

Diccionarios y otras colecciones



A pesar de que los vectores, matrices y listas son los tipos de colecciones más habituales en R, el lenguaje permite trabajar con otras colecciones alternativas, que repasaremos brevemente en este documento.

1. Diccionarios

Los diccionarios, también llamados tablas *hash* o mapas, son colecciones de datos no indexadas numéricamente. Es decir, no accedemos a los elementos almacenados indicando en qué posición están, sino asociándoles una clave. Así, se establecen parejas de *clave-valor* donde guardamos una clave con un valor asociado y, para recuperar o acceder a ese valor, debemos hacerlo mediante su clave.

En realidad, para **crear un diccionario** lo que haremos será crear un vector o array asociativo donde, además de indicar los valores que almacenaremos, indicaremos qué clave o nombre va asociado a cada uno. Por ejemplo:

```
diccionario <- c(a=1, b=2, c=3)
# Usamos doble corchete para indicar la clave
print(diccionario[["a"]]) # 1
```

Podemos **añadir nuevos elementos al diccionario** podemos hacerlo por su clave, de este modo:

```
diccionario[["d"]] <- 4
```

Finalmente, podemos **eliminar elementos del diccionario** filtrando todos los elementos que no tengan esa clave/nombre:

```
diccionario <- diccionario[!names(diccionario) %in% c("b")]
```

El **tamaño** o número de elementos del diccionario, como de otras colecciones, se puede obtener con `length`:

```
cat(length(diccionario))
```

2. Conjuntos

Los conjuntos son otro tipo de colección no indexada (no existen índices numéricos) donde no se permiten elementos duplicados. Así, podemos concebir un conjunto como un conjunto no ordenado de datos, donde no existen dos iguales.

El lenguaje R no proporciona una forma nativa de trabajar con conjuntos. Podemos simularlos a partir de vectores, empleando funciones como `unique`, que eliminan duplicados:

```
vector <- c(1, 2, 3, 1, 4)
conjunto <- unique(vector) # 1, 2, 3, 4
```

Alternativamente podemos emplear alguna librería o paquete externo como el paquete `sets`, instalando el paquete y utilizando sus funciones. No lo veremos en este documento, aunque [aquí](#) explicamos cómo trabajar con paquetes externos.

Ejercicio 1:

Crea un programa llamado `kilometrajes.r` donde crearemos un diccionario cuyas claves serán matrículas de coches (textos) y cuyos valores serán los kilómetros recorridos por cada coche. Al principio el usuario dará de alta 5 coches en el diccionario, y luego le pediremos que escriba una matrícula y le responderemos qué kilometraje tiene asociado.

AYUDA: [vídeo con solución del ejercicio](#)