

PRINCIPIOS Y HERRAMIENTAS DE PROGRAMACIÓN
Departamento de Ciencias e Ingeniería de la Computación

TRABAJO PRÁCTICO Nº3
LENGUAJE R

Ejercicio 1.

Ejecute las siguientes expresiones y analice la razón del resultado.

```
> n = 10
> 1:n-1
```

```
> n = 10
> 1:(n-1)
```

Ejercicio 2.

a. Ejecute y analice el resultado de las siguientes expresiones para crear vectores:

- | | |
|----------------------|-----------------------------|
| i. 1:10 | viii. seq(10, 15, length=6) |
| ii. 3:5.5 | ix. seq(10, 15, length=4) |
| iii. 2.8:10.2 | x. seq(10, 15, len=4) |
| iv. seq(1, 30, by=2) | xi. rep(c(10,12), 3) |
| v. seq(10, -6, -2) | xii. rep(c(5,8), c(2,3)) |
| vi. seq(10, -6, 2) | xiii. rep(1:4, c(2,1,5,2)) |
| vii. rep('a', 5) | xiv. double(20) |

from: indica el valor inicial de la sucesión

to: indica el valor final de la sucesión

by: indica el espaciado entre los valores

length: indica la longitud del vector resultante

along: un objeto cuya longitud se usará para el objeto a construir.

Función **seq** con argumentos opcionales.

b. Escriba al menos dos expresiones para crear cada uno de los siguientes vectores:

v1 = 10 11 12 13 14 15 16 17 18

v2 = 1 4 7 10 13 16 19 22 25

c. Con la función **length** podemos averiguar el tamaño de un vector y también podemos definir el tamaño de un vector. Analice el resultado de ejecutar la siguiente secuencia de expresiones:

```
v1 = c(1,2,4,7,10); length(v1)
length(v1) = 20; v1
length(v1) = 2; v1
```

d. Existen muchas formas de acceder a los elementos de un vector. Ejecute las siguientes expresiones y analice los resultados.

```
set.seed(123)
x = sample(1:100, 20); x
x[10]
x[22]
x[c(2,8,4,10:12)]
x[10:15]
x[-5]
x[-(5:15)]
x%%2==0
x[x%%2==0]
sum(x%%2==0)
sum(x[x%%2==0])
```

```
nombres=c("juan","ana","maria","luis","pedro")
sexo=c("H","M","M","H","H")
nombres[sexo=="M"]
sum(sexo=="M")
```

Ejercicio 3. Usando **sample** generar un vector de 20 valores entre 0 y 10, y guardarlo en el objeto **X**. Luego:

- Crear un vector **soloPARES** con los valores pares de **X**.
- Obtener el primer valor múltiplo de 4 en **X** (distinto de 0).
- Obtener la posición del primer valor múltiplo de 4 en **X** (distinto de 0).
- Obtener la cantidad de elementos de **X** son pares.
- Obtener la cantidad de elementos de **X** son múltiplos de 3 y no de 2 (distintos de 0).
- Determinar si todos los elementos de **X** son pares.
- Determinar si algún elemento de **X** es múltiplo de 10.

Ejercicio 4.

- Escriba una función para devolver un vector de dos elementos con la componente más grande y la componente más chica de una matriz. ¿Sirve para arreglos?
- Escriba una expresión para devolver un vector que contenga la suma de las componentes de cada fila de la matriz.
- Escriba una expresión para devolver la menor de las sumas de las componentes de cada fila de la matriz.
- Escriba una expresión para indicar si una matriz es simétrica. ¿Imaginas cómo podría implementarse este ejercicio en C?

Ejercicio 5. Escribir una expresión en R para calcular la sumatoria de todos los elementos de una matriz. Comparar la solución con el ejercicio análogo en C del TP2.

Ejercicio 6. Analice la información contenida en el *dataset* *ToothGrowth*.

- Escriba una o varias expresiones para obtener el tamaño promedio de los dientes del grupo de cerdos tratados con una dosis de 0.5 mg., agrupados según la forma de administración.
- Analice los gráficos mostrados luego de ingresar:
> `hist(ToothGrowth)`
> `hist(ToothGrowth$len)`

Ejercicio 7. El dataset *PlantGrowth* contiene datos de experimentos sobre el crecimiento de ciertas plantas en situaciones de control y de dos tratamientos distintos (3 grupos).

- Escriba una expresión para calcular el promedio de peso de cada grupo (usar **tapply()**).
- Representar la información obtenida en el inciso **a** con un gráfico de torta..

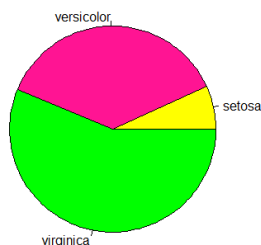
Ejercicio 8. “iris” es un conjunto de datos de R, que contiene distintas características de las plantas pertenecientes a la familia con ese nombre. A partir de los datos almacenados en el dataset “iris” escribir expresiones para:

- Generar un objeto llamado *longPetalos* con los *promedios* de longitud de pétalo (*Petal.Width*) agrupados por especie (*Species*) (usar **tapply()**). Debe quedar así:

```
> longPetalos
  setosa versicolor virginica
  0.246    1.326    2.026
```

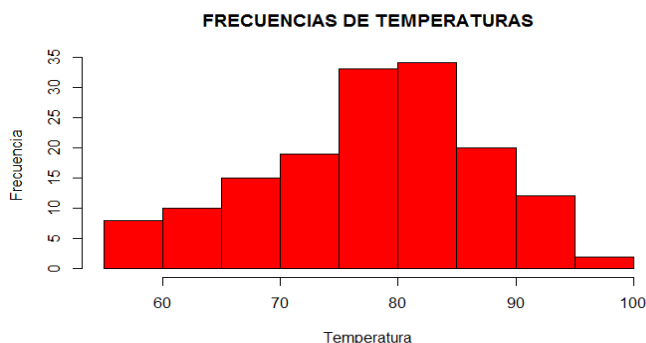
- b. Generar un gráfico de torta como el que se muestra a continuación representando los valores almacenados en el objeto *longPetalos*.

Longitud promedio de pétalos por especie



Ejercicio 9. Cargue el dataset *airquality* y analice la información contenida en el mismo. Luego escriba expresiones para:

- Calcular cuantas veces la temperatura fue mayor a 75 grados *Fahrenheit*.
- Generar un subconjunto con los datos correspondientes al mes de junio.
- Generar un subconjunto con los datos correspondientes a los primeros 15 días del mes de agosto.
- Obtener el promedio de temperaturas agrupado por mes (usar **`tapply()`**).
- Ilustrar la frecuencia de valores de temperaturas en un histograma como se muestra a continuación.



Opcional: Buscar en la web si existe en R una función para convertir de grados *Fahrenheit* a grados *Celsius*. 💡

En caso de encontrarla, genere el gráfico con la temperatura en grados Celsius.