ndeviația standard = ", dev\_standard\_temp)

cat("\n\n 6) Extras")

# 6) Extras:

#    a) A doua linie

    a\_doua\_linie <- airquality[2,]

    cat("\na doua linie: \n")

    print(a\_doua\_linie)

#    b) A treia coloană

    a\_treia\_coloana <- airquality[, 3]

    cat("\na treia coloană: \n")

    print(a\_treia\_coloana)

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

    liniile\_124 <- airquality[c(1, 2, 4),]

    cat("\nliniile 1, 2 și 4 cu o singură comandă c (): \n")

    print(liniile\_124)

#    d) Liniile 2-6 cu comanda ':'

    liniile\_2\_spre\_6 <- airquality[2:6,]

    cat("\nliniile 2-6 cu comanda ':'\n")

    print(liniile\_2\_spre\_6)

#    e) Toate, cu excepția coloanelor 1 și 2

    toate\_exceptie\_12 <- airquality[, - c(1, 2)]

    cat("\ntoate, cu excepția coloanelor 1 și 2 ':' = \n")

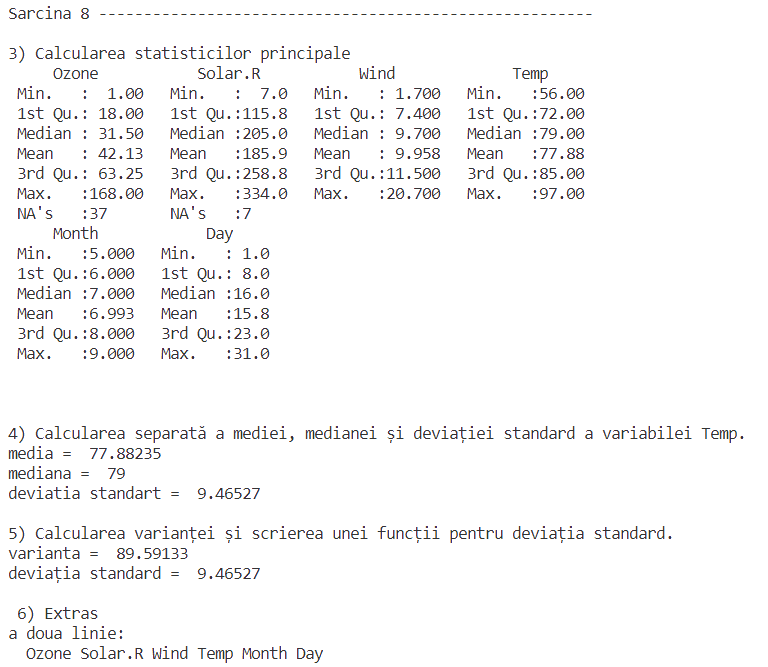
    print(toate\_exceptie\_12)

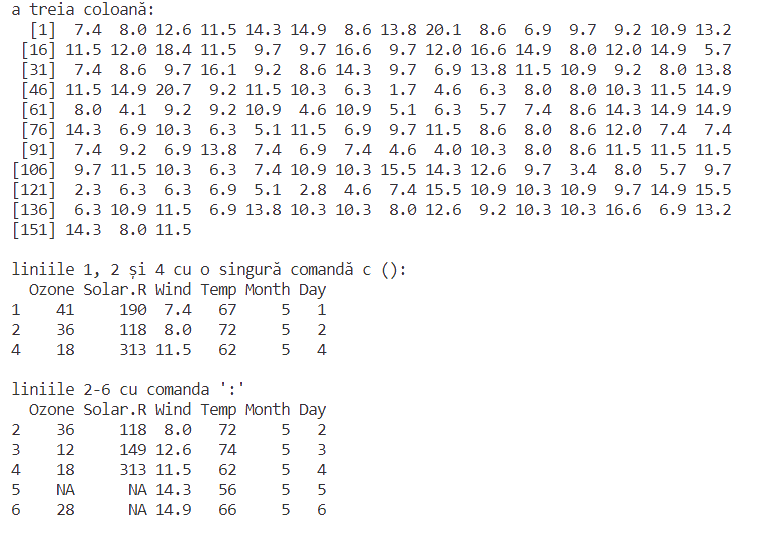
#    f) Toate liniile cu temperatura mai mare de 90°.

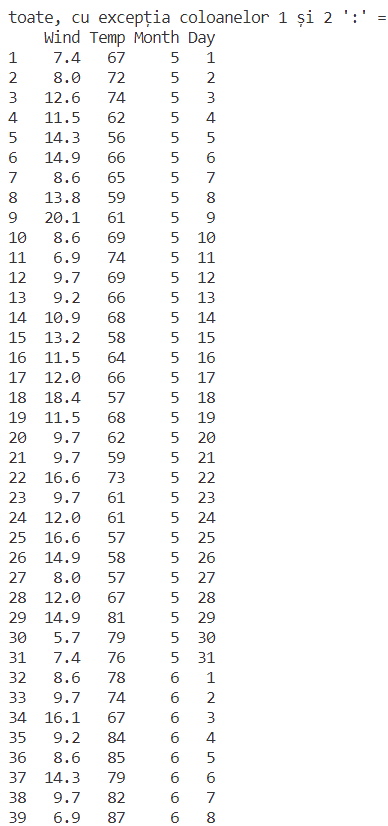
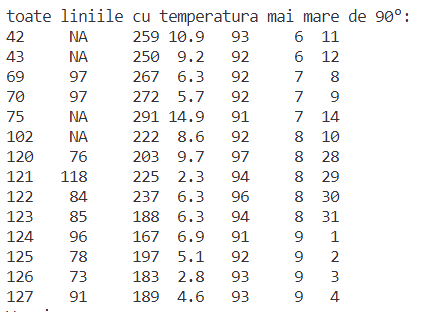
    temperatura\_mai\_mare\_de\_90 <- airquality[airquality$Temp > 90,]

    cat("\ntoate liniile cu temperatura mai mare de 90°: \n")

    print(temperatura\_mai\_mare\_de\_90)







**Concluzie**

În cadrul acestor instrucțiuni R, am acoperit mai multe aspecte ale manipulării și analizei datelor folosind limbajul R. Am început prin încărcarea setului de date "airquality" și am explicat variabilele incluse. Apoi, am calculat statisticile de bază pentru fiecare variabilă, inclusiv medie, mediană, minim, maxim și cuartile. De asemenea, am calculat media, mediana și deviația standard pentru variabila "Temp".

Am creat o funcție pentru calcularea deviației standard. Aceasta a fost apoi utilizată pentru a calcula și afișa varianța și deviația standard pentru variabila "Temp".

În cele din urmă, am explorat extrageri specifice din setul de date, cum ar fi a doua linie, a treia coloană, anumite rânduri specifice și altele.

Aceste instrucțiuni reprezintă doar o introducere în ceea ce poate face R pentru analiza datelor. Cu cunoștințe suplimentare și aplicări adecvate, R poate fi o unealtă extrem de puternică în analiza și vizualizarea datelor.