

Programación en R para el análisis de datos

Repaso

Nicolás Schmidt

`mail::nschmidt@cienciassociales.edu.uy`

`GitHub::@Nicolas-Schmidt`

Departamento de Ciencia Política

Facultad de Ciencias Sociales

1. Estructuras y tipos de datos
2. Identificadores y coercionadores
3. Indexación

Estructuras y tipos de datos

Estructuras de datos en R

Dimensiones	Homogéneo	Heterogéneo
1	Vector atómico	List
2	Matrix	Data Frame
n	Array	

Fuente: Wickham, Hadley. *Advanced R*

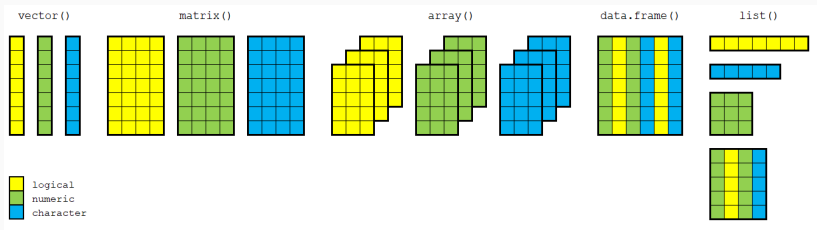
Tipos de datos: `typeof()`

Los tipos de objetos más frecuentes son estos:

<code>typeof*</code>	Descripción	Ejemplo
<code>NULL</code>	NULO	<code>a <- NULL</code>
<code>closure</code>	Una función	<code>a <- function(x){x*pi}</code>
<code>logical</code>	un vector que contiene valores lógicos	<code>a <- c(TRUE, FALSE, TRUE)</code>
<code>integer</code>	un vector que contiene valores enteros	<code>a <- c(1:10)</code>
<code>double</code>	un vector que contiene valores reales	<code>a <- seq(1:10, 0.3)</code>
<code>complex</code>	un vector que contiene valores complejos	<code>a <- sqrt(-17+0i)</code>
<code>character</code>	un vector que contiene valores de caracteres	<code>a <- letters[1:5]</code>
<code>list</code>	una lista	<code>a <- list(b=1:10, c="j")</code>

*Nota: en este [link](#) está la lista completa de posibles respuestas de la función `typeof()`.

Estructuras y tipos de datos en R



Fuente: adaptación de Friendly, Michael, Meyer, David (2015)

Identificadores y coercionadores

Identificadores y coercionadores

Tipo	Verificación	Cambio
vector	<code>is.vector()</code>	<code>as.vector()</code>
list	<code>is.list()</code>	<code>as.list()</code>
data.frame	<code>is.data.frame()</code>	<code>as.data.frame()</code>
matrix	<code>is.matrix()</code>	<code>as.matrix()</code>
logical	<code>is.logical()</code>	<code>as.logical()</code>
factor	<code>is.factor()</code>	<code>as.factor()</code>
character	<code>is.character()</code>	<code>as.character()</code>
numeric	<code>is.numeric()</code>	<code>as.numeric()</code>
double	<code>is.double()</code>	<code>as.double()</code>

Para ver todas las opciones disponibles

```
ls(patter = "^is.", baseenv())  
ls(patter = "^as.", baseenv())
```


Coerciones implícitas

Usemos la funciønn '`ci()`' del paquete '`IntRo`':

```
ci()
```

```
##          logical  integer  double  raw      complex  character
##          -----  -
## logical  | logical
## integer  | integer  integer
## double   | double   double   double
## raw      | logical  integer  double   raw
## complex  | complex  complex  complex  complex  complex
## character| character character character character character character
```

Indexación

Hay 4 maneras básicas de indexación: esto es de acceso a una parte dentro de una estructura.

Dependiendo de la estructura es la forma en la que se debe indexar. Y dependiendo de la manera de indexar es la estructura que va a permanecer

'[]', '[,]', '[[]]', '\$'