

# Basicos R parte 1

## Basicos

```
TRUE == TRUE
```

```
## [1] TRUE
```

## Vectores

¿Qué es un vector?

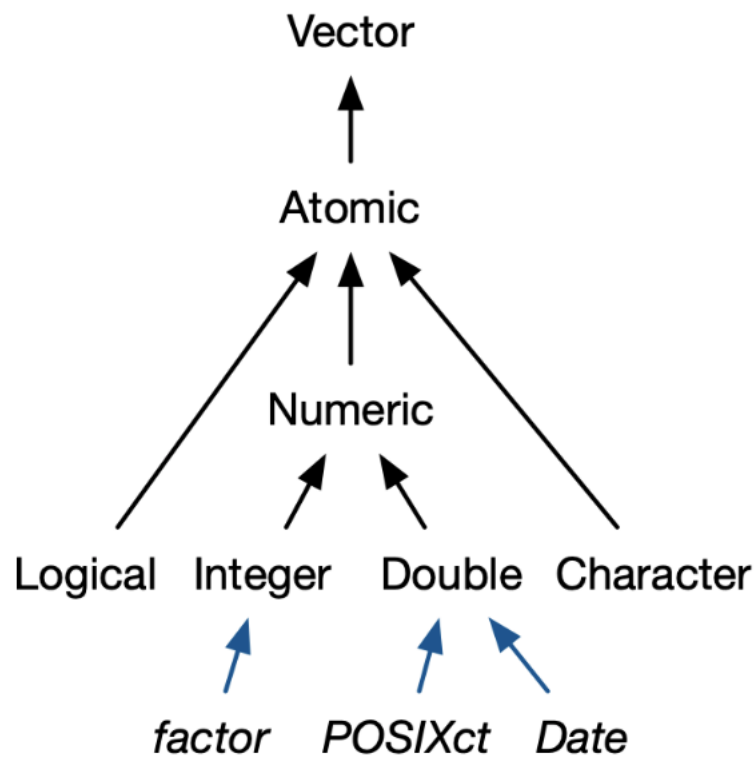


Figure 1: Vector S3

La estructura más simple de datos en R.

```
x <- c("a", "b", "c")
```

Se puede construir un vector de tipo numérico, lógico o carácter. La letra **c** significa “concatenar”, y de hecho es un acrónimo para dicha palabra.

Existen cuatro tipos comunes: *logical*, *integer*, *double*, *character*

### Propiedades de un vector:

#### 1. Tipo

```
x <- 1:5
typeof(x)
```

```
## [1] "integer"
```

#### 2. Longitud

```
x <- 1:5
length(x)
```

```
## [1] 5
```

#### 3. Atributos

```
x <- 1:5
class(x)
```

```
## [1] "integer"
```

### Trabajar con vectores

En R se indexa la información desde 1, para acceder a la información de una posición específica de un vector se usan los `[]`.

```
x <- c("Montserrat", "Nidia", "Armando", "Oscar", "Diego")
x[3]
```

```
## [1] "Armando"
```

```
x[c(1,2,3)]
```

```
## [1] "Montserrat" "Nidia"      "Armando"
```

```
x[1:3]
```

```
## [1] "Montserrat" "Nidia"      "Armando"
```

```
x[-3]
```

```
## [1] "Montserrat" "Nidia"      "Oscar"      "Diego"
```

Para agregar un valor al vector se debe tomar en cuenta el tipo de vector. *Nota: si indexo con signo negativo, se elimina esa posición*

```
x[6] <- "Fernanda"
x
```

```
## [1] "Montserrat" "Nidia"      "Armando"    "Oscar"      "Diego"
## [6] "Fernanda"
```

```
x <- x[-6]
x
```

```
## [1] "Montserrat" "Nidia"      "Armando"    "Oscar"      "Diego"
```

## Listas

Pueden tener mas de un objeto de diferente clase, es decir, puedo almanecer dentro de la misma lista objetos de clase caracter, factor y numericos, inclusive pueden añadirse data.frame!!!

```
my_list <- list(c("a", 1) , c("hola", "como", "estas"),
               data.frame(Nombres = c("Nidia", "Mont"), Sexo= c("F", "M")))
my_list

## [[1]]
## [1] "a" "1"
##
## [[2]]
## [1] "hola" "como" "estas"
##
## [[3]]
##      Nombres Sexo
## 1   Nidia     F
## 2    Mont     M
```

Para ver la estructura de cualquier tipo de dato se ocupa el comando **str()**

```
str(my_list)

## List of 3
## $ : chr [1:2] "a" "1"
## $ : chr [1:3] "hola" "como" "estas"
## $ : 'data.frame': 2 obs. of 2 variables:
## ..$ Nombres: Factor w/ 2 levels "Mont","Nidia": 2 1
## ..$ Sexo : Factor w/ 2 levels "F","M": 1 2
```

¿Cómo se accede a los datos de una lista?

¿Cuál es la diferencia entre los siguientes dos codigos?

```
my_list[1]

## [[1]]
## [1] "a" "1"

my_list[[1]]

## [1] "a" "1"
```

Con `[]` accedemos al valor (genera un objeto de menor peso de memoria), y con `[[` accedemos a la posición de la lista. Para obtener el numero 1 de la posición 1 de mi lista se hace lo siguiente:

```
my_list[[1]][2]

## [1] "1"
```

Para acceder a la palabra “como” de nuestra lista se hace lo siguiente:

```
my_list[[2]][2]

## [1] "como"
```

Tambien se puede ocupar el simbolo `$` para acceder a los datos de una lista cuando cada valor tiene un nombre

```
my_list_2 <- list(Datos_sin_sentido = c("a", "b", 4, 8), Palabras = c("hola", "como", "estas"),
                 Un_data_frame = data.frame(Nombres = c("Nidia", "Mont"), Sexo= c("F", "M")))
```

```
my_list_2
```

```
## $Datos_sin_sentido
## [1] "a" "b" "4" "8"
##
## $Palabras
## [1] "hola" "como" "estas"
##
## $Un_data_frame
##   Nombres Sexo
## 1   Nidia    F
## 2    Mont    M
```

```
my_list_2$Un_data_frame
```

```
##   Nombres Sexo
## 1   Nidia    F
## 2    Mont    M
```