

R Short Course | MBBC Students 18/19

Introduction to R

GUIDED EXERCISES

Basic Operations

1. Create a vector `w` with the components 1,-1,2,-2.
`w<-c(1,-1,2,-2)`
2. Print that vector in the R console.
`w`
3. Obtain a description of `w` by using function `str()`.
`str(w)`
4. Create vector `w+1` and print it in the R console.
`x<-w+1; x`
5. Create vector (0, 1, 5, 10, 15, ..., 75) by using functions `c()` and `seq()`.
`s<-seq(5,75,by=5)`
`j<-c(0,1,s);j`
6. Construct an alphanumeric vector with the labels "Patient", "Entry number" and "Disease", with separator ":" and dimension 10.
`S<-paste("Patient","Entry number","Disease",1:10,sep=":")`
7. Consider $x = 4$ (`x<-4`), and check the following conditions:
 - a) $x > 23$
`a <- x>23`
 - b) $x \in [-1, 10]$
`b <- x>=-1 & x<=10`
 - c) $x = 3$
`c<-x==3`
 - d) $x \leq 45$
`d <- x<=45`

Indicate the logical conditions:

- (a) $a \wedge b$
`a&b`
- (b) $b \vee c$
`b|c`
- (c) $d \wedge b$
`d&b`

Vectors, Matrices and Data Frames

8. Create a numeric vector, a *string* (character) vector and a logical vector:

```
num.vec <- c(3,4,2,6,20)
char.vec <- c("koala","kangaroo","monkey")
logic.vec <- c(F,F,T,T)
```

9. Create the vector $\mathbf{y}=(0, 4, 2, 1, 0, 4, 0, 3, 0, 3, 3, 3, 4, 4, 2, 2, 0)$ and print it in the R console.

```
y<-c(0, 4, 2, 1, 0, 4, 0, 3, 0, 3, 3, 3, 4, 4, 2, 2, 0)
y
```

- (a) Search for elements of \mathbf{y} which are less or equal than 3.
`y[y<=3]`
- (b) Search for elements of \mathbf{y} equal to 0.
`y[y==0]`
- (c) Search for components different from 0.
`y[y!=0]`
- (d) How many elements of \mathbf{y} are lower than 3?
`length(y[y<3])`

10. Consider the following vector: (12,14,35,7,6,12,5,22,7,17,9,11).

- (a) Use this vector to build a matrix (by column) of order 3×4 and print it in the R console.

```
m<-matrix(c(12,14,35,7,6,12,5,22,7,17,9,11),ncol=4)
m
```
- (b) What is the element of matrix \mathbf{m} , which is in position "line 2, column 3"?
`m[2,3]`

11. Consider the following table:

student nr.	class	marks
2355	tp1	10.3
3456	tp1	9.3
2334	tp2	14.2
5456	tp3	15.0

- (a) Put the table into a data frame. `final.marks<- data.frame(stud
= c(2355, 3456, 2334, 5456),
class = c("tp1", "tp1", "tp2", "tp3"),
marks = c(10.3,9.3,14.2,15))`
`final.marks`
- (b) What is the element in line 2 and column 2?
`final.marks[2,2]`
- (c) Access the column corresponding the students' number.
`final.marks$stud`
- (d) What is the information obtained from the data frame `final.marks` when we execute the following commands?
`final.marks[final.marks$marks > 10,]`
`final.marks[final.marks$marks > 14, "stud"]`
`final.marks[final.marks$class == "tp1", c("stud", "marks")]`
- (e) Execute the function which allows you to access the columns of the data frame directly.
`attach(final.marks)`
- (f) Insert a new column corresponding to the final results: "approved", "oral", "approved", "approved" (use function `edit()`).
`final.marks<-edit(final.marks)`
- (g) Print the variables' names of the data frame.
`names(final.marks)`

Graphical Functions

12. Consider the data frame `births` of package `Epi`. This database concerns births of babies in a hospital in England.

- (a) Identify the variables in the data frame `births`.

```
library(Epi)
data(births)
str(births)
```
- (b) Analyse the relation between the weight of babies and the gestational week, of the data frame `births`.

```
attach(births)
plot(gestwks,bweight)
```
- (c) Plot the age of the mother *vs.* the weight of the babies.

```
plot(matage,bweight)
plot(matage,bweight,xlab="Mother age", ylab="Babies weight")
```
- (d) Change the points color into green circles.

```
plot(matage,bweight,pch=19,col="green")
```
- (e) Consider the graph `'plot(gestwks,bweight)'`. Use different color for baby girls and baby boys:

```
plot(gestwks,bweight)
points(gestwks[sex==1],bweight[sex==1],col="blue")
points(gestwks[sex==2],bweight[sex==2],col="red")
```
- (f) Include a legend and a title.

```
legend("topleft",pch=1,legend=c("Baby boys","Baby girls"),
col=c("blue","red"))
title("Birth weight vs. gestational week (500 births)")
```
- (g) Save the graphic into a `.pdf` file.
Graph window - Export - Save as pdf

Loops and Functions

13. (a) Create a loop, using the `for()` function, that, given a numeric vector prints a number per line along with its square and its cube.

```
x<-c(1,3,5) # for example
n<-length(x)
for(i in 1:n)
  cat(x[i], ' square =', x[i]^2, '; cube =', x[i]^3, "\n")
```

- (b) Use function `while()` in order to reach the same result.

```
i<-1 # initialize counter
while(i<(n+1))
{
  cat(x[i], ' square =', x[i]^2, '; cube =', x[i]^3, "\n")
  i<-i+1
}
```

14. Write a function that takes a single argument: a matrix. The function should return a matrix, identical to the original, but in which every odd number is duplicated. For example, the function applied to matrix

$$\begin{bmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{bmatrix}$$

is converted into

$$\begin{bmatrix} 2 & 2 & 6 \\ 10 & 2 & 6 \\ -2 & -2 & -6 \end{bmatrix}$$

```
Func <- function(mat)
{
  mat[mat%%2 == 1] <- 2 * mat[mat%%2 == 1]
  mat
}
mat<-matrix(c(1,5,-2,1,2,-1,3,6,-3),3,3)
Func(mat)
```

NOTE: The operator `%%` returns the remainder of division, e.g., `5%%2 = 1`.