

Introducción a R y ejercicios de la primera práctica:

Índice:

1. Información básica:.....	2
- Asignación de valores a una variable:.....	2
- Visualizar las variables:	2
- Eliminar objetos:	2
-Escribir comentarios	2
-Preguntar por un comando	2
2. Vectores:	2
- Escribir vectores:.....	2
- Operaciones con vectores:	3
o Suma y resta:.....	3
o Producto componente a componente:.....	3
o Producto escalar:	3
3. Ejercicios:	4
- Ejercicio 1: Uso del comando seq y creación de gráficos.	4
-Ejercicio 2: Suma de vectores. Introducción a los bucles (for y while).....	6
o Suma mediante un bucle “for”:	6
o Suma mediante el bucle condicional “while”:	7

1. Información básica:

- **Asignación de valores a una variable:** Se debe poner el nombre de la variable, seguido de un igual y el contenido que se le quiera dar, en ese orden. Si el contenido al que se asocia la variable es no numérico (como texto), se debe poner entre comillas ("").

```
Alfonso='Valladolid'
```

```
Num_uvas=4
```

*Si lo que queremos asociar a la variable es un vector u otro tipo de dato, se debe de hacer con su formato correspondiente.

- **Visualizar las variables:** Introducir el comando `ls()` para que el programa muestre todas las variables almacenadas en su memoria.

- **Eliminar objetos:**

- Usando el comando `rm(list=ls(all=TRUE))` se eliminan todos los objetos de la memoria del ordenador. Es recomendable usarlo al comienzo de cada programa para que el programa anterior no interfiera.
- Para borrar solo una variable, introducir en la consola `rm(...)`, poniendo entre paréntesis la variable (para eliminar varias, dentro del paréntesis se separan entre sí por comas).

-**Escribir comentarios** se pone un # delante y no afecta al programa:

```
#esto es un comentario
```

-**Preguntar por un comando:** Escribir `?` seguido del comando. De esta forma, el programa te da más información sobre el comando.

Por ejemplo: `?plot`

2. Vectores:

- **Escribir vectores:** Para escribir vectores se le pone un nombre al vector seguido de un =, y luego el vector, con el formato: `c(x,y,z)`; siendo x,y,z los elementos del vector (que pueden ser infinitos) separados por comas.

Por ejemplo:

```
v1=c(8,9/6, sin(pi/5), exp(2))
```

```
v2=c(17/7, cos(8+pi), sqrt(6),5^4)
```

*Para expresar valores de la función e^x , se usa la notación `exp(x)`: `exp(1)` es e^1

*Los `sin` y `cos` vienen en radianes por defecto. El número π se escribe como `pi`

*La notación `sqrt` es una raíz cuadrada

*Las potencias de números se escriben como `x^y`

- Operaciones con vectores:

*Los elementos usados en todos los tipos de operaciones deben haberse definido anteriormente o dará error

*No mezclar tipos distintos de variables en un mismo vector (números y palabras), porque no se puede operar con ellos

- **Suma y resta:** Se escriben ambos vectores con el símbolo de una suma o una resta (+ ó -), y son operados componente a componente. Ambos vectores tienen que ser numéricos. El resultado es un vector suma (o vector resta) que se puede almacenar en una variable:

Por ejemplo: `rosilla=v1+v2`

- **Producto componente a componente:** Se escriben ambos vectores usando como símbolo de multiplicación el *. Ambos tienen que ser numéricos. El producto es un vector multiplicación que se puede almacenar en una variable:

Por ejemplo: `prod=v1*v2`

- **Producto escalar:** Se escriben ambos vectores con `%*%`. Ambos tienen que ser numéricos. El producto es un valor escalar que se puede almacenar en una variable:

Por ejemplo: `pesc=v1%*%v2`

3. Ejercicios:

- Ejercicio 1: Uso del comando seq y creación de gráficos.

Consideramos la producción de manzanas durante los 10 primeros meses del año

1	2	3	4	5	6	7	8	9	10
100	90.50	45.23	68.14	30.60	25.55	12.01	16.48	32.00	87.10

1: Comando seq: Construir un vector meses que contenga los números de 1 a 10 asignados a los meses. Emplear para ello la instrucción: `seq(valor inicial, valor final, incremento)`

2. Construir un vector llamado manzanas que contenga la producción mensual de manzanas, a partir de la tabla dada.

3. - Creación de gráficos: Representar gráficamente la producción mensual de manzanas. Usando línea continua y color azul: `plot(meses, manzanas, type = 'b', col = 'blue')`.

4. Etiquetar los ejes como: 'Meses del año'(abscisas), 'Toneladas de manzanas' (ordenadas). Se usará `xlab ' ', ylab = ' '` para tal fin.

5. Utilizar `pch= número` (entre 0 y 25) para cambiar símbolos.

6. Poner título al gráfico. Se usará la instrucción `main = ' '`.

.....

0. Empezamos limpiando todas las variables existentes en el entorno de trabajo

```
rm(list=ls(all=TRUE))
```

Este comando es utilizado para eliminar todas las variables cargadas. "rm" elimina los objetos y las variables, "list" especifica que estamos eliminando elementos de una lista y "ls(all=TRUE)" lista todas las variables disponibles en el entorno, llevando a la eliminación de todas ellas.

1. Creamos el vector "meses" que tiene que contener valores desde 1 hasta 10 con incremento 1

```
meses=seq(1,10,1) otra opción: meses=seq(1,10, length=10)
```

2. Creamos el vector "manzanas" que tiene que contener los datos numéricos que representan las toneladas de manzanas en diferentes meses. Cada valor corresponde a un mes específico.

```
manzanas=c(100,90.50,45.23,68.14,30.60,25.55,12.01,16.48,32.00,87.10)
```

3, 4, 5, 6. Creamos el gráfico de dispersión que nos muestra los datos en "meses" en el eje x y los datos de "manzanas" en el eje y.

```
plot(meses,manzanas,type="b",col="purple",xlab="Meses del  
año",ylab="Toneladas de manzanas",pch=11,main="PRODUCCIÓN DE MANZANAS")
```

Añadiendo `type="b"`, creamos un plot con un tipo de línea con puntos

`col="purple"` especificamos el color morado

`xlab="Meses del año"`, `ylab="Toneladas de manzanas"` agregamos etiquetas a los ejes `x` e `y`.

`pch=11`. Definimos el símbolo de los puntos

`main="PRODUCCIÓN DE MANZANAS"` Agregamos un título

-Ejercicio 2: Suma de vectores. Introducción a los bucles (for y while).

1. Empezamos limpiando todas las variables existentes en el entorno de trabajo

```
rm(list=ls(all=TRUE))
```

Este comando es utilizado para eliminar todas las variables cargadas. "rm" elimina los objetos y las variables, "list" especifica que estamos eliminando elementos de una lista y "ls(all=TRUE)" lista todas las variables disponibles en el entorno, llevando a la eliminación de todas ellas.

2. Creamos el vector "u" que contiene los valores {8, 9/6, el seno de pi/5 y el exponente de 2 (e^2)}

```
u=c(8,9/6,sin(pi/5),exp(2))
```

3. Creamos el vector "v" que contiene los valores {17/7, el coseno de 8+pi, la raíz cuadrada de 6 y 5 elevado a la cuarta}

```
v=c(17/7,cos(8+pi),sqrt(6),5^4)
```

o Suma mediante un bucle "for":

4. Creamos la variable "w" que tiene un valor = 0. En esta almacenaremos la suma de "u" y "v"

```
w=0
```

5. Empezamos el bucle "for" para los elementos "u" y "v", uno por uno utilizando el índice i. El bucle varía desde 1 hasta la longitud de los vectores, que en este caso es 4, porque ambos "u" y "v" tienen 4 elementos. En la primera iteración del bucle, se suma el primer elemento de "u" al primer elemento de "v" y se almacena en "w[1]". En la segunda iteración se hace lo mismo para el segundo par de elementos y así sucesivamente hasta 4 (en este caso).

```
for (i in 1:length(u)){  
w[i]=u[i]+v[i]  
}
```

6. Expresamos el vector resultante.

```
w
```

○ *Suma mediante el bucle condicional "while":*

1. Creamos la variable "z" y le asignamos un valor inicial de 0,

```
z=0
```

2. Creamos la variable "i" y le asignamos un valor inicial de 1

```
i=1
```

3. Almacenamos en "n" la longitud del vector "u" utilizando la función "length()

```
n=length(u)
```

4. Empezamos el bucle "while" con la condición ($i \leq n$). Esto asegura que el bucle se ejecute mientras el valor de "i" sea menor o igual a la longitud de "u" pasando por todos los elementos de "u" (y "v"). En el bucle se suma el elemento "u" al elemento correspondiente "v".

El resultado es almacenado en $z[i]$

Se incrementa el valor de "i" en 1 utilizando $i = i + 1$ para asegurar que en la próxima iteración se procese el siguiente par de elementos de los vectores "u" y "v".

El bucle continúa ejecutando hasta que "i" sea mayor de que la longitud de "u". Cuando la condición $i < n$ se vuelve falsa el bucle se detiene.

```
while (i<=n){  
z[i]=u[i]+v[i]  
i=i+1  
}
```

5. Expresamos el vector resultante.

```
z
```