

Semana 8

Repaso

Los tipos de datos de uso más común en R son los siguientes.

Tipo	Ejemplo	Nombre en inglés
Entero	1	integer
Numérico	1.3	numeric
Cadena de texto	"uno"	character
Factor	uno	factor
Lógico	TRUE	logical
Perdido	NA	NA
Vacio	NULL	null

Entero y numérico

Enteros: representan números sin una parte decimal o fraccionaria. Pueden ser usados en operaciones matemáticas.

Numérico representan números enteros que pueden incluir una parte decimal o fraccionaria

Cadena de texto

character representa texto. En R este se encuentra entre comillas simples o dobles (Puede contener letras, números, espacios, signos de puntuación, símbolos especiales)

Factor

Un dato representado por una etiqueta. Por ejemplo: los números 0 y 1 expresa hembra y macho en el sexado de especies. De esta manera reducimos el espacio de almacenamiento en nuestros datos.

Lógico

Representan si una condición se cumple o no (TRUE y FALSE). Solo admite dos valores.

NA y NULL

Se usa *NA* para representar datos perdidos mientras que *NULL* representa la ausencia de datos.

Coerción

En R, los datos pueden ser coercionados, es decir, forzados, para transformarlos de un tipo a otro. Habrá ocasiones en que la operación ocurra sin problemas, y otras en las que falle y se obtenga un error. Esto ocurre debido a que existe una jerarquía:

- lógico
- entero
- numérico
- caracter

```
x <- c(letters[1:10])  
class(x)
```

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

```
as.numeric(x)
```

```
## [1] NA NA NA NA NA NA NA NA NA NA
```

Las coerciones no pueden ocurrir en orden inverso.

Función	Tipo al que hace coerción
<code>as.integer()</code>	Entero
<code>as.numeric()</code>	Numerico
<code>as.character()</code>	Cadena de texto
<code>as.factor()</code>	Factor
<code>as.logical()</code>	Lógico
<code>as.null()</code>	NULL

Verificar un tipo de dato

Para este fin podemos usar la función `class()` de R. Esto es de utilidad para asegurarnos que las operaciones que deseamos realizar tendrán los datos apropiados para llevarse a cabo con éxito.

```
class(3)
```

```
## [1] "numeric"
```

```
class(TRUE)
```

```
## [1] "logical"
```

```
class("3")
```

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

También podemos verificar si un dato es de un tipo específicos con la familia de funciones `is()`.

Función	Tipo que verifican
<code>is.integer()</code>	Entero
<code>is.numeric()</code>	Numerico
<code>is.character()</code>	Cadena de texto
<code>is.factor()</code>	Factor
<code>is.logical()</code>	Lógico
<code>is.na()</code>	NA
<code>is.null()</code>	NULL

```
is.numeric(3)
```

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

```
is.character("5")
```

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

Operadores

Los operadores son los símbolos que le indican a R que debe realizar una tarea. Combinando datos y operadores logramos que R haga su trabajo.

- Aritméticos
- Relacionales
- Lógicos
- De asignación

Operadores aritméticos

Es usado para realizar operaciones aritméticas.

Operador	Operación	Ejemplo	Resultado
+	Suma	5 + 3	8
-	Resta	5 - 3	2
*	Multiplicación	5 * 3	18
/	División	5 / 3	1.666667
^	Potencia	5 ^ 3	125
%%	División entera	5 %% 3	2

Operadores relacionales

Son usado para hacer comparaciones y siempre devuelven como resultado *TRUE* y *FALSE*.

Operador	Comparación	Ejemplo	Resultado
<	Menor que	5 < 3	FALSE
<=	Menor o igual que	5 <= 3	FALSE
>	Mayor que	5 > 3	TRUE
>=	Mayor o igual que	5 >= 3	TRUE
==	Exactamente igual que	5 == 3	FALSE
!=	No es igual que	5 != 3	TRUE

Operadores lógicos

Describen relaciones lógicas, expresadas como verdadero o falso.

Operador	Comparación	Ejemplo	Resultado
x y	x Ó y es verdadero	TRUE FALSE	TRUE
x & y	x Y y son verdaderos	TRUE & FALSE	FALSE
!x	x no es verdadero (negación)	!TRUE	FALSE
isTRUE(x)	x es verdadero (afirmación)	isTRUE(TRUE)	TRUE

Los operadores “|” y “&” siguen estas reglas:

- devuelve TRUE si alguno de los datos es TRUE
- & solo devuelve TRUE si ambos datos son TRUE
- solo devuelve FALSE si ambos datos son FALSE
- & devuelve FALSE si alguno de los datos es FALSE

Estructura de datos

Las estructuras de datos son objetos que contienen datos. Tienen diferentes características entre ellas son: su número de **dimensiones** y si son **homogéneas** o **heterogénea**.

Dimensiones	Homogeneas	Heterogeneas
1	Vector	Lista
2	Matriz	Data frame
n	Array	

```
#VECTORES  
#vector numérico  
c(1,2,4,5,6)
```

```
## [1] 1 2 4 5 6
```

```
#vector caracter  
c("casa","persona","computador")
```

```
## [1] "casa"      "persona"    "computador"
```

```
#vector lógico  
c(TRUE, FALSE, TRUE, TRUE)
```

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

```
#MATRICES - ARRAYS  
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
```

```
matrix(1:12, nrow = 3, ncol = 4)
```

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

Dos matrices A y B solo podrán ser multiplicadas si el número de columnas de la matriz A coincide con el número de filas de la matriz B. Por ejemplo:

```
mi_matriz1 <- matrix(1:12, nrow = 4, ncol = 3)
mi_matriz2 <- matrix(1:30, nrow = 3, ncol = 10)
```

```
#Multiplicación de matrices = Operados %*%
mi_matriz1 %*% mi_matriz2
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]  38   83  128  173  218  263  308  353  398  443
## [2,]  44   98  152  206  260  314  368  422  476  530
## [3,]  50  113  176  239  302  365  428  491  554  617
## [4,]  56  128  200  272  344  416  488  560  632  704
```

```
#Operador de división cuyo resultado es el residuo
5%%3
```

```
## [1] 2
```

Dataframe

Son estructuras de datos de dos dimensiones. Pueden contener datos de diferentes tipos o clases, por tanto son heterogéneas. Esta estructura es la más usada para realizar análisis de datos.

```
desembarques <- read.csv("sample_landings_data_raw.csv")
head(desembarques, 4)
```

```
##      yy      dat trip effort  gr      sp l_cm      w_cm
## 1 2003 4/30/03     1     10 Trap Caesoi cunning   36 1089.1402
## 2 2003 4/30/03     1     10 trap Caesio cuning   29  565.3879
## 3 2003 4/30/03     1     10 Trap Caesio cuning   34  915.8276
## 4 2003 4/30/03     1     10 Trap Caesio cuning   36 1089.1402
```

```
#renombrar variables
```

```
namesNew <- c("año", "fecha", "viaje_ID", "esfuerzo_hora", "aparejo", "especies", "tall_cm", "peso")
names(desembarques) <- namesNew
```

```
#agregar columnas
```

```
desembarques$id2 <- paste(desembarques$aparejo, desembarques$especies, sep="_")
head(desembarques, 4)
```

```
##      año  fecha viaje_ID esfuerzo_hora aparejo      especies tall_cm      peso
## 1 2003 4/30/03         1          10   Trap Caesoi cunning   36 1089.1402
## 2 2003 4/30/03         1          10   trap Caesio cuning   29  565.3879
## 3 2003 4/30/03         1          10   Trap Caesio cuning   34  915.8276
## 4 2003 4/30/03         1          10   Trap Caesio cuning   36 1089.1402
##
##      id2
## 1 Trap_Caesoi cunning
## 2 trap_Caesio cuning
## 3 Trap_Caesio cuning
## 4 Trap_Caesio cuning
```

```
#valores perdidos
```

```
desembarques[!complete.cases(desembarques),]
```

```
##      año   fecha viaje_ID esfuerzo_hora aparejo   especies tall_cm
## 28   2003   5/1/03      10             10 Handline Caesio cuning 19.000
## 1637 2004 12/18/04      NA             9   Trap Caesio cuning 20.104
##           peso           id2
## 28           NA Handline_Caesio cuning
## 1637 186.1558   Trap_Caesio cuning
```

```
#valores tipos
```

```
unique(desembarques$aparejo)
```

```
## [1] "Trap"      "trap"      "Muroami"   ""           "Handline" "Gillnet"  "Trolling"
## [8] "Speargun"
```

```
desembarques$aparejo <- tolower(desembarques$aparejo)
```

```
unique(desembarques$especies)
```

```
## [1] "Caesoi cunning" "Caesio cuning"
```

```
table(desembarques$especies == "Caesoi cunning")
```

```
##
## FALSE TRUE
## 7212    2
```

```
table(desembarques$especies == "Caesio cuning")
```

```
##
## FALSE TRUE
##      2 7212
```

```
desembarques[desembarques$especies == "Caesoi cunning",]
```

```
##      año   fecha viaje_ID esfuerzo_hora aparejo   especies tall_cm   peso
## 1 2003 4/30/03      1             10   trap Caesoi cunning    36 1089.1402
## 6 2003 4/30/03      1             10   trap Caesoi cunning    28  508.3185
##           id2
## 1 Trap_Caesoi cunning
## 6 Trap_Caesoi cunning
```

```
#cambiando nombre variable
```

```
desembarques$especies[desembarques$especies == "Caesoi cunning"] <- "Caesio cuning"
```

```
#quitar valores perdidos
```

```
desembarques <- na.omit(desembarques)
```