

ESQUEMA 1

CONTENIDOS BÁSICOS PARA USAR R.

Operaciones aritméticas: (básicas)

- Suma $\rightarrow +$
- Resta $\rightarrow -$
- Producto $\rightarrow *$
- División $\rightarrow /$
- Exponente $\rightarrow \wedge$
- Exponencial $\rightarrow \exp()$
(e^{\wedge})
- Raíz cuadrada $\rightarrow \text{sqrt}()$
- Valor absoluto $\rightarrow \text{abs}()$
- Logaritmo $\rightarrow \text{log}()$
neperiano

Ejemplos:

- $3 * 4 = 12$
- $2^{\wedge} 2 = 4$
- $\exp(1) = e$
- $\text{sqrt}(4) = 2$
- $\text{log}(1) = 0$
- $\text{log}_2(4) = 2$

Operaciones aritméticas: (avanzadas)

- Operadores:
 - Menor $\rightarrow <$ (o igual: $<=$)
 - Mayor $\rightarrow >$ (o igual: $>=$)
 - Distinto $\rightarrow !=$
 - Igualdad $\rightarrow ==$
- Orden de operaciones:
 - 1- Potencias.
 - 2- Producto y división.
 - 3- sumas y restas.
- otros funciones de interés:
 - $\sin(x) \rightarrow$ seno
 - $\cos(x) \rightarrow$ coseno
 - $\tan(x) \rightarrow$ tangente
 - factorial() \rightarrow factorial
 - arcos: $\arcsin(x)$
 - arcos: $\arccos(x)$
 - arcos: $\text{atan}(x)$

Asignaciones:

- Escoges una variable.
- $n \leftarrow n^{\circ}$
variable \leftarrow valor asignado.
- Para ver el valor de n:
 - $> n$
 - $[1] n^{\circ}$
 - o
 - $> \text{print}(n)$
 - $[1] n^{\circ}$

IMPORTANTE:
El valor de n es el último que se le haya asignado, los anteriores se pierden.

- Ej:
- $> \text{sample}(2:30, 10, \text{replace} = \text{TRUE})$
 - 10 valores entre 2 y 30 con repetición.

Comandos importantes:

1. $\text{runif}(N)$:
 - obtener valores aleatorios de $[0, 1]$ (básico)
 - Ej: $> \text{runif}(6)$
 - 6 valores entre 0 y 1.
 - En otro intervalo: $> n^{\circ} * \text{runif}(N)$ (avanzado)
 - Ej: $> 50 * \text{runif}(5)$
 - 5 valores entre 0 y 50.

2. $\text{sample}(a:b, M)$:
 - obtener valores M en intervalo $[a, b]$.

- con repetición: $\text{replace} = \text{TRUE}$.
- Sin repetición: $\text{replace} = \text{FALSE}$.
(más específico que runif)

3. $\text{ls}()$:

- Listar todo lo que se haya realizado con variables en el directorio de trabajo.
- si se desea información avanzada: $> \text{ls.str}()$ ó $> \text{ls}(pat = "otra")$:
- Lista las variables junto con su contenido o por otra en específico.

4. $\text{rm}()$:

- Borrar elementos de la memoria.
- si se desea eliminar uno o varios elementos en concreto, se deben introducir dichos elementos en el interior del paréntesis.

ESQUEMA 2

MATRICES Y VECTORES:

Vectores:

- Almacenamiento de valores.
- Los valores no tienen que ser números obligatoriamente.
- `> variable <- c()`
valores con comas de separación.

Ej: `x <- c(0, 2, "hoy")`

- Extracción de valor concreto:

`> variable [i]`

posición numérica del valor.
[i] valor.

- suma de componentes:

`> x <- variable[i] + variable[j]`
posiciones que queremos sumar.

IMPORTANTE:

1- Los valores numéricos no se pueden sumar con valores no numéricos.

2- Al igual que la suma, se pueden realizar diversas operaciones (resta, producto, etc...)

- Operaciones entre vectores:

- suma:
`> vector 1 + vector 2.`
- resta:
`> vector 1 - vector 2.`
- producto:
`> vector 1 * vector 2.`
- Producto escalar:
`> vector 1 %*% vector 2.`

(Diferenciar producto y p. escalar)

Matrices:

- Almacenamiento de gran cantidad de datos a la vez.

`A <- matrix(c(), nrow=, ncol=)`

matriz. vector que engloba todos los valores a incluir.
nº de filas. nº de columnas.

- Extracción de valor concreto:

`> A [i, j]`
fila columna
[1] valor

- Otra manera de formar matriz:
- A partir de diversos vectores.

`> v1 <- c()`

`> v2 <- c()`

`> v3 <- c()`

- si deseamos que los vectores sean columnas:

`> A <- cbind(v1, v2, v3)`

- si deseamos que sean filas:

`> A <- rbind(v1, v2, v3)`

(Los vectores deben tener todas las mismas dimensiones)

- Operaciones entre matrices:

- suma: (las matrices deben tener las mismas dimensiones)

`> A + B.`

- resta: (mismas dimensiones)

`> A - B.`

- producto escalar:

`> M = A %*% B`

(dimensiones apropiadas)

$M(m,n) = A(m,p) \cdot B(p,n)$

- producto: (elemento a elemento)

`> A * B.`

- Otras operaciones:

`t(A)` → transpuesta. solve (A, b)
`det(A)` → determinante. solución
`solve(A)` → inversa. $Ax = b$

ESQUEMA 3

BUCLES Y ESTRUCTURAS :

CONDICIONALES SECUENCIALES

Bucles for :

- Simples :

• Por cada vuelta se realiza un comando u operaci3n.

> for (i in vinit : vfin) {

comando u operaci3n

IMPORTANTE :
dejar ese espacio,
y cerrarlos.

SIGNIFICADO :

- i : valor que vara.

- vinit : valor de inicio.

- vfin : valor final.

• El comando se realiza con el n3mero de valores entre vinit y vfin.
(esos valores)

- Anidados :

• 2 o m3s bucles unos dentro de otros.

> for (i in vinit1 : vfin1) {

for (j in vinit2 : vfin2) {

for (k in vinit3 : vfin3) {

comando

se dejan espacios hacia dcha.

se demuestran todos

IMPORTANTE :

NO poner comandos entre bucles, solo al final.

ese comando incluir3 i, j y k

• uso clave : operaciones con matrices.

Bucles while :

• se puede resumir como una mezcla de los bucles for e if.

> while (condici3n 1) {

comando

cerrarlo siempre.

NO tiene que ser una 3nica operaci3n, de hecho lo normal es que sean varias.

Bucles if :

SON ESTRUCTURAS SECUENCIALES CONDICIONALES

• No dan vueltas, solo se propone una condici3n a cumplir.

- Simples :

• Una 3nica condici3n.

> if (condici3n 1) {

comando 1

} else {

comando 2

} siempre cerrado.

si se cumple la condici3n comando 1 se realiza.

sino se realiza el comando 2.

- Compuestos :

• Ramificaci3n de condici3nes.

> if (condici3n 1) {

comando 1

} else if (condici3n 2) {

comando 2

} else if (condici3n 3) {

comando 3

} else {

comando 4

}

Es 3nico que no tiene condici3n es el de "else"

conectores l3gicos :

• Para usar en condici3nes if.

• Destacan AND (y) y OR (o)

- AND : &

• se deben cumplir ambas condici3nes.

- OR : |

• s3lo debe cumplirse una condici3n.

(varias condici3nes en una 3nica expresi3n)

- SIGNIFICADO :

• Mientras la condici3n del bucle se cumpla, las operaciones del comando se realizarn.

• Una vez una condici3n no se cumpla, el bucle se acaba.

FUNCIONES Y REPRESENTACIÓN GRÁFICA

Función:

se expresa como:

> nombre = función (variable 1, variable 2, etc...)
 expresión de función

variables de las que depende la función creada.

Ejemplo: $f(x) = \text{sen}(x) + 1$

Nombre

Variable

expresión.

> $f = \text{función}(x) + 1$
 $\text{sen}(x) + 1$

Valores para representar:

Para crear una gráfica, hay que dar valores.

Opciones:

- runif
- rnorm
- vector creado con $\text{seq}(n^{\circ}, n^{\circ}, n^{\circ})$

valor inicial valor final
 Intervalos entre valor inicial y final.

4. TAMAÑO DEL TEXTO:
 (opción menos importante)
 $\text{cex} = \text{"número"}$

6. LÍMITES:

Para cercar la función:
 importante para cuando representamos más de 1 función.

$\text{ylim} = c(\text{valor}, \text{valor})$
 $\text{xlim} = c(\text{valor}, \text{valor})$
 son vectores.

9. SUPERPOSICIÓN:

para representar más de 1 función en un gráfico.
 $\text{par}(\text{new} = \text{"TRUE"})$
 se usa entre plots
 significa que va en la misma gráfica

Representación gráfica:

- Incluye gran cantidad de comandos.
- comando básico es:

> $\text{plot}(x, f(x))$

Aquí también puede ser el nombre del vector creado con seq.

2. COLOR:

debe escribirse en inglés.
 $\text{col} = \text{"color"}$

5. ESTILO DEL TEXTO:

$\text{font} = \text{"n}^{\circ} \text{ del 1 al 4"}$

7. SIMBOLOGÍA:

$\text{pch} = \text{n}^{\circ} \text{ del 0 al 25}$

□	0	△	+	x	◇
○	1	2	3	4	5
▽	⊠	*	◇	⊕	☆
6	7	8	9	10	11
⊞	⊗	⊠	■	●	▲
12	13	14	15	16	17
◆	●	●	●	■	◆
18	19	20	21	22	23
		▲	▼		
		24	25		

10. LEGENDA:

Instrucción que incluye la información representada.

$\text{legend}(\text{"posición"}, \text{rownames(A)} / \text{colnames(A)}, \text{fill} = c())$
 lugar de la leyenda (en inglés) que aparecen (en vectores) funciones (en comandos) adicionales (en vectores)

Dentro del comando "plot", se pueden añadir gran cantidad de comandos para completar la función y la representación.

1. TÍTULO:

es una asignación:
 $\text{main} = \text{"título"}$

va dentro del plot entre comas con otros comandos

3. NOMBRE DE EJES:

Importancia de saber lo que representa cada eje.
 $\text{xlab} = \text{"título eje x"}$
 $\text{ylab} = \text{"título eje y"}$

1. Normal

2. cursiva

3. Negrita

4. Negrita cursiva.

8. TIPOLOGÍA:

diferentes formas de la gráfica.
 $\text{type} = \text{"letra"}$

p → de puntos.

l → de líneas.

b → de puntos y líneas.

o → líneas y enema puntos.

s → escalera.

h → Histograma.