

## **Exercise 1.**

### **Basics in R**

**Create the script “exercise1.R” (in R Studio: File -> ...) and save it to the “Rintro/day1” directory: you will save all the commands of exercise 1 in that script.**

**Remember you can comment the code using #.**

**From the console:**

1. Go to **Rintro/day1**.

First check where you currently are with **getwd()**; then go to Rintro/day1 with **setwd()**

2. Using R as a calculator, calculate the square root of 654.

**sqrt()**

3. Using R as a calculator, calculate the percentage of males and females currently present in the classroom.

4. Create a new object **myobject** with value 60. Show **myobject** in the console.

5. Reassign **myobject** with value 87.

6. Subtract **1** to **myobject**. Reassign.

7. Create a new object **mysqrt** that will store the square root of **myobject**.

8. Create a new object **mydiv** that will store the result of **myobject** divided by **mysqrt**.

## **Exercise 2.**

### **Numeric vector manipulation**

**Create the script “exercise2.R” and save it to the “Rintro/day1” directory: you will save all the commands of exercise 2 in that script.**

1. Go to **Rintro/day1**

First check where you currently are with **getwd()**; then go to Rintro/day1 with **setwd()**

2. Create a numeric vector **y** from 2 to 11 included. Show **y** in the console.

3. How many elements are in **y**? I.e what is the length of vector **y**?

**length()**

4. Show the 2<sup>nd</sup> element of **y**.

5. Show the 3<sup>rd</sup> and the 6<sup>th</sup> elements of **y**.

6. Remove the 4<sup>th</sup> element of **y**: reassign. Check the length of **y**.
7. Show all elements of **y** that are **inferior to 7**.  
`y[? < ?]`
8. Show all elements of **y** that are superior to 4 **and** inferior to 9, both included.  
`y[? & ?]`
9. Show all elements of **y** that are **either inferior to 4, or superior to 9**.  
`y[? | ?]`
10. Create the vector **x** of 1000 random numbers from the normal distribution:  
 First read the help page of **rnorm()** function.
11. What are the mean, median, minimum and maximum values of **x**?  
**mean(); median(); min(); max()**
12. Run the **summary()** function on **x**.  
 What additional information to you obtain?
13. Create vector **y2**:  
`y2 <- c(1, 11, 5, 62, 18, 2, 8)`
14. What is the sum of all elements in **y2**?  
**sum()**
15. Which elements of **y2** are present in **y**?  
*Note: remember %in%.*
16. Multiply each element of **y2** by 1.5: reassign.
17. Use the function **any()** to check if the number 3 is present.

### **Exercise 2b. OPTIONAL**

1. Create vector **myvector**:  
`myvector <- c(1, 2, 3, 1, 2, 3, 1, 2, 3)`  
 Create the same vector using the **rep()** function (?rep)
2. Replace the values at the 5<sup>th</sup>, 6<sup>th</sup> and 7<sup>th</sup> position in **myvector** with 8, 12 and 32.
3. Calculate the **fraction/percentage** of each element of **myvector** relative to the total sum of the vector.  
**sum()** can be useful.
4. Add vector `c(2, 4, 6, 7)` after the last element of **myvector**: reassign!

### **Exercise 3.**

#### **Character vector manipulation.**

**Create the script “exercise3.R” and save it to the “Rintro/day1” directory: you will save all the commands of exercise 3 in that script.**

1. Go to **Rintro/day1**

First check where you currently are with **getwd()**; then go to Rintro/day1 with **setwd()**

2. Create vector **w**:

```
w <- rep(x=c("miRNA", "mRNA"), times=c(3, 2))
```

3. View vector **w** in the console: what is function **rep()** doing?  
Play with the “times” argument.

4. What does **table(w)** outputs?

5. Type **w[grepl(pattern="mRNA", w)]** and **w[w == "mRNA"]**

Is there a difference between the two outputs?

6. Now type **w[grepl(pattern="RNA", w)]** and **w[w == "RNA"]**

Is there a difference between the two outputs?

7. Create vector **g**:

```
g <- c("hsa-let-7a", "hsa-mir-1", "CLC", "DKK1", "LPA")
```

How many elements do **w** and **g** contain?

```
length()
```

8. Use the function **identical()** to check whether the lengths of **w** and **g** are the same.

9. Name the elements of **g** using **w** (i.e. the names of each element of **g** will be the elements of **w**).

```
names()
```

### **Exercise 3b. Optional**

1. Use the **sub()** function to replace “miRNA” with “microRNA” in the names of **g**.

2. Count how many microRNAs and mRNAs there are in **g** based on the column names.

3. Create vector **tt**:

```
tt <- "Introduction to R course"
```

How many characters does **tt** contain? **nchar()**

4. Remove “Introduction to R ” from **tt**.

You can try with **substr()** or **gsub()**.