

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")`

- Elección de valor concreto:
 - `> variable []`
posición numérica del valor.
`[]` valor.

- suma de componentes:
 - `> x ← variable [] + variable []`
posiciones que queremos sumar.

IMPORTANTE:

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

Elección de valor concreto:

`> A [,]`

↓ ↓
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:

`> A * B` (elemento a elemento)

Otras operaciones:

`t(A)` → transpuesta.

`det(A)` → determinante.

`solve(A)` → inversa.

solu. (A, b)

solución
↓
 $Ax = b$