Programming in R

Task 3

Sofía Gianelli Nan, Gabriela Levenfeld Sabau and Miguel Díaz-Plaza Cabrera

2023-10-06

Contents

1. Exercise I: Create the following vectors

- **a) [10, 11, 12, . . . , 38]**

```
vect_a <- seq(10, 38, 1)
vect_a
```

```
##  [1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
## [26] 35 36 37 38
```

- **b) [30, 29, 28, . . . , 1]**

```
vect_b <- c(30:1)
vect_b
```

```
##  [1] 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6
## [26]  5  4  3  2  1
```

- **c) [1, 2, 3, 4, 3, 2, 1]**

```
vect_c <- c(seq(1, 4, 1), seq(3, 1, -1))
vect_c
```

```
## [1] 1 2 3 4 3 2 1
```

- **d) [2, 4, 6, . . . , 16, 18, 20]**

```
vect_d <- seq(2, 20, 2)
vect_d
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

- **e) [1, 2, 3, 1, 2, 3, . . . , 1, 2, 3]**

```
vect_e <- rep(seq(1, 3, 1), 10)
vect_e
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

- **f) [1, 2, 3, 1, 2, 3, . . . , 1]**

```
vect_f <- head(vect_e, -2)
vect_f
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1
```

- **g) [“label 1”, “label 2”,. . . , “label 30”]**

```r
vect_g <- paste('label', seq(1, 30))
vect_g
```

```
##  [1] "label 1"  "label 2"  "label 3"  "label 4"  "label 5"  "label 6"
##  [7] "label 7"  "label 8"  "label 9"  "label 10" "label 11" "label 12"
## [13] "label 13" "label 14" "label 15" "label 16" "label 17" "label 18"
## [19] "label 19" "label 20" "label 21" "label 22" "label 23" "label 24"
## [25] "label 25" "label 26" "label 27" "label 28" "label 29" "label 30"
```

- **h)** [**"label-1", "label-2",. . . , "label-30"**]

```r
vect_h <- paste('label', seq(1, 30), sep = '-')
vect_h
```

```
##  [1] "label-1"  "label-2"  "label-3"  "label-4"  "label-5"  "label-6"
##  [7] "label-7"  "label-8"  "label-9"  "label-10" "label-11" "label-12"
## [13] "label-13" "label-14" "label-15" "label-16" "label-17" "label-18"
## [19] "label-19" "label-20" "label-21" "label-22" "label-23" "label-24"
## [25] "label-25" "label-26" "label-27" "label-28" "label-29" "label-30"
```

- **i)** x^2e^x, $\mathbf{x = 0.1, 0.2, \ldots, 1.}$

```r
x <- seq(0.1, 1, 0.1)
vect_i <- (x^2)*(exp(x))
vect_i
```

```
##  [1] 0.01105171 0.04885611 0.12148729 0.23869195 0.41218032 0.65596277
##  [7] 0.98673883 1.42434619 1.99227852 2.71828183
```

2. Exercise II: Calculate the followings sums with and without using loops

- $\sum_{j=5}^{23}(j^2 + 3 * j^{0.5})$

```
# With loops
j <- 5; sol1 = 0;
while(j<=23){
  sol1 = sol1 + (j^2 + 3*(j^0.5))
  j = j+1
}
sol1
```

```
## [1] 4502.766
```

```
# Without loops
j_vect <- 5:23; result1 <- 0;
f_j_values <- function(x, y){
  (x^2 + 3*(x^0.5))
}
result1 <- sum(outer(j_vect, 0, FUN = f_j_values))
result1
```

```
## [1] 4502.766
```

- $\sum_{i=1}^{18} \frac{1.3^i}{i}$

```
# With loops
i <- 1; sol2 <-0;
while(i <= 18){
  sol2 = sol2 + (1.3^i)/i
  i = i+1
}
sol2
```

```
## [1] 37.23156
```

```
# Without loops
i_vect <- 1:18; result2 <- 0;
f_i_values <- function(x, y){
  ((1.3^x)/x)
}
result2 <- sum(outer(i_vect, 0, FUN = f_i_values))
result2
```

```
## [1] 37.23156
```

- $\sum_{i=1}^{10} \sum_{j=1}^{6} \frac{i^4}{3+j}$

```r
# With loops
sol3 <- 0;
for(i in 1:10){
  for(j in 1:6){
    sol3 = sol3 + (i^4)/(3+j)
  }
}
sol3
```

```
## [1] 25222.42
```

```r
# Without loops
i_vect <- 1:10; j_vect <- 1:6; result3 <- 0;
f_ij_values <- function(x, y){
  (x^4)/(3+y)
}
result3 <- sum(outer(i_vect, j_vect, FUN = f_ij_values))
result3
```

```
## [1] 25222.42
```

3. Exercise III

(a) What does the next code do?

```
set.seed(75)
M = matrix(sample(1:10, size=60, replace=TRUE), nrow=6, ncol=10)
```

The first line of code allows us to create the same randoms numbers every time we execute the code. Then we create a matrix named M which has 6 rows and 10 columns thanks to the last two parameters. Finally, the 60 numbers inside the matrix are selected randomly from 1 to 10 and by setting "replace=TRUE", it is being specify that the same number can be chosen several times.

(b) Find the number of entries in each row that are greater than 4.

```
greater_4 <- function(row){
  sum(row>4)
}
apply(M, MARGIN = 1, FUN = greater_4)
```

```
## [1] 8 7 8 7 4 3
```

Where first position of the output corresponds to the number of values greater than 4 in the first row, the second one to the second row and so on.

(c) Replace the third column of the previous vector M by the sum of the second and third column.

```
M[,3] = M[,2]+M[,3]
M
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    8    8   16    7    7    5    2    2    6     5
## [2,]    9    5    7    6    6    1    6    6    3     7
## [3,]    5    1   11    2    5    6    8    9   10     8
## [4,]    9    3    4    1    6   10   10    7    9    10
## [5,]    7    3    6    3    6    4    4    6   10     2
## [6,]    9    3    6    4    1    2    1   10    6     1
```

4. Exercise IV

Write a function which takes a single argument which is a matrix. The function must return a matrix which is the same as the function argument but every odd number is doubled.

```
oddDoubled <- function(X){
  ifelse(X %% 2 == 0, X, 2*X)
}
a <- matrix(1:8, nrow = 2, ncol = 4)
newA = oddDoubled(a)
```

```
a
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    5    7
## [2,]    2    4    6    8
```

```
newA
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    6   10   14
## [2,]    2    4    6    8
```

To illustrated that this code works, it is attached an example.