



Curso: Introducción a R

Imparte: Cynthia Elizabeth Castillo Silva

Estructuras de Datos

¿Cómo organizo mi información?



Estructuras de Datos

¿Cómo organizo mi información?



Estructuras de Datos

Son una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente.

Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas.

¿Cómo organizarías la información de una secuencia de números en una misma variable? ¿Qué tal con una tabla que tenga la información del grupo del curso de R? ¿Y la información de la discografía de tu artista favorito?



Estructuras de Datos

Son una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente.

Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas.

¿Cómo organizarías la información de una secuencia de números en una misma variable? ¿Qué tal con una tabla que tenga la información del grupo del curso de R? ¿Y la información de la discografía de tu artista favorito?



Estructuras de Datos

Tienen diferentes clasificaciones de acuerdo a su cantidad de dimensiones y a los tipos de datos que contienen.

Dimensiones	Homogéneas	Heterogéneas
1	vector	list
2	matrix	dataframe
n	array	



Estructuras de Datos

Vectores



Vectores

Colección de uno o más datos del mismo tipo.

Propiedades:

1. **Tipo.** Sólo pueden contener datos del mismo tipo.
2. **Longitud.** Cantidad de elementos del vector.
3. **Atributos.** Metadatos de los vectores. No lo veremos en el curso.

Tipos de Vectores:

Tantos como tipos de datos conozcamos.



Funciones relacionadas con los vectores

Crear un vector

```
vector(mode, length)
```

Combinar/concatenar varios vectores

```
c()
```

Verificar que una variable sea un vector

```
is.vector()
```



Funciones relacionadas con los vectores

Conocer la longitud de un vector
`length()`

Conocer el tipo de dato del vector
`typeof()`



Secuencias Numéricas

Crear un vector de secuencia numérica

:

```
seq(from, to, by, length.out)
```

```
rep(x, times=1, length.out=NA, each=1)
```



NA, NULL, NaN y otros valores especiales

NA indica que el valor no fue registrado para algún registro (Not Available). Valor faltante intencional.

NULL indica que el valor no pudo ser recuperado. Valor faltante no intencional.

NaN indica que el resultado no es un número (Not a Number).

Inf -Inf indican valores que matemáticamente arrojarían estos valores en matemáticas.



NA, NULL, NaN y otros valores especiales

NA tiene varios subtipos dependiendo del tipo de dato del vector para facilitar la coerción.

`NA_integer_`, `NA_real_`, `NA_character_`, `NA_complex_`

NULL tiene longitud 0 y por lo tanto no tiene tipo.

NaN **Inf** **-Inf** son siempre de tipo numérico por ser resultados de cálculos numéricos.



NA, NULL, NaN y otros valores especiales

NA responde "TRUE" a la primera y "FALSE" a la segunda.
`is.na()`, `is.nan()`

NULL responde "TRUE".
`is.null()`

NaN responde "TRUE" a ambas.
`is.na()`, `is.nan()`

Inf -Inf responden "TRUE" a la primera y "FALSE" a la segunda.
`is.infinite()`, `is.finite()`



Coerción

Recordemos que se realiza de los tipos de datos más restringidos a los más flexibles.

```
a <- c(T, F)
```

```
b <- c(1, 2, 3)
```

```
c <- c("verde", "azul", "amarillo", "rojo")
```

```
d <- c(a, b, c)
```



Vectorización de operaciones

Hay operaciones que se aplican a todos los elementos del vector. A este proceso se le conoce como **vectorización**.

Aunque se llame "vectorización", también funciona con otras estructuras de datos. Más adelante veremos cómo.

