

Clase 1.1

Estructuras de datos

Marcos Rosetti y Luis Pacheco-Cobos

Estadística y Manejo de Datos con R (EMDR) — Virtual

Listas

Listas

- Una estructura de datos que puede contener ¡todos los demás tipos!

```
una.lista <-list(df = data.frame(pago = c(33, 50, 40),
                                dia = c("Lunes", "Martes", "Miercoles")),
                m = matrix(letters[1:25], ncol = 5),
                vc = seq(from = 1, to = 5, by = 0.4))

una.lista
```

```
## $df
##   pago      dia
## 1   33   Lunes
## 2   50   Martes
## 3   40 Miercoles
##
## $m
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "a"  "f"  "k"  "p"  "u"
## [2,] "b"  "g"  "l"  "q"  "v"
## [3,] "c"  "h"  "m"  "r"  "w"
## [4,] "d"  "i"  "n"  "s"  "x"
## [5,] "e"  "j"  "o"  "t"  "y"
##
## $vc
## [1] 1.0 1.4 1.8 2.2 2.6 3.0 3.4 3.8 4.2 4.6 5.0
```

Listas

- Podemos referenciar un elemento de la lista por su nombre (si lo tiene).

```
una.lista$df
```

```
##      pago      dia  
## 1     33     Lunes  
## 2     50     Martes  
## 3     40 Miercoles
```

Listas

- Podemos referenciar un elemento por su ubicación numérica.

```
una.lista[1]
```

```
## $df
##   pago      dia
## 1   33   Lunes
## 2   50   Martes
## 3   40 Miercoles
```

Listas

- Podemos referenciar, obtener el elemento en su forma original, con doble corchete `[[]]`

```
una.lista[[3]]
```

```
## [1] 1.0 1.4 1.8 2.2 2.6 3.0 3.4 3.8 4.2 4.6 5.0
```

```
una.lista[[3]] + 1
```

```
## [1] 2.0 2.4 2.8 3.2 3.6 4.0 4.4 4.8 5.2 5.6 6.0
```

- ¿Qué pasa si referenciamos con corchete simple?

```
una.lista[3]
```

Listas

- Es posible agregar componentes a una lista.

```
una.lista[[length(una.lista) + 1]] <- c("un", " ", "vector", " ", "mas")
str(una.lista)
```

```
## List of 4
## $ df:'data.frame': 3 obs. of 2 variables:
## ..$ pago: num [1:3] 33 50 40
## ..$ dia : chr [1:3] "Lunes" "Martes" "Miercoles"
## $ m : chr [1:5, 1:5] "a" "b" "c" "d" ...
## $ vc: num [1:11] 1 1.4 1.8 2.2 2.6 3 3.4 3.8 4.2 4.6 ...
## $ : chr [1:5] "un" " " "vector" " " ...
```

Listas

- Es posible remover componentes a una lista.

```
una.lista[length(una.lista)] <- NULL  
str(una.lista)
```

```
## List of 3  
## $ df:'data.frame': 3 obs. of 2 variables:  
## ..$ pago: num [1:3] 33 50 40  
## ..$ dia : chr [1:3] "Lunes" "Martes" "Miercoles"  
## $ m : chr [1:5, 1:5] "a" "b" "c" "d" ...  
## $ vc: num [1:11] 1 1.4 1.8 2.2 2.6 3 3.4 3.8 4.2 4.6 ...
```


Listas

- Operaciones rápidas con `lapply()`

```
otra.lista <- list(c(1:4), c(10:25))  
lapply(otra.lista, mean) # media aritmética (promedio)
```

```
## [[1]]  
## [1] 2.5  
##  
## [[2]]  
## [1] 17.5
```

Listas

- Operaciones rápidas con `lapply()`

```
otra.lista <- list(c(1:4), c(10:25))  
lapply(otra.lista, sum) # suma
```

```
## [[1]]  
## [1] 10  
##  
## [[2]]  
## [1] 280
```

- Observa que `lapply()` aplica una FUN (función) a cada elemento de la lista y regresa una *lista*.

Listas

- Operaciones rápidas con `sapply()`

```
otra.lista <- list(c(1:4), c(10:25), c(35:40))  
sapply(otra.lista, mean) # media aritmética (promedio)
```

```
## [1] 2.5 17.5 37.5
```

Listas

- Operaciones rápidas con `sapply()`

```
otra.lista <- list(c(1:4), c(10:25), c(35:40))  
sapply(otra.lista, sum)
```

```
## [1] 10 280 225
```

- Observa que `sapply()` aplica una FUN (función) a cada elemento de la lista y regresa un *vector*.

Listas

- Operaciones rápidas con `do.call()`

```
otra.lista <- list(c(1:4), c(10:25))  
do.call("c", otra.lista)
```

```
## [1] 1 2 3 4 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

```
do.call("rbind", otra.lista)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]  
## [1,]    1    2    3    4    1    2    3    4    1    2    3    4    1    2  
## [2,]   10   11   12   13   14   15   16   17   18   19   20   21   22   23  
##      [,15] [,16]  
## [1,]     3     4  
## [2,]    24    25
```

- Observa que `do.call()` pasa a una FUN (función) a cada elemento de la lista como argumento.

Licencia CC BY



Estadística y Manejo de Datos con R (EMDR) por Marcos F. Rosetti S. y Luis Pacheco-Cobos se distribuye bajo una [Licencia Creative Commons Atribución 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).