

Tema 3 - Estructuras de datos

Juan Gabriel Gomila & María Santos

Vectores

Tipos de datos en R

Un **vector** es una secuencia ordenada de datos. R dispone de muchos tipos de datos, por ejemplo:

- ▶ `logical`: lógicos (`TRUE` o `FALSE`)
- ▶ `integer`: números enteros, \mathbb{Z}
- ▶ `numeric`: números reales, \mathbb{R}
- ▶ `complex`: números complejos, \mathbb{C}
- ▶ `character`: palabras

En los vectores de R, todos sus objetos han de ser del mismo tipo: todos números, todos palabras, etc. Cuando queramos usar vectores formados por objetos de diferentes tipos, tendremos que usar **listas generalizadas**, `lists` que veremos al final del tema.

Básico

- ▶ `c()`: para definir un vector
- ▶ `scan()`: para definir un vector
- ▶ `fix(x)`: para modificar visualmente el vector x
- ▶ `rep(a,n)`: para definir un vector constante que contiene el

Factores

Factor

Factor: es como un vector, pero con una estructura interna más rica que permite usarlo para clasificar observaciones

- ▶ `levels`: atributo del factor. Cada elemento del factor es igual a un nivel. Los niveles clasifican las entradas del factor. Se ordenan por orden alfabético.
- ▶ Para definir un factor, primero hemos de definir un vector y transformarlo por medio de una de las funciones `factor()` o `as.factor()`.

La función `factor()`

- ▶ `factor(vector, levels=...)`: define un factor a partir del vector y dispone de algunos parámetros que permiten modificar el factor que se crea:
 - ▶ `levels`: permite especificar los niveles e incluso añadir niveles que no aparecen en el vector
 - ▶ `labels`: permite cambiar los nombres de los niveles
- ▶ `levels(factor)`: para obtener los niveles del factor

Lists

Similar a los diccionarios en Python

List

List. Lista formada por diferentes objetos, no necesariamente del mismo tipo, cada cual con un nombre interno

- ▶ `list(...)`: función que crea una list
 - ▶ Para obtener una componente concreta usamos la instrucción `list$componente`
 - ▶ También podemos indicar el objeto por su posición usando dobles corchetes: `list[[i]]`. Lo que obtendremos es una list formada por esa única componente, no el objeto que forma la componente

Obtener información de una list

- ▶ `str(list)`: para conocer la estructura interna de una list
- ▶ `names(list)`: para saber los nombres de la list

Obtener información de una list

(1 2 3 4 5 6 7 8 9 0)

Matrices

Cómo definir las

- ▶ `matrix(vector, nrow=n, byrow=valor_lógico)`: para definir una matriz de n filas formada por las entradas del vector
 - ▶ `nrow`: número de filas
 - ▶ `byrow`: si se iguala a `TRUE`, la matriz se construye por filas; si se iguala a `FALSE` (valor por defecto), se construye por columnas. `-ncol`: número de columnas (puede usarse en lugar de `nrow`)
 - ▶ R muestra las matrices indicando como $[i,]$ la fila i -ésima y $[,j]$ la columna j -ésima
 - ▶ Todas las entradas de una matriz han de ser del mismo tipo de datos

Cómo definir las

Ejercicio

- ▶ ¿Cómo definirías una matriz constante? Es decir, ¿cómo definirías una matriz A tal que $\forall i = 1, \dots, n; j = 1, \dots, m, a_{i,j} = k$ siendo $k \in \mathbb{R}$? Como R no admite incógnitas, prueba