Universitatea Tehnica a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software și Automatică

# Raport

Lucrarea de laborator Nr.2

Programarea Declarativa

Tema: Vectori,matrice si data frame-uri

A efectuat: St. gr. TI-216

Vlașițchi Ștefan

A verificat: Lect. Univ.

Viorel Rusu

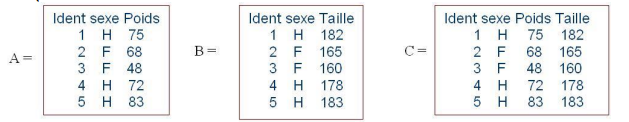
Chișinău 2023

**Scopul**:

**Exercițiul 1**

Creați două *data frames* **A** și **B**. Din aceste două *data frames* obțineți *data frames* **C**.

Ce funcție vom folosi în acest caz?



A <- data.frame( Indent=c(1:5),

B <- data.frame(Indent=c(1:5),

sexe =c("H","F","F","H","H"),

Taille = c(182,165,160,178,183)

)

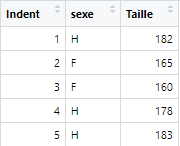
> View(B)

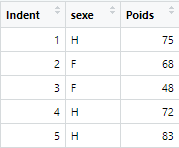
sexe =c("H","F","F","H","H"),

Poids = c(75,68,48,72,83)

)

View(A)



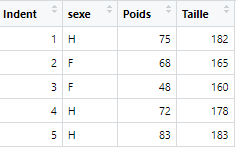


# Combinați data frame-urile "A" și "B" pe baza coloanelor "Indent" și "sexe"

C <- merge(A, B, by = c("Indent", "sexe"))

# Afișați data frame-ul "C"

View(C)



**Exercițiul 2**

Extrageți toate numerele între 2 și 3 din următorul vector:

> x=c(0.2, 0.6, 2.1, 3.7, 2.8, 2.7, 1.9, 2.3, 5.9)

x <- c(0.2, 0.6, 2.1, 3.7, 2.8, 2.7, 1.9, 2.3, 5.9)



# Extrageți numerele între 2 și 3

numere\_intre\_2\_si\_3 <- x[x >= 2 & x <= 3]

# Afișați rezultatul

cat("Numerele între 2 și 3 sunt:", numere\_intre\_2\_si\_3, "\n")



Numerele între 2 și 3 sunt: 2.1 2.8 2.7 2.3

**Exercițiul 3**

Creați următoarea matrice Y (respectând numele rândurilor și numele coloanelor):

column 1 column 2 column 3 column 4

row-1 1 6 5 0

row-2 0 6 6 1

row-3 3 0 2 2

row-4 4 4 3 4

# Definiți numele pentru rânduri și coloane

nume\_randuri <- c("row-1", "row-2", "row-3", "row-4")

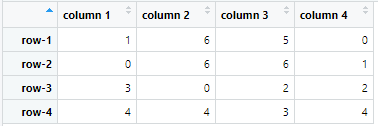
nume\_coloane <- c("column 1", "column 2", "column 3", "column 4")

# Definiți valorile matricei

valori\_matrice <- matrix(c(1, 0, 3, 4, 6, 6, 0, 4, 5, 6, 2, 3, 0, 1, 2, 4), nrow = 4, ncol = 4, byrow = FALSE)

# Creează matricea Y cu numele pentru rânduri și coloane

Y <- matrix(valori\_matrice, nrow = 4, ncol = 4, dimnames = list(nume\_randuri, nume\_coloane))



# Afișați matricea Y

Y

Calculați determinantul și inversați matricea folosind funcțiile necesare.

# Calculați determinantul matricei Y

determinant <- det(Y)

# Afișați determinantul

cat("Determinantul matricei Y:", determinant, "\n")



# Calculați inversa matricei Y

inversa <- solve(Y)

# Afișați inversa matricei (dacă este posibil)

if (!is.null(inversa)) {

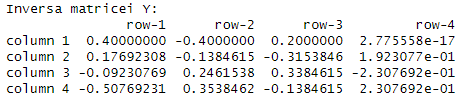
cat("Inversa matricei Y:\n")

print(inversa)

} else {

cat("Matricea Y nu este inversabilă.\n")

}



**Exercițiul 4**

a) Încărcați datele „Orange” (disponibile în R). Calculați statisticile de bază (media, abaterea standard, min, etc.) ale ultimelor două variabile ale acestui set de date.

b) Calculați quartilele ambelor variabile.

c) Folosind funcția “apply”, calculați toate decilele ambelor variabile folosind argumentul „probs” al funcției „quantile”.

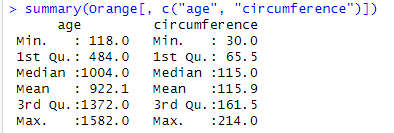
1. Calculați statisticile de bază pentru variabilele "age" și "circumference":

# Încărcați setul de date "Orange"

data("Orange")

# Calculați statisticile de bază pentru variabilele "age" și "circumference"

summary(Orange[, c("age", "circumference")])



1. Calculați quartilele pentru variabilele "age" și "circumference":

# Calculați quartilele pentru variabilele "age" și "circumference"

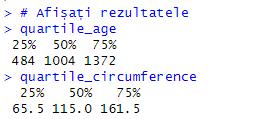
quartile\_age <- quantile(Orange$age, probs = c(0.25, 0.5, 0.75))

quartile\_circumference <- quantile(Orange$circumference, probs = c(0.25, 0.5, 0.75))

# Afișați rezultatele

quartile\_age

quartile\_circumference



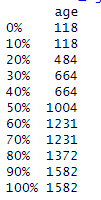
1. Folosind funcția "apply", calculați toate decilele pentru variabilele "age" și "circumference":

# Calculați decilele pentru variabila "age"

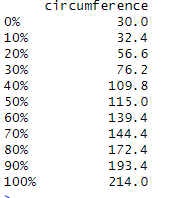
deciles\_age <- apply(Orange["age"], 2, function(x) quantile(x, probs = seq(0, 1, 0.1)))

# Calculați decilele pentru variabila "circumference"

deciles\_circumference <- apply(Orange["circumference"], 2, function(x) quantile(x, probs = seq(0, 1, 0.1)))

# Afișați rezultatele

decile\_circumference

decile\_age

**Exercițiul 5**

1. Creați vectorul k format din de trei ori șirul de numere (8; 2; 6).
2. Creați vectorul w compus din de șapte ori numărul 4, de 5 ori numărul 9 și de 3 ori numărul

2 (prin două metode diferite).

1. # Creați vectorul "k" cu șirul (8, 2, 6) repetat de trei ori

k <- rep(c(8, 2, 6), each = 3)

# Afișați vectorul "k"

k

1. # Creați vectorul "w" folosind funcții rep() și c()

w <- c(rep(4, 7), rep(9, 5), rep(2, 3))

# Afișați vectorul "w"

w

**Exercițiul 6**

1. Introduceți variabila „ size” care conține următoarele 9 valori: 178, 175, 160, 191, 176, 155, 163, 174, 182.
2. Introduceți variabila “size\_1” care conține următoarele 5 valori : 164, 172, 156, 195, 166.
3. Din variabilele „size” și „size\_1”, creați variabila „new.size” care conține: cele cinci valori ale „size1” repetate de două ori și ultimele șapte valori ale „size” .
4. Salvați în directorul dvs. de lucru variabila „new.size” într-un fișier în format .csv.
5. # Creați variabila "size"

size <- c(178, 175, 160, 191, 176, 155, 163, 174, 182)

1. # Creați variabila "size\_1"

size\_1 <- c(164, 172, 156, 195, 166)

1. # Creați variabila "new.size" conform specificațiilor date

new.size <- c(rep(size\_1, each = 2), tail(size, 7))

1. # Salvați variabila "new.size" într-un fișier .csv

write.csv(new.size, file = "new\_size.csv")

**Exercițiul 7**

* 1. Încărcați setul de date „iris”, apoi vizualizați primele 7 linii. Creați un subset de date care să conțină doar datele din modalitatea „versicolor” a variabilei „ Species ” (numiți acest nou set de date „new.iris”).
  2. Sortați în ordine descrescătoare datele „new.iris” în funcție de variabila Sepal.Length.

1. # Încărcați setul de date "iris"

data(iris)

# Vizualizați primele 7 linii ale setului de date "iris"

head(iris, 7)

# Creați un subset de date "new.iris" care conține doar datele pentru "versicolor"

new.iris <- subset(iris, Species==”versicolor”)

1. # Sortați "new.iris" în ordine descrescătoare după variabila "Sepal.Length"

new.iris <- new.iris[order(new.iris$Sepal.Length, decreasing = TRUE), ]

**Exercițiul 8**

Convertiți matricea A de tip caracter într-o matrice digitală.

> A [,1] [,2]

[1,] "8" "16"

[2,] "9" "2"

# Definiți matricea "A" de tip caracter

A <- matrix(c("8", "9", "16", "2"), nrow = 2, ncol = 2)

# Convertiți matricea "A" într-o matrice numerică

A\_numeric <- as.numeric(A)

# Afișați matricea numerică rezultată

A\_numeric

**Exercițiul 9**

Creați următorul cadru de date (data frame):

> person

height weight age c.eyes

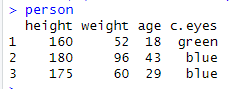
1 160 52 18 green

2 180 96 43 blue

3 175 60 29 blue

* + 1. Schimbați numele coloanei 3 cu „new.age”
    2. Schimbați numele liniei 2 cu „Mary”
    3. Ștergeți numele rândurilor
    4. Schimbați toate numele coloanelor cu a, b, c, d
    5. Extrageți elementul rândului 1 și al coloanei 3
    6. Extrageți variabila 2 (rezultat în data.frame , rezultat în vector)
    7. Extrageți elementul 1 și 3 al variabilei 3
    8. Extrageți valorile 160 superioare și 180 inferioare ale variabilei „height”.
    9. Extrageți valorile greutății persoanelor ale căror valori de înălțime sunt mai mari de 170
    10. Extrageți toate persoanele care au o greutate mai mare de 52 kg 11) Schimbați înălțimea primelor 2 persoane la 190 și 158

# Creați data frame "person"

person <- data.frame(

height = c(160, 180, 175),

weight = c(52, 96, 60),

age = c(18, 43, 29),

c.eyes = c("green", "blue", "blue")

)

# Afișați cadru de date "person"

Person

1) Schimbați numele coloanei 3 cu "new.age":

colnames(person)[3] <- "new.age"

2) Schimbați numele liniei 2 cu "Mary":

rownames(person)[2] <- "Mary"

3) Ștergeți numele rândurilor:

rownames(person) <- NULL

4) Schimbați toate numele coloanelor cu "a", "b", "c", "d":

colnames(person) <- c("a", "b", "c", "d")

5) Extrageți elementul rândului 1 și al coloanei 3:

element <- person[1, 3] 

6) Extrageți variabila 2 (rezultat în data.frame, rezultat în vector):

variabila\_df <- person[, 2]

variabila\_vector <- person[, 2]

7) Extrageți elementul 1 și 3 al variabilei 3:



element1 <- person[1, 3]

element3 <- person[3, 3]

8) Extrageți valorile 160 superioare și 180 inferioare ale variabilei "height":

valori\_superioare <- person$height[person$height > 160]

valori\_inferioare <- person$height[person$height < 180]



9) Extrageți valorile greutății persoanelor ale căror valori de înălțime sunt mai mari de 170:

greutate\_persoane\_inalte <- person$weight[person$height > 170]



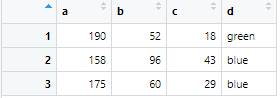
10) Extrageți toate persoanele care au o greutate mai mare de 52 kg:

persoane\_greutate\_mai\_mare <- person[person$weight > 52, ]



11) Schimbați înălțimea primelor 2 persoane la 190 și 158:

person[1:2, "height"] <- c(190, 158)



**Exercițiul 10**

Creați următoarea listă:

> my\_list [[1]]

[1] 5

[[2]]

[1] 160 180 175

[[3]]

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

[1,] 1 5 9

[2,] 2 6 10

[3,] 3 7 11

[4,] 4 8 12

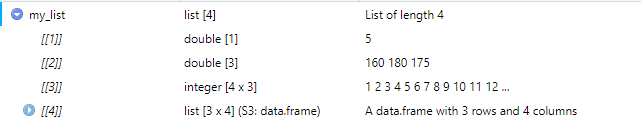
[[4]]

height weight age c.eyes

1 160 52 18 green

2 180 96 43 blue

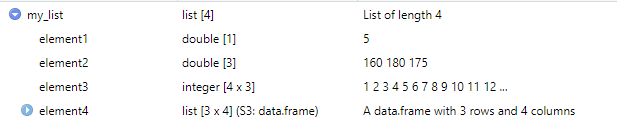
3 175 60 29 blue



1) Dați nume elementelor listei:

# Dați nume elementelor listei

names(my\_list) <- c("element1", "element2", "element3", "element4")



2) Extrageți al doilea element al listei (rezultat în vector, rezultat în listă):

Pentru a extrage al doilea element al listei într-un vector:

# Extrageți al doilea element al listei într-un vector

element2\_vector <- unlist(my\_list["element2"])



Pentru a extrage al doilea element al listei într-o listă nouă:

# Extrageți al doilea element al listei într-o listă nouă

element2\_list <- list(my\_list[["element2"]])



3) Extrageți primul și al treilea element din listă:

# Extrageți primul și al treilea element din listă

element1\_element3 <- my\_list[c("element1", "element3")]



4) Extrageți al treilea element din a doua coloană a celui de-al patrulea compartiment:

# Extrageți al treilea element din a doua coloană a celui de-al patrulea compartiment

element4\_col2\_row3 <- my\_list[["element4"]][3, 2]



**Concluzie**

În urma efectuării acestei lucrări de laborator, am dobândit o serie de abilități esențiale în utilizarea limbajului de programare R pentru analiza datelor. Am învățat să manipulez matrice, vectori, cadre de date și liste, să efectuez operații de selectare, calcul statistic și vizualizare grafică. Această experiență mi-a dezvoltat competențe în prelucrarea datelor și extragerea informațiilor relevante din seturi de date complexe. În plus, am învățat să folosesc funcții R pentru a automatiza procesele și să interpretez rezultatele obținute. Aceste abilități sunt esențiale în domeniul analizei datelor și mă pregătesc pentru abordarea problemelor complexe în acest domeniu în viitor.