

Resumen

Cosas a tener en cuenta

- Libro. Biblioteca.
- Función generic == específica -> distintos tipos de datos.
- Explicar magnitudes y variables codificadas como:

```
Sexo = c("H", "M")
```

Cosas básicas

- Asignación: <-, =
- nombres largos de variables con puntos

vectores

- 1:10
- todos mismo tipo
- empiezan por el índice 1 (no 0)
- las operaciones entre vectores se hacen componente a componente
- selección de componentes de un vector:

```
x[i]  
x[seq(1,10,by=2)] # subconjunto de índices explícito  
x[y==2] # subconjunto de índices lógicos
```

- cuidado con vector(, 50). mejor vector(mode="numeric", 50)

matrices

- matrix(1:10, 2, 5)
- todos mismo tipo
- empiezan por el índice 1 (no 0)
- las operaciones entre matrices se hacen componente a componente
- selección de componentes de una matriz:

```
matriz[i,j] # componente (i,j)  
matriz[1,] # 1ª fila  
matriz[,1] # 1ª columna
```

- producto matricial: %*%

listas

- `list(A=1:10, B=2)`
- pueden ser distinto tipo
- nombrar siempre a las variables
- seleccion de componentes

```
lista$A # elemento con nombre A
lista[[1]] # primer elemento
```
- usar para devolver datos desde una función

dataframe

- `data.frame(Peso=c(50, 60), Altura=c(1.60, 1.70), Sexo=c("H","M"))`
- pueden ser de distinto tipo
- nombrar siempre a las variables
- selección de componentes

```
hoja$Peso
hoja[[1]]
```
- `I()` para identificadores como nombres

funciones

- `function(*args)`
- usar siempre `return` al final

Lista funciones

- `help()`
- `ls()`
- `mean()`
- `c()`
- `vector(mode, length)`
- `length()`
- `search()`
- `matriz(data, nrow, ncol, byrow)`
- `t()`
- `crossprod()`
- `outer()`
- `solve()`
- `eigen()`
- `determinant()`
- `list()`
- `data.frame()`
- `I()`
- `is()`
- `as()`
- `apropos()`

- `summary()`
- `typeof()`
- `sin()`
- `sum()`
- `prod()`
- `function()`
- `var()`

`var()` es un estimador de varianza poblacional (dividido por $n-1$). Para calcular la varianza muestral (dividido por n) hay tres formas:

```
# 1 forma
mean((x-mean(x))^2)
# 2 forma
((n-1)/n)*var(x)
# 3 forma
library(moments)
moment(x, 2, TRUE)
```

Para calcular los momentos centrales de orden n hay dos formas:

```
# 1 forma
mean((x-mean(x))^n)
# 2 forma
library(moments)
moment(x, n, TRUE)
```

Fragmentos reutilizables

Si tenemos una serie de datos, es común crear la matriz utilizando solo el número de columnas (igual al número de características):

```
Peso <- c(74, 47, 83)
Altura <- c(1.73, 1.55, 1.79)
matrix(c(Peso, Altura), ncol=2)
```

Si queremos escribir rápidamente una matriz dada:

```
3 2
1 -1
matrix(c(3, 2, 1, -1), ncol=2, byrow=T)
```

Ejemplo de función

```
name <- function(x = NA) {
  x=x[!is.na(x)]
  # ...
  return(x)
}
```