

# Introducción a R y RStudio

## Estructuras de datos en R

Joselyn Chávez

@josschavezf

18/12/2020



Este material posee una licencia tipo Creative Commons Attribution-ShareAlike 4.0 International License.

Para conocer más sobre esta licencia, visite:

<http://creativecommons.org/licenses/by-sa/4.0/>



# Estructuras de datos en R

# R opera en estructuras de datos

- Vectores
- Matrices
- Listas
- Data frames
- Funciones

En la primera sección de este curso aprendimos a usar vectores y manejar su contenido.



# Matrices

Una matriz es una estructura rectangular, que contiene datos de un solo tipo.

Son usadas principalmente en operaciones matemáticas y estadística.

Para crear una matriz usaremos la función **matrix()**.

Si usamos esta función sin proporcionar el número de renglones o columnas que deseamos crear, por defecto hará el acomodo de los datos en una sola columna.

```
matrix(1:12)
```

```
##           [,1]
## [1,]         1
## [2,]         2
## [3,]         3
## [4,]         4
## [5,]         5
## [6,]         6
## [7,]         7
## [8,]         8
## [9,]         9
## [10,]        10
## [11,]        11
## [12,]        12
```



Si especificamos el número de renglones que deseamos en la matriz, el arreglo será diferente.

```
matrix(1:12, nrow = 3)
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    1    4    7   10  
## [2,]    2    5    8   11  
## [3,]    3    6    9   12
```

Notemos que los datos comienzan a llenar las columnas de arriba a abajo y de izquierda a derecha.



Si le indicamos un número de columnas y renglones menor al número de datos, se usarán los que cubran los espacios en la matriz y el resto se descartará

```
matrix(1:12, nrow = 3, ncol = 3)
```

```
##      [,1] [,2] [,3]  
## [1,]    1    4    7  
## [2,]    2    5    8  
## [3,]    3    6    9
```

Por otro lado, si le indicamos un número de columnas y renglones que sobrepasa el número de datos disponibles, los datos se reciclan.

```
matrix(1:12, nrow = 4, ncol = 4)
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    1    5    9    1  
## [2,]    2    6   10    2  
## [3,]    3    7   11    3  
## [4,]    4    8   12    4
```





# Operaciones con matrices

Podemos aplicar operaciones matemáticas a las matrices

```
matriz1 <- matrix(1:12, nrow = 3)
```

## Sumas y restas

```
matriz1 + 1
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    2    5    8   11  
## [2,]    3    6    9   12  
## [3,]    4    7   10   13
```



# Multiplicación y división

```
matriz1 * 2
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    2    8   14   20  
## [2,]    4   10   16   22  
## [3,]    6   12   18   24
```

# Potenciación

```
matriz1 ^ 3
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    1   64  343 1000  
## [2,]    8  125  512 1331  
## [3,]   27  216  729 1728
```



# Transposición

```
t(matriz1)
```

```
##      [,1] [,2] [,3]  
## [1,]    1    2    3  
## [2,]    4    5    6  
## [3,]    7    8    9  
## [4,]   10   11   12
```



# Operaciones entre matrices

```
matriz1 + matriz1
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    2    8   14   20  
## [2,]    4   10   16   22  
## [3,]    6   12   18   24
```

```
matriz1 * matriz1
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    1   16   49  100  
## [2,]    4   25   64  121  
## [3,]    9   36   81  144
```



# Listas

# ¿Qué es una lista?

Una lista es una colección ordenada de objetos, conocidos como componentes.

## ¿Cuáles son sus características?

- Los componentes pueden ser de diferente tamaño y clase.
- Las listas pueden contener vectores, matrices, cadenas de caracteres, data frames, funciones, etc.



Para crear una lista usaremos la función `list()`.

Veamos un ejemplo:

```
frutas <- list(fruta = "manzana", unidades = 5,  
              características = c("roja", "pequeña", "dulce") )
```

```
frutas
```

```
## $fruta  
## [1] "manzana"  
##  
## $unidades  
## [1] 5  
##  
## $características  
## [1] "roja"      "pequeña"  "dulce"
```



Observemos que sus componentes se encuentran numerados, de manera que podemos referirnos a ellos por su posición en la lista.

¿Cómo extraemos el elemento "fruta"?

```
frutas[1]
```

```
## $fruta  
## [1] "manzana"
```

```
frutas[[1]]
```

```
## [1] "manzana"
```

¿Cuál es la diferencia entre estas dos opciones? Vamos a revisar de qué clase es cada una de ellas

```
class(frutas[1])
```

```
## [1] "list"
```

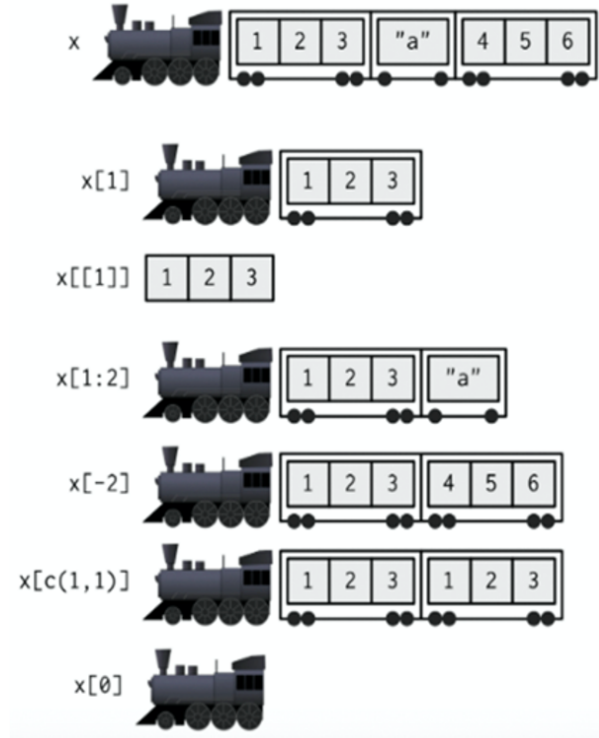
```
class(frutas[[1]])
```

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





# ¿Por qué son salidas diferentes?



Tomado de [Chapter 4 Subsetting. Advanced R. Second ed.](#) by Hadley Wickham



Ahora, vamos a explorar los elementos del componente "características".

Si el componente tiene múltiples elementos, usamos una segunda numeración para referirnos al elemento deseado.

Para extraer la primera característica de la manzana (el color):

```
frutas[[3]][1]
```

```
## [1] "roja"
```

¿Cómo extraerías la característica que se refiere al dulzor de la fruta?



Otra forma de extraer los componentes de una lista es usando el símbolo \$ y el nombre del componente que deseamos extraer.

```
frutas$unidades
```

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

```
frutas$caracteristicas
```

```
## [1] "roja"      "pequeña" "dulce"
```

Esto es muy útil porque puede ser más fácil recordar el nombre del componente que su posición en la lista.

Ahora, ¿cómo extraemos los elementos del componente característica usando \$?

```
frutas$caracteristicas[1]
```

```
## [1] "roja"
```

¿Cómo extraerías la característica que se refiere al dulzor de la fruta usando \$ ?



# Hagamos un pequeño ejercicio

Generemos una nueva lista:

```
jugadores <- list(nombre = c("Julie", "Juan", "Sofia"),  
                  edad = c(22, 29, 34),  
                  estatura = c(1.62, 1.75, 1.65))
```

¿Cómo podemos conocer la estatura de Sofía?

¿Cómo podemos obtener la edad de Julie y Juan al mismo tiempo?



Como alternativa, podemos usar el nombre del componente dentro de los corchetes.

```
jugadores[["edad"]][1:2]
```

```
## [1] 22 29
```



# Data frames

# ¿Qué es un data frame?

Un data frame es una lista de vectores de la misma longitud en una estructura rectangular, que R representa de forma similar a una tabla.

## ¿Cuáles son sus características?

- Su información está organizada en renglones y columnas.
- Sus vectores deben tener la misma longitud.
- Cada columna puede contener un tipo diferente de datos.



Para crear un data frame usaremos la función **data.frame()**.

Veamos un ejemplo:

```
clima <- data.frame(dia = c(1,2,3,4,5),  
                    temperatura = c(26,25,26,27,27),  
                    nublado = c(1,0,0,1,1),  
                    viento = c(5,3,2,3,6) )  
  
clima
```

```
##   dia temperatura nublado viento  
## 1    1          26       1      5  
## 2    2          25       0      3  
## 3    3          26       0      2  
## 4    4          27       1      3  
## 5    5          27       1      6
```





# ¿Cuáles son las dimensiones del data frame?

**dim()** nos mostrará el número de filas y columnas del data frame.

```
dim(clima)
```

```
## [1] 5 4
```

**nrow()** nos mostrará el número de filas.

```
nrow(clima)
```

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

**ncol()** nos mostrará el número de columnas.

```
ncol(clima)
```

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



# ¿Cuál es el nombre de las columnas y renglones de mi data frame?

**rownames()** nos muestra el nombre de los renglones.

```
rownames(clima)
```

```
## [1] "1" "2" "3" "4" "5"
```

**colnames()** nos muestra el nombre de las columnas.

```
colnames(clima)
```

```
## [1] "dia"          "temperatura" "nublado"      "viento"
```



Estas funciones también nos ayudan a cambiar los nombres de las columnas y renglones

```
colnames(clima)[4] <- "velocidad_viento"

rownames(clima) <- c("Lunes", "Martes", "Miércoles",
                    "Jueves", "Viernes")

clima
```

##	dia	temperatura	nublado	velocidad_viento
## Lunes	1	26	1	5
## Martes	2	25	0	3
## Miércoles	3	26	0	2
## Jueves	4	27	1	3
## Viernes	5	27	1	6



# Otras formas de crear data frames

Podemos crear un data frame a partir de una lista, siempre que sus componentes tengan la misma longitud

```
df_jugadores <- as.data.frame(jugadores)
df_jugadores
```

```
##   nombre edad estatura
## 1  Julie   22    1.62
## 2   Juan   29    1.75
## 3  Sofia   34    1.65
```



Podemos importar un data frame desde un archivo usando la función `read.table()`.

```
pinguinos <- read.table("pinguinos.txt", header = TRUE)
head(pinguinos)
```

```
##   especie      isla largo_pico_mm alto_pico_mm  sexo anio
## 1  Adelia    Biscoe          37.8         20.0 macho 2009
## 2  Papúa     Biscoe          50.5         15.2 hembra 2009
## 3 Barbijo    Dream          42.5         16.7 hembra 2008
## 4  Papúa     Biscoe          49.1         14.8 hembra 2008
## 5  Adelia Torgersen          35.1         19.4 macho 2008
## 6  Papúa     Biscoe          48.2         14.3 hembra 2007
```

Esta función tiene diversas opciones que podemos consultar en la ayuda con `help("read.table")`.



# ¿Cómo exporto mi data frame?

Podemos usar la función **write.table()**.

```
write.table(jugadores, "jugadores.txt",  
            quote = FALSE, row.names = FALSE)
```

Para ver los parámetros disponibles en esta función consultemos la ayuda con `help("write.table")`.



# ¿Cómo seleccionamos elementos del data frame?

La sintaxis para acceder a los elementos de un data frame es

`nombre_del_dataframe[renglones, columnas]`



Veamos el renglón 5 del data frame pingüinos

```
pinguinos[5,]
```

```
##   especie      isla largo_pico_mm alto_pico_mm  sexo anio
## 5  Adelia Torgersen          35.1          19.4 macho 2008
```

Veamos la columna 1 del data frame pingüinos

```
pinguinos[,1]
```

```
## [1] "Adelia"  "Papúa"   "Barbijo" "Papúa"   "Adelia"  "Papúa"   "Adelia"
## [8] "Adelia"  "Papúa"   "Barbijo" "Papúa"   "Barbijo" "Barbijo"  "Papúa"
## [15] "Adelia"  "Adelia"  "Barbijo" "Barbijo" "Barbijo" "Papúa"
```

Veamos los primeros 5 renglones

```
pinguinos[1:5,]
```

```
##   especie      isla largo_pico_mm alto_pico_mm  sexo anio
## 1  Adelia  Biscoe          37.8          20.0 macho 2009
## 2  Papúa   Biscoe          50.5          15.2 hembra 2009
## 3 Barbijo  Dream          42.5          16.7 hembra 2008
## 4  Papúa   Biscoe          49.1          14.8 hembra 2008
## 5  Adelia Torgersen          35.1          19.4 macho 2008
```





Veamos los primeros 5 renglones y sólo la primera y segunda columna

```
pinguinos[1:5,1:2]
```

```
##   especie      isla
## 1  Adelia      Biscoe
## 2  Papúa       Biscoe
## 3 Barbijo      Dream
## 4  Papúa       Biscoe
## 5  Adelia      Torgersen
```

Veamos los primeros 5 renglones y las columnas 1,3 y 5

```
pinguinos[1:5,c(1,3,5)]
```

```
##   especie largo_pico_mm  sexo
## 1  Adelia           37.8 macho
## 2  Papúa           50.5 hembra
## 3 Barbijo           42.5 hembra
## 4  Papúa           49.1 hembra
## 5  Adelia           35.1 macho
```



Veamos todo el set de datos, excepto la columna 5

```
pinguinos[, -5]
```

##	especie	isla	largo_pico_mm	alto_pico_mm	anio
## 1	Adelia	Biscoe	37.8	20.0	2009
## 2	Papúa	Biscoe	50.5	15.2	2009
## 3	Barbijo	Dream	42.5	16.7	2008
## 4	Papúa	Biscoe	49.1	14.8	2008
## 5	Adelia	Torgersen	35.1	19.4	2008
## 6	Papúa	Biscoe	48.2	14.3	2007
## 7	Adelia	Biscoe	42.7	18.3	2009
## 8	Adelia	Biscoe	43.2	19.0	2009
## 9	Papúa	Biscoe	50.8	15.7	2009
## 10	Barbijo	Dream	50.3	20.0	2007
## 11	Papúa	Biscoe	45.1	14.4	2008
## 12	Barbijo	Dream	55.8	19.8	2009
## 13	Barbijo	Dream	46.9	16.6	2008
## 14	Papúa	Biscoe	51.5	16.3	2009
## 15	Adelia	Torgersen	40.9	16.8	2008
## 16	Adelia	Dream	42.3	21.2	2007
## 17	Barbijo	Dream	46.4	17.8	2008
## 18	Barbijo	Dream	50.8	18.5	2009
## 19	Barbijo	Dream	50.5	19.6	2007
## 20	Papúa	Biscoe	50.0	15.3	2007



También podemos seleccionar los datos usando su nombre de renglón o columna

```
pinguinos["1", "especie"]
```

```
## [1] "Adelia"
```



Así como en las listas, también podemos usar el símbolo \$ para seleccionar una columna del data frame

```
pinguinos$isla
```

```
## [1] "Biscoe" "Biscoe" "Dream" "Biscoe" "Torgersen" "Biscoe"  
## [7] "Biscoe" "Biscoe" "Biscoe" "Dream" "Biscoe" "Dream"  
## [13] "Dream" "Biscoe" "Torgersen" "Dream" "Dream" "Dream"  
## [19] "Dream" "Biscoe"
```

Y seleccionar elementos contenidos en la columna

```
pinguinos$isla[1:3]
```

```
## [1] "Biscoe" "Biscoe" "Dream"
```

## En el data frame "clima" ¿Cómo obtendrías el valor de la temperatura del Viernes?



# Explorando los datos del data frame

Existen algunas funciones que nos ayudan a darle un vistazo a los datos:

**attributes()** nos muestra el nombre de las columnas y renglones, así como la clase de nuestro objeto.

```
attributes(pinguinos)
```

```
## $names
## [1] "especie"      "isla"          "largo_pico_mm" "alto_pico_mm"
## [5] "sexo"        "anio"
##
## $class
## [1] "data.frame"
##
## $row.names
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```



**str()** nos muestra la clase de cada vector que compone nuestro data frame.

```
str(pinguinos)
```

```
## 'data.frame':    20 obs. of  6 variables:
##  $ especie      : chr  "Adelia" "Papúa" "Barbijo" "Papúa" ...
##  $ isla         : chr  "Biscoe" "Biscoe" "Dream" "Biscoe" ...
##  $ largo_pico_mm: num  37.8 50.5 42.5 49.1 35.1 48.2 42.7 43.2 50.8 50.3 .
##  $ alto_pico_mm : num  20 15.2 16.7 14.8 19.4 14.3 18.3 19 15.7 20 ...
##  $ sexo         : chr  "macho" "hembra" "hembra" "hembra" ...
##  $ anio         : int  2009 2009 2008 2008 2008 2007 2009 2009 2009 2007 .
```



**summary()** nos muestra datos estadísticos de cada uno de los vectores en el data frame.

```
summary(pinguinos)
```

```
##      especie              isla      largo_pico_mm      alto_pico_mm
## Length:20      Length:20      Min.      :35.10      Min.      :14.30
## Class :character      Class :character      1st Qu.:42.65      1st Qu.:15.60
## Mode  :character      Mode  :character      Median :47.55      Median :17.30
##                                         Mean   :46.52      Mean   :17.48
##                                         3rd Qu.:50.50      3rd Qu.:19.45
##                                         Max.    :55.80      Max.    :21.20
##      sexo              anio
## Length:20      Min.      :2007
## Class :character      1st Qu.:2008
## Mode  :character      Median :2008
##                                         Mean   :2008
##                                         3rd Qu.:2009
##                                         Max.    :2009
```



# Filtrando datos de un data frame

Los corchetes nos ayudan a seleccionar datos que cumplen con una condición.

Seleccionemos los datos de los pingüinos del año 2009

```
pinguinos[pinguinos$anio == 2009,]
```

##	especie	isla	largo_pico_mm	alto_pico_mm	sexo	anio
## 1	Adelia	Biscoe	37.8	20.0	macho	2009
## 2	Papúa	Biscoe	50.5	15.2	hembra	2009
## 7	Adelia	Biscoe	42.7	18.3	macho	2009
## 8	Adelia	Biscoe	43.2	19.0	macho	2009
## 9	Papúa	Biscoe	50.8	15.7	macho	2009
## 12	Barbijo	Dream	55.8	19.8	macho	2009
## 14	Papúa	Biscoe	51.5	16.3	macho	2009
## 18	Barbijo	Dream	50.8	18.5	macho	2009





Seleccionemos los datos de los pingüinos del año 2009 y tomemos sólo el nombre de la especie

```
pinguinos[pinguinos$anio == 2009,1]
```

```
## [1] "Adelia" "Papúa" "Adelia" "Adelia" "Papúa" "Barbijo" "Papúa"  
## [8] "Barbijo"
```

Otra forma de hacerlo

```
pinguinos[pinguinos$anio == 2009,"especie"]
```

```
## [1] "Adelia" "Papúa" "Adelia" "Adelia" "Papúa" "Barbijo" "Papúa"  
## [8] "Barbijo"
```



Seleccionemos los pingüinos con un largo de pico mayor a la media

```
media_pico <- mean(pinguinos$largo_pico_mm)

pinguinos[pinguinos$largo_pico_mm > media_pico,]
```

##	especie	isla	largo_pico_mm	alto_pico_mm	sexo	anio
## 2	Papúa	Biscoe	50.5	15.2	hembra	2009
## 4	Papúa	Biscoe	49.1	14.8	hembra	2008
## 6	Papúa	Biscoe	48.2	14.3	hembra	2007
## 9	Papúa	Biscoe	50.8	15.7	macho	2009
## 10	Barbijo	Dream	50.3	20.0	macho	2007
## 12	Barbijo	Dream	55.8	19.8	macho	2009
## 13	Barbijo	Dream	46.9	16.6	hembra	2008
## 14	Papúa	Biscoe	51.5	16.3	macho	2009
## 18	Barbijo	Dream	50.8	18.5	macho	2009
## 19	Barbijo	Dream	50.5	19.6	macho	2007
## 20	Papúa	Biscoe	50.0	15.3	macho	2007

¿Cómo seleccionarías a los pingüinos que viven en la isla "Dream" ?



# Otros data frames en R

R contiene sets de datos de ejemplo, como `airquality`, que es un data frame.

Podemos acceder a sets de datos en español con el paquete "datos" que podemos instalar desde CRAN con:

```
install.packages("datos")
```



# ¡Gracias!

Estas diapositivas fueron hechas utilizando el paquete [xaringan](#) y configuradas con el tema [rmb](#).

Puedes consultar el material en [https://github.com/ComunidadBioInfo/minicurso\\_dic2020](https://github.com/ComunidadBioInfo/minicurso_dic2020)

