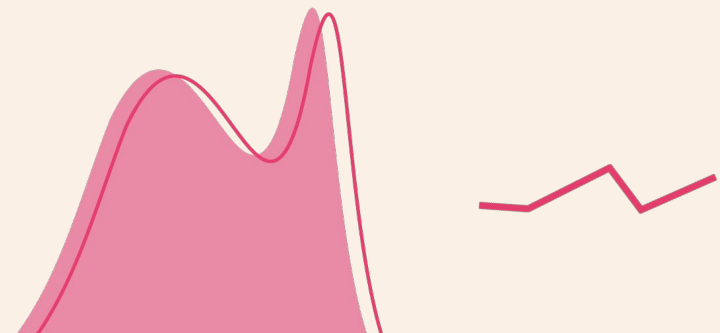


¡BIENVENIDOS!

Introducción a la Programación en R

Primeros Pasos.



Acuerdos

- ✓ Escucha activa
- ✓ Puntualidad
- ✓ Avisar quiénes participarán en el curso de manera virtual
- ✓ Apertura al aprendizaje
- ✓ Toda pregunta es bienvenida

The background is a solid orange color. There are several white circles of different sizes scattered across the slide. Three are in the top-left corner, and three are in the bottom-right corner.

Clase 01. R

R y Rstudio

Temario 01 - A

- ✓ R y Rstudio
- ✓ Aplicaciones e instalación de R y Rstudio
- ✓ Componentes de Rstudio
- ✓ Librerías y paquetes

Hoja de Ruta

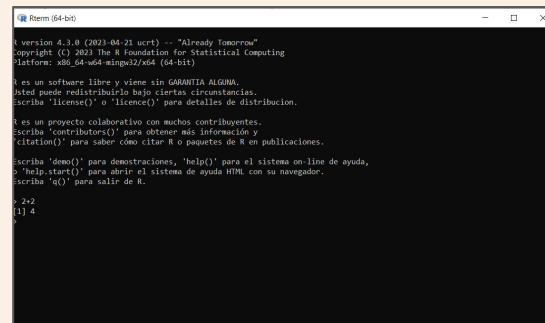


- R yRstudio: ¿Qué es? ¿Para qué sirve? Aplicaciones.
- Instancia práctica: Instalación de la herramienta.
- Componentes de Rstudio: Demostración de la herramienta.
- ¿Qué es un paquete?
- Dudas y comentarios.

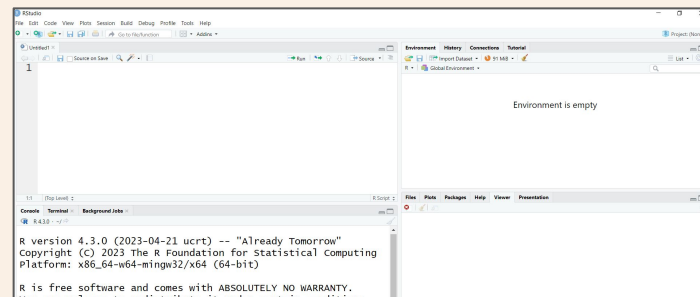
R y Rstudio



lenguaje de
programación

A screenshot of the R console window. It displays the R version 4.3.0 (2023-04-21 ucrt) -- "Already Tomorrow" and copyright information. It also shows the R license text in Spanish, stating that R is free software and comes with absolutely no warranty. The prompt is 2> and the input is 4.

IDE: entorno
integrado de
desarrollo



Temario 01 - B



- ✓ Tipos de datos
- ✓ Asignación de variables
- ✓ Operadores
- ✓ Funciones
- ✓ Objetos

Tipos de datos

Tipos de datos principales

- ✓ **Numeric:** Números enteros y reales.
- ✓ **Character:** Es una variable de tipo texto.
- ✓ **Factor:** Es una variable categórica que se utiliza para representar grupos.
- ✓ **Logical:** Es una variable de tipo booleano y puede tener contener uno de dos posibles valores:
 - TRUE (equivalente a 1)
 - FALSE (equivalente a 0).

Asignación de variables

Asignación

Asignación de valor a una variable

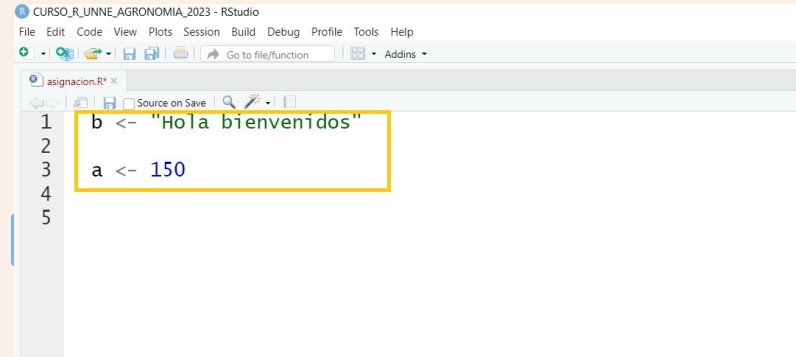
Formas de asignación:

`x <- 3`

`y = 3`

`3 -> z`

En R la forma más utilizada para asignar un valor a una variable es utilizando el operador `"<-"`.



Operadores

Operadores

Antes de comenzar a manipular y realizar operaciones con datos, es necesario conocer los distintos operadores que R utiliza para realizar cálculos y comparaciones.

- ✓ Operadores Aritméticos

- ✓ Operadores Lógicos

Operadores aritméticos

Operadores aritméticos

Operación	Símbolo	Ejemplo
Suma	+	$> 4 + 2 = 6$
Resta	-	$> 4 - 1 = 3$
Multiplicación	*	$> 4 * 2 = 8$
División	/	$> 4 / 2 = 2$
Potencial	^	$> 4 ^ 2 = 16$
Módulo	%%	$> 4 \% 2 = 0$

Operadores lógicos

Operadores lógicos

Son los que se utilizan típicamente para crear condiciones lógicas y comparar valores.

Operación	Símbolo	Ejemplo
Mayor que	>	$4 > 1 = \text{TRUE}$
Menor que	<	$4 < 1 = \text{FALSE}$
mayor o igual que	>=	$4 >= 1 = \text{TRUE}$
menor o igual que	<=	$4 <= 1 = \text{FALSE}$
exactamente igual a	==	$4 == 1 = \text{FALSE}$
distinto de	!=	$4 != 1 = \text{TRUE}$
cláusula “O” o “OR”		$\text{TRUE} \text{FALSE} = \text{TRUE}$
cláusula “Y” o “AND”	&	$\text{TRUE} \& \text{FALSE} = \text{FALSE}$

Función

Función

Nombre de la
función



Argumentos



```
name <- function(arg1,arg2, arg3, arg-n) {  
  sentencia1  
  sentencia2  
  .....  
  sentenciaN  
}
```

Operaciones/acciones

Ejemplo:

```
funcion_bienvenida <- function() {  
  print("Bienvenidos al curso de R")  
}
```

Función

Una función tiene tres componentes importantes:

- ✓ los **parámetros o argumentos**, que son los valores que recibe la función como entrada;
- ✓ el **código** de la función, compuesto por sentencias que son las operaciones que realiza la función;
- ✓ y el **resultado** (o valor de retorno), que es el valor final que entrega la función.

Paquetes

Un paquete está formado por un conjunto de funciones con un fin específico. Es posible escribir un paquete propio o utilizar un paquete ya existente. Hay más de 10.000 paquetes disponibles en CRAN (la red completa de archivos R).

- ✓ `install.packages("nombre_del_paquete")`: esta sentencia se utiliza para instalar un paquete.
- ✓ `library("nombre_del_paquete ")`: esta sentencia permite que el paquete esté disponible para su uso.

Conceptos vistos en clase

Asignación: La asignación sirve para pasarle a un objeto un valor , donde el misma puede ser un valor, vector, lista o dataframe.

Tipo de Dato: Hace referencia a si el dato es un valor de tipo carácter, numérico, lógico o factor. Para verificar el tipo de dato se utiliza la función `class()`.

Operadores Aritméticos: son utilizados para realizar operaciones numéricas sobre objetos del mismo tipo.

Operadores Lógicos: son utilizados para comparar dos o más valores booleanos (TRUE o FALSE). Típicamente se utilizan para evaluar una condición.

Función: Las funciones son fragmentos de código que definen una o varias acciones, que pueden usarse tantas veces como sea necesario, usado así para optimizar el código y evitar repetirlo.

Temas vistos en la clase 1

- ✓ Tipos de datos
- ✓ Asignación de variables
- ✓ Operadores
- ✓ Funciones
- ✓ Objetos
- ✓ Bonus: Proyectos

Temas de la clase 2

- ✓ Estructura de datos
- ✓ Vectores (Variables)
- ✓ Dataframes (bases de datos)
- ✓ Operaciones básicas



Clase 02. R

Operaciones básicas con R

Temario 02



- ✓ Estructura de datos
- ✓ Vectores
- ✓ Dataframes (bases de datos)
- ✓ Acceso a vectores y dataframes
- ✓ Bonus: Buenas prácticas para nombrar variables

Estructura de datos

Existen 3 estructuras de Datos



1. Vectores



2. Listas



3. Dataframes

Vectores

Es la estructura de datos más sencilla, la cual se caracteriza por ser unidimensional, es decir, tiene únicamente una dimensión. Adicionalmente, otra característica a destacar es que solo admite elementos del mismo tipo de dato(character, numeric, booleano, etc.). Es decir, no es posible definir vectores con múltiples tipos de datos.

La longitud de un vector está definido por la cantidad de elementos que conforman el mismo.

Creación de vectores

Para generar un vector, se utiliza la función de R base `c()`, que simplemente concatena valores. A continuación definimos un vector de caracteres y otro numérico respectivamente.

- ✓ Vector de caracteres:

```
a <- c("Santa fe", "Misiones", " Chaco", "Corrientes")
```

- ✓ Vector numérico:

```
b <- c(1,2,3,4,5,6,7,8,9,10)
```

Listas

Son probablemente la estructura de datos más compleja y flexible.

La principal característica de las listas es que pueden contener elementos de distintos tipos de datos: numeric, boolean, vectores, dataframes, etc.

La forma de crear una lista es utilizando la función `list()` de R base.

Ejemplo:

```
mi_primer_lista <- list(  
  a = c(1:15),  
  b = c("Maíz", "Soja", "Alfalfa"),  
  c = c(TRUE, FALSE, TRUE, FALSE, TRUE, FALSE),  
  d = clima)
```


Dataframes

Un **dataframe** o **df** es una tabla o un arreglo de dos dimensiones, en la cual cada **columna** corresponde a una variable y cada **fila** a las observaciones.

columna mes

↓

fila 3 →

	origen	anio	mes	dia	hora	temperatura	punto_rocio
	<chr>	<int>	<int>	<int>	<int>	<dbl>	<dbl>
1	EW	2013	1	1	1	39.0	26.1
2	EW	2013	1	1	2	39.0	27.0
3	EW	2013	1	1	3	39.0	28.0
4	EW	2013	1	1	4	39.9	28.0
5	EW	2013	1	1	5	39.0	28.0
6	EW	2013	1	1	6	37.9	28.0
7	EW	2013	1	1	7	39.0	28.0
8	EW	2013	1	1	8	39.9	28.0
9	EW	2013	1	1	9	39.9	28.0
10	EW	2013	1	1	10	41	28.0

Creación de un dataframe

Para generar un vector, se utiliza la función de R base `data.frame()`, que simplemente concatena valores. A continuación definimos un dataframe.

✓ Ejemplo:

```
muestra_citricos <- data.frame(Anio= c(2018, 2018, 2019, 2020, 2021, 2022),  
                               Campania = c(1, 2, 2, 3, 4, 3),  
                               Especie = c("Naranja", "Mandarino", "Naranja", "Mandarino", "Mandarino", ),  
                               Suelo = c("arcilloso", "arenoso", "arenoso", "arcilloso", "arcilloso", "arenoso"),  
                               Riego = c("si", "no", "si", "no", "no", "si"),  
                               Edad = c(10, 7, 13, 20, 9, 16))
```

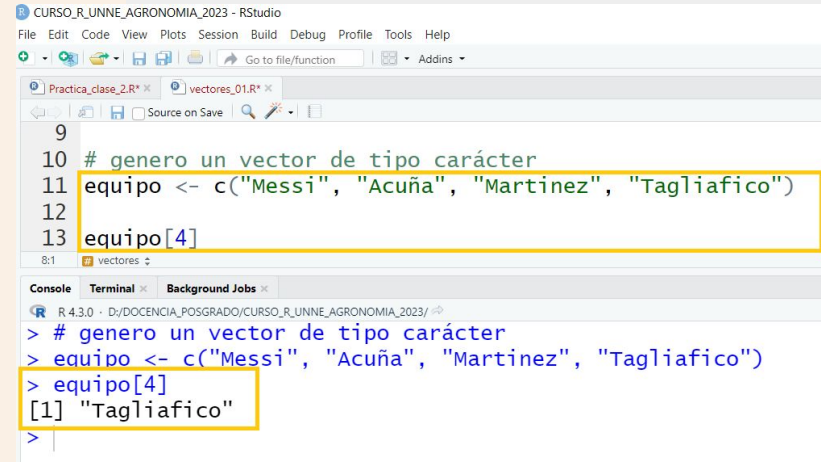
Manipulación de vectores y dataframes

Acceso a **vectores**

Selección de un elemento

Para **seleccionar un elemento** de un vector, **se usan corchetes []** dentro de los cuales se señala el **número o posición del elemento** al cual se desea acceder.

Por ejemplo, para seleccionar el cuarto elemento del vector “equipo” antes creado, solamente deberemos nombrar al vector, seguido por los corchetes conteniendo la posición correspondiente: **equipo[4]**



The screenshot shows the RStudio interface. The script editor on the left contains the following R code:

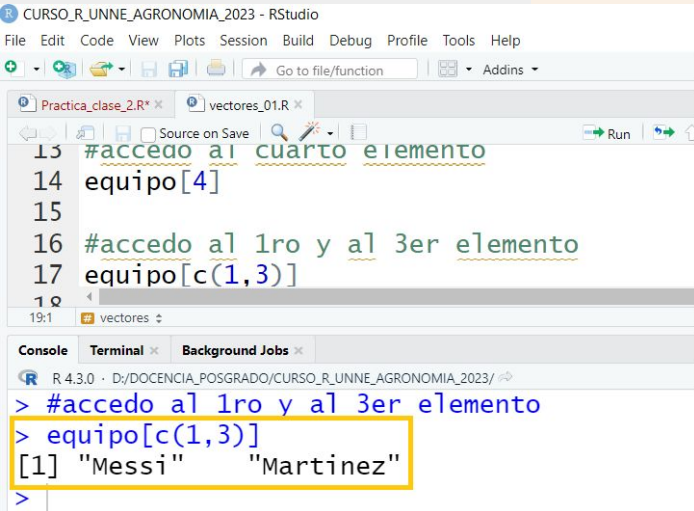
```
9  
10 # genero un vector de tipo carácter  
11 equipo <- c("Messi", "Acuña", "Martinez", "Tagliafico")  
12  
13 equipo[4]
```

The console on the right shows the execution of the code:

```
R 4.3.0 · D:/DOCENCIA_POSGRADO/CURSO_R_UNNE_AGRONOMIA_2023/  
> # genero un vector de tipo carácter  
> equipo <- c("Messi", "Acuña", "Martinez", "Tagliafico")  
> equipo[4]  
[1] "Tagliafico"  
>
```

Yellow boxes highlight the code for creating the vector and the selection of the fourth element, as well as the corresponding output in the console.

Selección de varios elementos



```
CURSO_R_UNNE_AGRONOMIA_2023 - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Practica_clase_2.R* x vectores_01.R x
13 #accedo al cuarto elemento
14 equipo[4]
15
16 #accedo al 1ro y al 3er elemento
17 equipo[c(1,3)]
18
19:1 vectores
```

```
R 4.3.0 · D:/DOCENCIA_POSGRADO/CURSO_R_UNNE_AGRONOMIA_2023/
> #accedo al 1ro y al 3er elemento
> equipo[c(1,3)]
[1] "Messi" "Martinez"
```

Para **seleccionar varios elementos a la vez**, se utilizarán los **corchetes []** más la función **concatenar c()** y las **posiciones de los elementos que se desee localizar**.

Por ejemplo, para seleccionar el tercer y quinto elemento del vector “equipo”, deberemos nombrar al vector, seguido por la función concatenar entre los corchetes: **equipo[c(1,3)]**

Más ejemplos

Dado el siguiente vector: `equipo <- c("Messi", "Acuña", "Martinez", "Tagliafico")`

- ✓ Con `equipo[1]` podemos obtener el primer elemento.
- ✓ Con la función `c()` podemos seleccionar múltiples elementos a la vez.
- ✓ Con la siguiente sentencia `equipo[c(1,34)]` se obtiene el primer y último elemento del vector.

Esta manera de acceder a los elementos del vector se realiza por la posición de cada elemento dentro del vector. Es importante saber que en R las posiciones inician en 1.

Excluir elementos de la selección

Si en lugar de seleccionar quisiéramos quitar algunos elementos es posible hacerlo **agregando el signo negativo delante del elemento o de `c()` si se desea quitar más de un elemento elemento.**

Ejemplo:

- ✓ `equipo[-1]` para quitar el primer elemento.
- ✓ `equipo[-c(2,3)]` para quitar el 2do y 3er elemento.



Break

¡20 minutos y volvemos!



Por favor, trata de alejarte de las pantallas.

Manipulación de dataframes

Manipulación de dataframes

Vamos a utilizar el dataset "clima" del paquete *datos* que ya hemos instalado.

Con los corchetes `[]`, al igual que con vectores, se puede acceder a las filas y columnas de un dataframe.

El primer argumento antes de la coma hace referencia a las filas y el segundo argumento hace referencia a las columnas:

`df[fila, columna]`.

Acceso a **dataframes**

Selección de un elemento

Para **seleccionar un elemento** de un dataframe, **se usan corchetes []** dentro de los cuales se señala el número de fila y **columna del elemento** al cual se desea acceder.

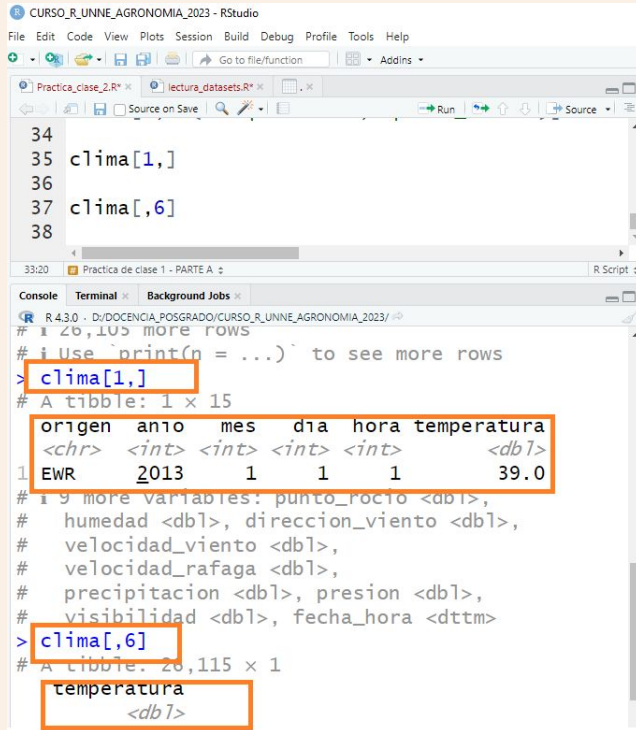
Por ejemplo, para acceder al elemento ubicado en la fila 10, columna 6, deberemos nombrar al df, seguido por los corchetes conteniendo la posición correspondiente. Ejemplo: **clima[10, 6]** accederá al valor **41**, correspondiente a la observación nro. 10 de la columna temperatura.

columna
temperatura
(posición 6)

	origen	año	mes	día	hora	temperatura	punto_rocio
	<chr>	<int>	<int>	<int>	<int>	<dbl>	<dbl>
1	EWB	2013	1	1	1	39.0	26.1
2	EWB	2013	1	1	2	39.0	27.0
3	EWB	2013	1	1	3	39.0	28.0
4	EWB	2013	1	1	4	39.9	28.0
5	EWB	2013	1	1	5	39.0	28.0
6	EWB	2013	1	1	6	37.9	28.0
7	EWB	2013	1	1	7	39.0	28.0
8	EWB	2013	1	1	8	39.9	28.0
9	EWB	2013	1	1	9	39.9	28.0
10	EWB	2013	1	1	10	41	28.0

fila 10

Selección por posición



The screenshot shows the RStudio interface. The script editor contains the following code:

```
34
35 clima[1,]
36
37 clima[,6]
38
```

The console shows the output of the first command, `clima[1,]`, which is a tibble with 1 row and 15 columns. The first row of data is highlighted:

	origen	año	mes	día	hora	temperatura
1	EWR	2013	1	1	1	39.0

The console also shows the output of the second command, `clima[,6]`, which is a tibble with 26 rows and 1 column, named "temperatura".

Para extraer Filas

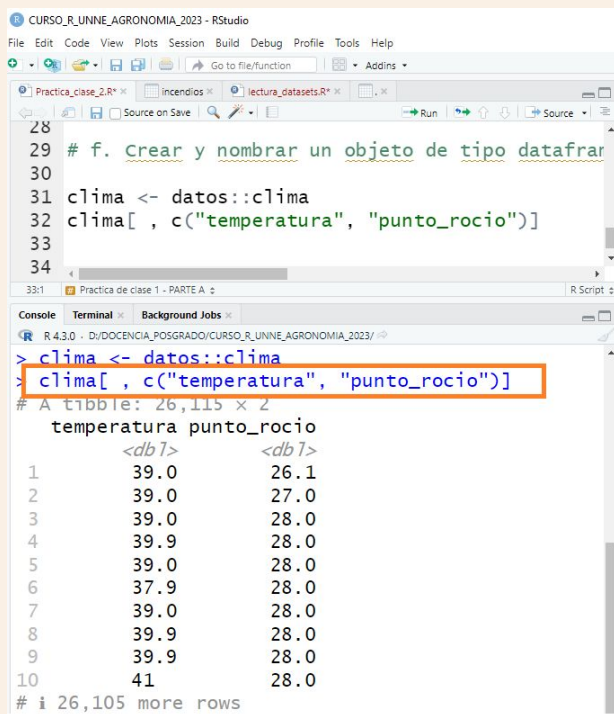
> `clima[1,]` Obtengo la primera fila

> `clima[c(1,2),]` Obtengo múltiples filas (la 1 y 2)

Para extraer Columnas

> `clima[,6]` Obtengo todos los valores de la segunda columna "temperatura"

> `clima[,c(1,6)]` Obtengo todos los valores de la primera y segunda columna.



The screenshot shows the RStudio interface. The script editor contains the following code:

```
28  
29 # f. Crear y nombrar un objeto de tipo datafram  
30  
31 clima <- datos::clima  
32 clima[, c("temperatura", "punto_rocio")]  
33  
34
```

The console shows the execution of the code:

```
> clima <- datos::clima  
> clima[, c("temperatura", "punto_rocio")]  
# A tibble: 26,115 x 2  
  temperatura punto_rocio  
    <dbl>         <dbl>  
1      39.0         26.1  
2      39.0         27.0  
3      39.0         28.0  
4      39.9         28.0  
5      39.0         28.0  
6      37.9         28.0  
7      39.0         28.0  
8      39.9         28.0  
9      39.9         28.0  
10     41          28.0  
# i 26,105 more rows
```

Selección por nombre

También es posible extraer las columnas que se requieren, utilizando su nombre en lugar de utilizar su posición numérica.

```
> clima[, c("temperatura", "punto_rocio")]
```

¿Preguntas?

Práctica nro. 1 - Parte A

- A.** Crear un OBJETO llamado ``COSA`` definido como el resultado de la multiplicación: $7*3$.
- B.** Crear un **vector** llamado ``plantas`` que contenga los números 105, 223, 1580 y 560.
- C.** Crear un **vector** llamado ``especies`` que contenga los valores: trébol rojo, stylosanthes, alfalfa y soja.
- D.** Crear un **vector** con la misma cantidad de valores que ``especies``, de tipo texto (character) .
- E.** Crear y nombrar un objeto de tipo **dataframe** que sea la combinación de todos los vectores creados previamente

Práctica nro. 1 - Parte B

G. Crear un objeto llamado *OTRA_COSA* que guarde el valor de la segunda fila y tercera columna del dataframe creado en el punto f.

H. Crear un vector con el siguiente contenido:

```
c("1", "dos", 3, "CUATRO", 5, NA)
```

I. Calcular la media de ``plantas`` para el dataframe creado.

J. Verificar de qué tipo es el vector recientemente creado en h.

K. Sobre el dataframe creado, ejecutar las siguientes funciones y describir brevemente qué hace cada una:

- `dim(aca_va_la_base_de_datos)`
- `names(aca_va_la_base_de_datos)`
- `summary(aca_va_la_base_de_datos)`
- `length(aca_va_la_base_de_datos)`
- `str(aca_va_la_base_de_datos)`

Buenas prácticas para nombrar objetos

Reglas

X No se aceptan espacios en blanco

Ejemplo: ` nombre y apellido`

X No se puede empezar con un número

Ejemplo: ` 4_objeto`

👎 Evitar símbolos (&, ^, /, !) y acentos

👎 Evitar que objetos y funciones tengan el mismo nombre

Convenciones

todoenminuscula

🔥 TODOENMAYUSCULA

100 espacio_con_guion_bajo

Clase 2

Conceptos vistos en clase

Vector: estructura de datos unidimensional, no admite elementos de diferente tipo.

Lista: estructura de datos unidimensional más flexible que el vector, acepta elementos de distintos tipos de datos.

Dataframe: estructura de datos de dos dimensiones formado por filas y columnas.

Funciones de R Base:

dim(): se utiliza para conocer la dimensión de un dataframe.

names(): se utiliza para conocer los nombres de un objeto(vector, df, entre otros).

length(): se utiliza para conocer la longitud de un vector.

summary(): se utiliza para obtener un resumen general sobre las variables de un dataframe.

str(): se utiliza para conocer la estructura de un objeto (vector, df, entre otros).

Bibliografía

- ✓ R para Ciencia de datos por by Hadley Wickham y Garrett Grolemond. | [Enlace](#)
- ✓ R para Análisis Científicos Reproducibles por The Carpentries | [Enlace](#)

Práctica adicional vectores y dataframes

1. Crear un vector llamado *cultivos* que contenga 6 valores
2. Extraer el segundo valor del vector creado y asignarlo a un nuevo objeto
3. Dados los siguientes vectores:
4. Crear un objeto de tipo `data.frame` (base de datos) que contenga el vector creado (*cultivos*) más los dos propuestos (localidad y tratamiento)
5. Extraer del `data.frame` el valor de la tercer fila y segunda columna.
6. 3. Consultar del `data.frame` sobre los valores de la columna `tipo_alojamiento`