

Objetos complejos: Vectores, Listas y Dataframes

Jorge Meneses y Paulo Peña

Estructuras de Datos

- Hasta el momento hemos visto tipos de datos básicos (cadenas de textos, enteros, numéricos, valores lógicos)
- Las estructuras de datos son formas particulares de organizar la información para usarla de forma más efectiva
- Veremos 3 estructuras:
 1. Vectores
 2. Listas
 3. Tablas de datos (*Data Frames*)

El comando `typeof` nos permite verificar el tipo de dato del vector

```
typeof(numeros_primos)
```

```
## [1] "double"
```

```
typeof(paises)
```

```
## [1] "character"
```

Listas

Listas son una colección **ordenada** de datos de **diferentes tipo**.

```
# Mis datos
```

```
yo <- c("Jorge Meneses", 36, "Sociología")
```

```
yo
```

```
## [1] "Jorge Meneses" "36"           "Sociología"
```

Se pueden crear listas usando el comando `list()`

```
gaseosas <- list(  
  "Coca-cola",  
  "Pepsi",  
  "Fanta"  
)
```

gaseosas

```
## [[1]]  
## [1] "Coca-cola"  
##  
## [[2]]  
## [1] "Pepsi"  
##  
## [[3]]  
## [1] "Fanta"
```

Largo (length) datos

Los vectores y listas tienen un largo que es igual a la cantidad de objetos que tienen. Para ver el largo de un objeto usamos el comando ``length()```

```
length(numeros_primos)
```

```
## [1] 4
```

```
length(paises)
```

```
## [1] 3
```

Accediendo datos

Tanto en listas como vectores podemos acceder a cada uno de los datos de forma individual. Para eso necesitamos indicar el **índice**.

```
# Accediendo un vector  
numeros_primos[4]
```

```
## [1] 7
```

```
# Accediendo una lista  
gaseosas[2]
```

```
## [[1]]  
## [1] "Pepsi"
```


Modificando datos

Podemos cambiar un dato particular de un vector o lista volviendolo a asignar

```
gaseosas <- list(  
  "Coca-cola",  
  "Pepsi",  
  "Fanta"  
)  
  
# Reasignando segundo elemento  
gaseosas[2] <- "Inka Kola"  
  
# Mostrando contenido final  
gaseosas  
  
## [[1]]  
## [1] "Coca-cola"  
##  
## [[2]]  
## [1] "Inka Kola"  
##  
## [[3]]  
## [1] "Fanta"
```

También podemos ordenar los vectores de forma ascendente o descendente

```
# Creando vector
```

```
numeros_desordenados <- c(89, 132, 1, 7, 21, 42)
```

```
# Ordenando ascendente
```

```
numeros_ascendente <- sort(numeros_desordenados)
```

```
# Ordenando descendente
```

```
numeros_descendente <- sort(numeros_desordenados, decreasing = TRUE)
```

```
numeros_ascendente
```

```
## [1] 1 7 21 42 89 132
```

```
numeros_descendente
```

```
## [1] 132 89 42 21 7 1
```

También se pueden ordenar vectores de texto

```
# Mostrando lista  
países
```

```
## [1] "Perú"    "Ecuador" "Bolivia"
```

```
países_ordenados <- sort(países)  
países_desc <- sort(países, decreasing = TRUE)
```

```
países_ordenados
```

```
## [1] "Bolivia" "Ecuador" "Perú"
```

```
países_desc
```

```
## [1] "Perú"    "Ecuador" "Bolivia"
```

Listas nombradas

Los valores de las listas pueden estar nombrados

```
id_empleados <- c(1, 2, 3, 4, 5)
nombre_empleados <- c("Ana", "Bernardo", "Cintia", "David", "Emilia")
```

```
names(id_empleados) <- nombre_empleados
```

```
id_empleados
```

##	Ana	Bernardo	Cintia	David	Emilia
##	1	2	3	4	5

combinar listas

Dos listas se pueden combinar usando el comando `c()`

```
ciudades_1 <- c("Arequipa", "Huancayo", "Piura")  
ciudades_2 <- c("Ayacucho", "Puno", "Huancavelica")
```

```
ciudades_3 <- c(ciudades_1, ciudades_2)
```

```
ciudades_3
```

```
## [1] "Arequipa"      "Huancayo"      "Piura"         "Ayacucho"      "Puno"  
## [6] "Huancavelica"
```

Tablas (Data Frames)

- Los *data frames* o tablas son ordenamiento de datos de forma tabulada.
- Son la forma más popular de ordenar datos en R.
- Tienen 2 dimensiones: filas y columnas

Diagram illustrating the structure of a Data Frame (Table) with rows and columns.

Columns: The header row defines the columns: **Nombre**, **Equipo**, **Edad**, and **Posición**.

Rows: The data rows are indexed from 0 to 5.

	Nombre	Equipo	Edad	Posición
0	Miguel Vargas	Cienciano	26	Portero
1	Paolo Puentes	Cienciano	26	Defensa central
2	Alexis Cossio	Cienciano	27	Lateral izquierdo
3	Ángel Romero	Cienciano	31	NA
4	Kevin Sandoval	Cienciano	NA	Extremo izquierdo
5	Danilo Carando	Cienciano	33	Delantero centro

Annotations:

- Filas:** Indicated by purple arrows pointing to the row indices (0 to 5).
- Columnas:** Indicated by green arrows pointing to the column headers.
- Datos:** Indicated by red arrows pointing to the data cells (e.g., Miguel Vargas, Cienciano, 26, Portero).

Características

- Las columnas deben estar nombradas.
- Cada columna debe tener el mismo número de elementos.
- Cada elemento en una columna debe ser del mismo tipo (Cada columna es un vector)
- Diferentes columnas pueden tener diferentes tipos de datos.

Crear un data frame

```
nombres <- c("Alberto", "Bertha", "Camila")
edad <- c(15, 75, 34)
ciudad_nacimiento <- c("Cusco", "Lima", "Puno")
hijos <- c(TRUE, FALSE, FALSE)
```

```
tabla <- data.frame(nombres, edad, ciudad_nacimiento, hijos)
```

tabla

```
##  nombres edad ciudad_nacimiento hijos
## 1 Alberto  15                Cusco  TRUE
## 2 Bertha   75                Lima  FALSE
## 3 Camila   34                Puno  FALSE
```


Rstudio nos permite visualizar una tabla de forma más amigable con el comando `View()`

```
View(tabla)
```

Podemos ver la estructura de la tabla con el comando `str()`

```
str(tabla)
```

```
## 'data.frame':   3 obs. of  4 variables:
## $ nombres      : chr  "Alberto" "Bertha" "Camila"
## $ edad         : num  15 75 34
## $ ciudad_nacimiento: chr  "Cusco" "Lima" "Puno"
## $ hijos        : logi  TRUE FALSE FALSE
```

Usando el comando `summary()` podemos pedirle a R que nos de un resumen de la información de la tabla

```
summary(tabla)
```

```
##      nombres          edad      ciudad_nacimiento      hijos
## Length:3          Min.    :15.00      Length:3          Mode :logical
## Class :character  1st Qu.:24.50      Class :character    FALSE:2
## Mode  :character  Median :34.00      Mode  :character    TRUE :1
##                      Mean   :41.33
##                      3rd Qu.:54.50
##                      Max.    :75.00
```

El comando `dim()` nos indica las dimensiones de nuestro data frame

```
dim(tabla)
```

```
## [1] 3 4
```

Si tenemos tablas grandes podemos usar `head()` y `tail()` para ver solo los primeros (head) o últimos (tail) datos de nuestra tabla

```
# Cargamos una data de ejemplo de R  
data(USArrests)
```

```
head(USArrests)
```

```
##           Murder Assault UrbanPop Rape  
## Alabama      13.2      236        58 21.2  
## Alaska       10.0      263        48 44.5  
## Arizona       8.1      294        80 31.0  
## Arkansas      8.8      190        50 19.5  
## California    9.0      276        91 40.6  
## Colorado      7.9      204        78 38.7
```

```
tail(USArrests)
```

##	Murder	Assault	UrbanPop	Rape
## Vermont	2.2	48	32	11.2
## Virginia	8.5	156	63	20.7
## Washington	4.0	145	73	26.2
## West Virginia	5.7	81	39	9.3
## Wisconsin	2.6	53	66	10.8
## Wyoming	6.8	161	60	15.6

Podemos acceder a los vectores individuales de la tabla usando el nombre de la columna

```
tabla$nombr
```

```
## [1] "Alberto" "Bertha" "Camila"
```

```
tabla$edad
```

```
## [1] 15 75 34
```

Nombres de columnas

Las columnas de un dataframe estan nombradas. Podemos usar el comando `names()` para extraer los nombres y para cambiarlos

```
# Extrae nombres del data frame  
names(tabla)
```

```
## [1] "nombres"          "edad"              "ciudad_nacimiento"  
## [4] "hijos"
```

```
# Creamos un nuevo vector con nuevos nombres  
nuevos_nombres <- c("Nombre completo", "Edad", "Ciudad de Nacimiento", "¿Tiene hijos?")
```

```
# Asignamos nuevos nombres a tabla  
names(tabla) <- nuevos_nombres
```

```
# Verificamos cambio  
names(tabla)
```

```
## [1] "Nombre completo"    "Edad"              "Ciudad de Nacimiento"  
## [4] "¿Tiene hijos?"
```

Ejercicio

Crear un data frame de su familia. El data frame debe contener las siguientes columnas:

- Nombre Completo (cadena de texto)
- Edad (Número entero)
- Altura (Número decimal)
- ¿Está Casado? (Valor lógico, TRUE/FALSE)

Los nombres de las columnas deben cambiarse:

- Ej: en lugar de nombre debe decir **“Nombre Completo”**

Glosario

Combinar

```
# Combina elementos en un objeto  
abecedario <- c("a", "b", "c")
```

```
notas <- c(15, 20, 08)
```

Imprimir

```
# Imprime el objeto en la consola  
x <- 5  
print(x)
```

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

Crear una lista

```
# Crea una lista
```

```
presidentes <- list("Ollanta", "PPK", "Vizcarra")
```

typeof

El comando `typeof()` nos permite ver el tipo de un vector

```
vector2 <- c(1L, 4L, 2L, 5L)  
typeof(vector2)
```

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

names

Extrae los nombres de un vector o los asigna

```
nombres <- c("Jorge", "Karla", "Luis")  
edad <- c(36, 45, 18)
```

```
# Asigna nombres al vector  
names(edad) <- nombres
```

```
# Extrae los nombres del vector  
names(edad)
```

```
## [1] "Jorge" "Karla" "Luis"
```