**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. 2

# la Programarea Declarativă

**Tema: Vectori, matrici și data frame-uri***.*

A efectuat:

st. gr. TI-211 Popa Cătălin

A verificat: lect.dr. Mariana Rusu

UTM, Chișinău 2023

**Exercițiul 1**

Creați două *data frames* A și B. Din aceste *data frames* obțineți data frames C. Ce funcție vom folosi în acest cas?

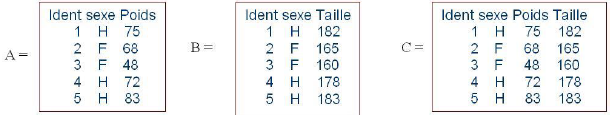


Figura 1 – Data frames.

# Creați data frame A  
A <- *data.frame*(Sexe = *c*("H", "F", "F", "H", "H"),  
 Poids = *c*(75, 68, 48, 72, 83))  
# Creați data frame B  
B <- *data.frame*(Sexe = *c*("H", "F", "F", "H", "H"),  
 Taille = *c*(182, 165, 160, 178, 183))  
# Combinarea frame-urilor A și B  
C <- *cbind*(A, B$Taille)  
# Redenumirea coloanelor pentru a se potrivi cerințelor  
*colnames*(C) <- *c*("Sexe", "Poids", "Taille")  
# Afișarea frame-ului C  
*print*(C)

**Rezultat :**

**1 H 75.0000 182.0002 F 68.0000 165.0003 F 48.0000 160.0004 H 72.0000 178.0005 H 83.0000 183.000**

Pentru a combina frame-urile A și B folosim funcția cbind, care va lipi coloanele frame-urilor într-unul singur.

**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)  
# Extrage numerele între 2 și 3  
rezultat <- x[x >= 2 & x <= 3]  
# Afișează rezultatul  
*print*(rezultat)

**Rezultat : 2.1 2.8 2.7 2.3**

Pentru a afișa numerele dintr-un interval, este nevoie să introducem vectorul și în [] indicăm intervalul.

**Exercițiul 3**

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

column 1 column2 column3 column4

row-1 1 6 5 0

row-2 0 6 6 1

row-3 3 0 2 2

row-4 4 4 3 4

# Definește datele  
matrice <- *matrix*(*c*(1, 0, 3, 4, 6, 6, 0, 4, 5, 6, 2, 3, 0, 1, 2, 4), nrow = 4, ncol = 4)  
# Creează matricea Y cu nume pentru rânduri și coloane  
*colnames*(matrice) <- *c*("column 1", "column 2", "column 3", "column 4")  
*rownames*(matrice) <- *c*("row-1", "row-2", "row-3", "row-4")  
# Afișează matricea Y  
*print*(matrice)

**Rezultat :**

**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**

**Exercițiul 4**

1. Încărcați datele **,,Orange”** (disponibile în R). Calculați statisticile de bază ale ultimelor două variabile ale acestui set de date.

# Încărcăm datele Orange  
*data*("Orange")  
# Accesăm ultimele două variabile din setul de date Orange  
variabile <- Orange[, *c*("circumference", "age")]  
# Calculăm statisticile de bază pentru aceste două variabile  
statistici <- *summary*(variabile)  
*print*(statistici)

**Rezultat :**

**circumference age**

**Min. : 30.0 Min. : 118.0**

**1st Qu.: 65.5 1st Qu.: 484.0**

**Median :115.0 Median :1004.0**

**Mean :115.9 Mean : 922.1**

**3rd Qu.:161.5 3rd Qu.:1372.0**

**Max. :214.0 Max. :1582.0**

1. Calculați quartilele ambelor variabile.

# Calculați quartilele pentru ambele variabile folosind funcția quantile  
quartile\_variabile <- *sapply*(variabile, quantile, probs = *c*(0.25, 0.5, 0.75))  
# Afișați rezultatele  
*print*(quartile\_variabile)

**Rezultat :**

**circumference age**

**25% 65.5 484**

**50% 115.0 1004**

**75% 161.5 1372**

* Primul Quartil (Q1) - Este valoarea de la care 25% dintre datele setului sunt mai mici. Corespunde percentilei 25 (25% dintre date sunt sub Q1).
* Al doilea Quartil (Q2) - Acesta este identic cu mediana (Q2 este, de asemenea, percentilele 50), împărțind setul de date în două jumătăți egale. 50% dintre date sunt mai mici decât Q2, iar 50% sunt mai mari.
* Al treilea Quartil (Q3) - Este valoarea de la care 75% dintre datele din set sunt mai mici. Corespunde percentilei 75 (75% dintre date sunt sub Q3).

1. Folosind funcția **,,apply”** , calculați toate decilele ambelor variabile folosind argumentul **,,probs”** al funcției **,,quantile”**.

# Calculăm toate decilele pentru ambele variabile  
probs <- *seq*(0.1, 1, by = 0.1) # Specificăm decilele dorite  
  
# Folosim funcția "apply" pentru a calcula decilele pentru fiecare variabilă  
decile <- *apply*(variabile, 2, function(x) *quantile*(x, probs = probs))  
*print*(decile)

**Rezultat :**

**circumference age**

**10% 32.4 118**

**20% 56.6 484**

**30% 76.2 664**

**40% 109.8 664**

**50% 115.0 1004**

**60% 139.4 1231**

**70% 144.4 1231**

**80% 172.4 1372**

**90% 193.4 1582**

**100% 214.0 1582**

**Exercițiul 5**

1. Creați vectorul k format din trei ori șirul de numere (8, 2, 6).

k <- *rep*(*c*(8, 2, 6), times = 3)  
*print*(k)

**Rezultat :** **8 2 6 8 2 6 8 2 6**

1. Creați vectorul w compus din șapte ori numărul 4, de 5 ori numărul 9 și de 3 ori numărul 2(prin două metode).

# Metoda 1  
w <- *rep*(*c*(4, 9, 2), times = *c*(7, 5, 3))  
*print*(w)  
  
# Metoda 2  
w <- *c*(*rep*(4, 7), *rep*(9, 5), *rep*(2, 3))  
*print*(w)

**Rezultat :**

**4 4 4 4 4 4 4 9 9 9 9 9 2 2 2**

**4 4 4 4 4 4 4 9 9 9 9 9 2 2 2**

**Exercițiul 6**

1. Introduceți variabila **,,size”** care conține următoarele 9 valori:

178,175,160,191,176,155,163,174,182.

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

1. Introduceți variabila **,,size\_1”** care conține următoarele 5 valori: 164, 172, 156, 195, 166.

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

1. Din variabilele ,,**size**” și ,,**size\_1**”, creați variabila **,,new.size”** care conține: cele cinci valori ale **,,size\_1”** repetate de două ori și ultimele șapte valori ale ,,size”.

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

1. Salvați în directorul dvs. de lucru variabila **,,new.size”** într-un fișier în format .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”**).

*data*(iris)  
 *head*(iris, 7)

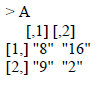
new.iris <- *subset*(iris, Species == "versicolor")

1. Sortați în ordine descrescătoare datele **,,new.iris”** în funcție de variabila Sepal.Lenght.

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

**Exercițiul 8**

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



Fgura 2 – Matrice de tip caracter.

A <- *matrix*(*c*("8", "9", "16", "2"), nrow = 2, ncol = 2)  
  
A\_num <- *matrix*(*as.numeric*(A), nrow = *nrow*(A), ncol = *ncol*(A))

**Exercițiul 9**

Creați următorul cadru de date (data frame):

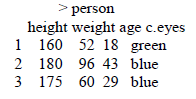


Figura 3 – Data frame.

# Crearea cadru de date "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")  
)

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)

# Data frame  
variable\_dataframe <- *data.frame*(person[, 2])  
  
# Vector  
variable\_vector <- person[, 2]

7) Extrageți elementul 1 și 3 al variabilei 3

elements <- person[*c*(1, 3), 3]

8) Extrageți valorile 160 superioare și 180 inferioare ale variabilei „height”.

superioare <- person$height[person$height >= 160]  
  
inferioare <- person$height[person$height <= 180]

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

var <- person$weight[person$height > 170]

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

mai\_mult\_52kg <- person[person$weight > 52, ]

11) Schimbați înălțimea primelor 2 persoane la 190 și 158

person$a[1:2] <- *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

my\_list <- *list*(  
 (5),  
 (*c*(160, 180, 175)),  
 (*matrix*(1:12, nrow = 4)),  
 (*data.frame*(  
 height = *c*(160, 180, 175),  
 weight = *c*(52, 96, 60),  
 age = *c*(18, 43, 29),  
 c.eyes = *c*("green", "blue", "blue")  
 ))  
)

1) Dați nume elementelor listei.

*names*(my\_list) <- *c*("lista1", "lista2", "lista3", "lista4")

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

# In vector  
vector <- *unlist*(my\_list$lista2)  
# In listă  
list <- my\_list$lista2

3) Extrageți primul și al treilea element din listă.

element1 <- my\_list$lista2  
element3 <- my\_list$lista3

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

compartiment4 <- my\_list$lista4  
element0 <- compartiment4[[3, 2]]

**Concluzie:**

În cadrul laboratorului, am explorat concepte legate de vectori, matrici, data frame-uri și lucrul cu aceste structuri de date în limbajul R. Am învățat să cream data frame-uri, să extragem și să manipulăm date din acestea, să lucrăm cu matrice, să calculăm determinantul și inversa matricii, să calculăm statistici de bază și cuantile pentru seturile de date, să construim vectori și să operăm cu aceștia, să manipulăm și să exportăm datele în fișiere, să lucrăm cu seturi de date existente, să sortăm și să convertim tipuri de date. De asemenea, am învățat să creăm și să manipulăm liste.