#Q: Daca vreau sa le sterg pe toate cum procedez?

rm(list = objects())

#Q: Explicati de ce nu merge urmatoarea atribuire multipla

d2+d4 ->d6<-d1

Nu merge pentru ca se incearca sa se atribuie valori diferite in acelasi timp unei variabile

#Q: Explicati rolul parantezelor si al operatorului : in urmatoarea situatie

1:n-1

1:(n-1)

2\*1:5

2\*1+1:5

2+1:5

(2\*1+1):5

30:3

Parantezele sunt folosite pentru ordinea operatiilor si : lista de la elementul din stanga si numarul din dreapta reprezinta ultimul element din lista si daca punem simbolul \* la elemental din dreapta si un numar in fata acel numar va fi ratia

#Q: Explicati comportamentul operatorului :

1.4:5

Lista de la elementul din stanga si numarul din dreapta reprezinta ultimul element din lista si daca punem simbolul \* la elemental din dreapta si un numar in fata acel numar va fi ratia. In cazul de fata incepem cu 1.4 si mergem cu ratia 1 pana la 4.4 care este ultimul numar inainte de 5.

#Q: Explicati urmatoarea secventa de cod

(b <- c(rep(rep(c(0,2),3),1:6),rep(rep(c(0:3,6),c(2,4:7)),24)))

rep(c(0,2), 3)

# [1] 0 2 0 2 0 2

rep(rep(c(0,2), 3), 1:6)

Secvența 1:6 pentru a repeta elementele vectorului produs mai sus

# [1] 0 2 2 0 0 0 2 2 2 2 0 0 0 0 0 2 2 2 2 2 2

rep(c(0:3,6), c(2,4:7))

# [1] 0 0 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3 6 6 6 6 6 6 6

rep(rep(c(0:3,6), c(2,4:7)), 24)

repeta 0 0 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3 6 6 6 6 6 6 6

de 24 de ori

#Q: Explicati urmatoarea secventa de cod

x <- sample(1:1000,100) #100 de valori random de la 1 la 1000

x[x%%8==0][1:3] #Primele 3 numere divizibile cu 8

#Q: Explicati urmatoarea secventa de cod

x\_1 <- x[-(1:3)] #x\_1 ia valorile din x mai putin primele 3 elemente

#Q: Explicati ce face urmatoarea comanda

c1 <- a1+b

c1 este egal cu elementele din a1 adunate cu elementele din b astfel : primul element din c1 este egal cu primul element din a1 adunat cu primul element din b si asa mai departe. C1 are atatea elemente cat a1, dar b are mai putin, in cazul de fata dup ace am adunat toate elemtele din b cu cele din a1, se reincepe de la capat cu adunarea elementelor din b pe rand pana le-am adunat cu toate elementele din a1

# Q: De ce vector\_mixt este in continuare un vector si nu o lista?

Valorile numerice sunt transformate in caractere precum literele si atunci vectorul este unul de caractere si nu o lista, pentru ca un vector are date omogene

y <- (x[1:50]<30)

#Q: Ce este in neregula cu expresia de mai sus?

X are doar 10 valori, deci Y nu poate lua 50 de valori din X, de aceea dupa a 10-a valorare Y ia valoarea NA

#Q: Cum putem elimina in mod automat valorile NA dintr-un vector?

#HINT: is.na()

y <- y[!is.na(y)]

#Q: Creati un vector logic t ce compara daca elementul de pe pozitia i(impara)

# este mai mic decat elementul de pe pozitia para imediat urmatoare

impar <- x[seq(1, length(x), by = 2)]

par <- x[seq(2, length(x), by = 2)]

t <- impar < par

t

#Q: Determinati pozitiile pentru care conditia de la prima intrebare este adevarata

#HINT: which()

which(t) -> u

u

#Q: Determinati cate numere cuprinse intre 411 si 7870 sunt divizibile cu 9

#dar nu cu 5

v <- 411:7870

v <- v[v%%9 == 0 & v%%5 !=0]

k <- length(v)

k

#To do

# Construiti doi vectori x si y cu 1000 de elemente fiecare, extrase in mod

#aleator din multimea cu numere intregi -24500:76000.

x <- sample(-24500:76000, 1000)

y <- sample(-24500:76000, 1000)

#a)Stabiliti care dintre cei doi vectori are mai multe elemente,

#luate in valoare absoluta, mai mari decat valoarea absoluta a elementului

#corespondent din celalalt vector

a <- sum(abs(x) > abs(y))

a

b <- sum(abs(x) < abs(y))

b

#b)Stabiliti care dintre cei doi vectori are minimul pe o pozitie mai mare

which.min(x) < which.min(y)

#c)Stabiliti care dintre cei doi vectori are cele mai multe valori care se repeta

c <- sum(duplicated(x))

d <- sum(duplicated(y))

c < d

#d)Stabiliti care dintre cei doi vectori are o secventa de cel putin 2 valori consecutive

e <- any(diff(x))

e

f <- any(diff(y))

f

#e)Stabiliti care dintre cei doi vectori are mai multe valori divizibile cu corespondentele

#lor din celalalt vector.

g <- sum(x %% y == 0)

h <- sum(y %% x == 0)

h > g