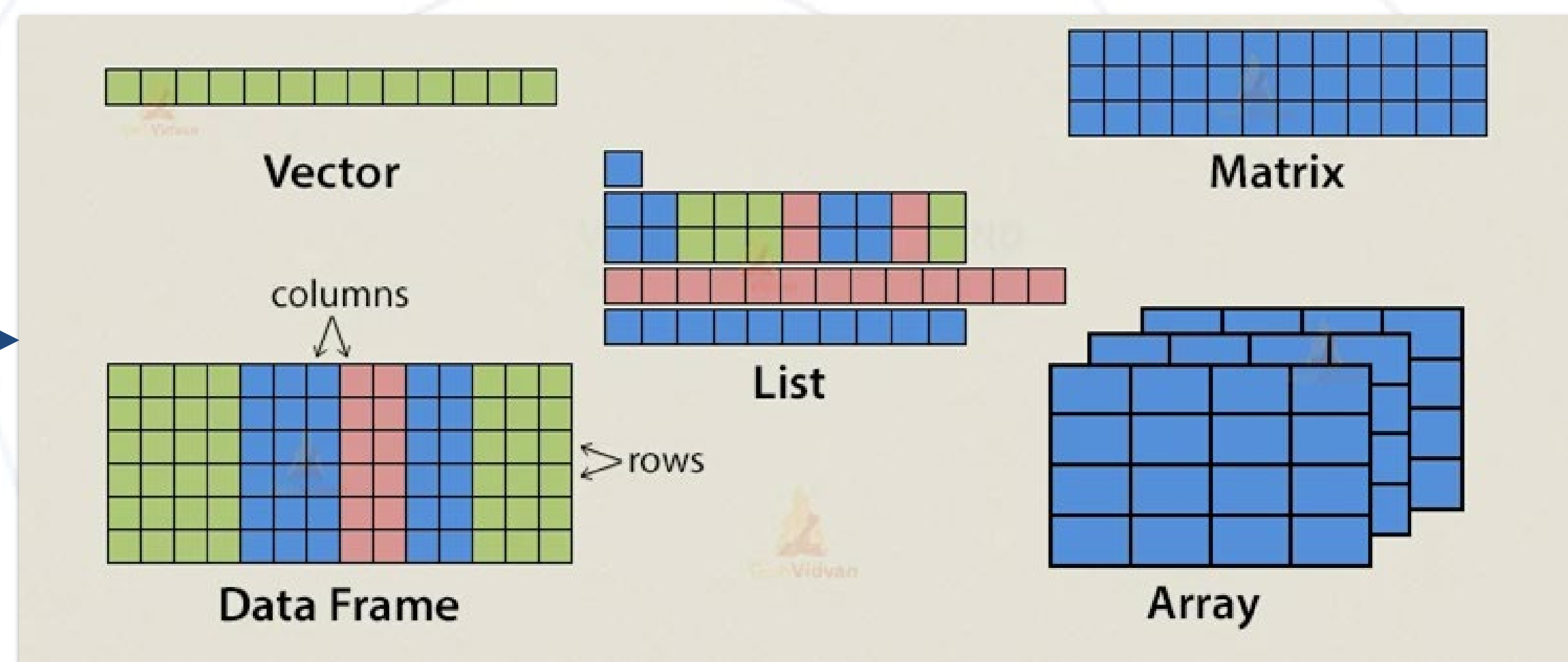


R para Ciencias de la Salud

Unidad 5 <- Estructura de Datos ("Listas", "*Data Frames*")

Rafael J. Puche Q. & Fernando Hernández
Centro de Medicina Experimental
Instituto Venezolano de Investigaciones Científicas (IVIC)

Estructura de Datos



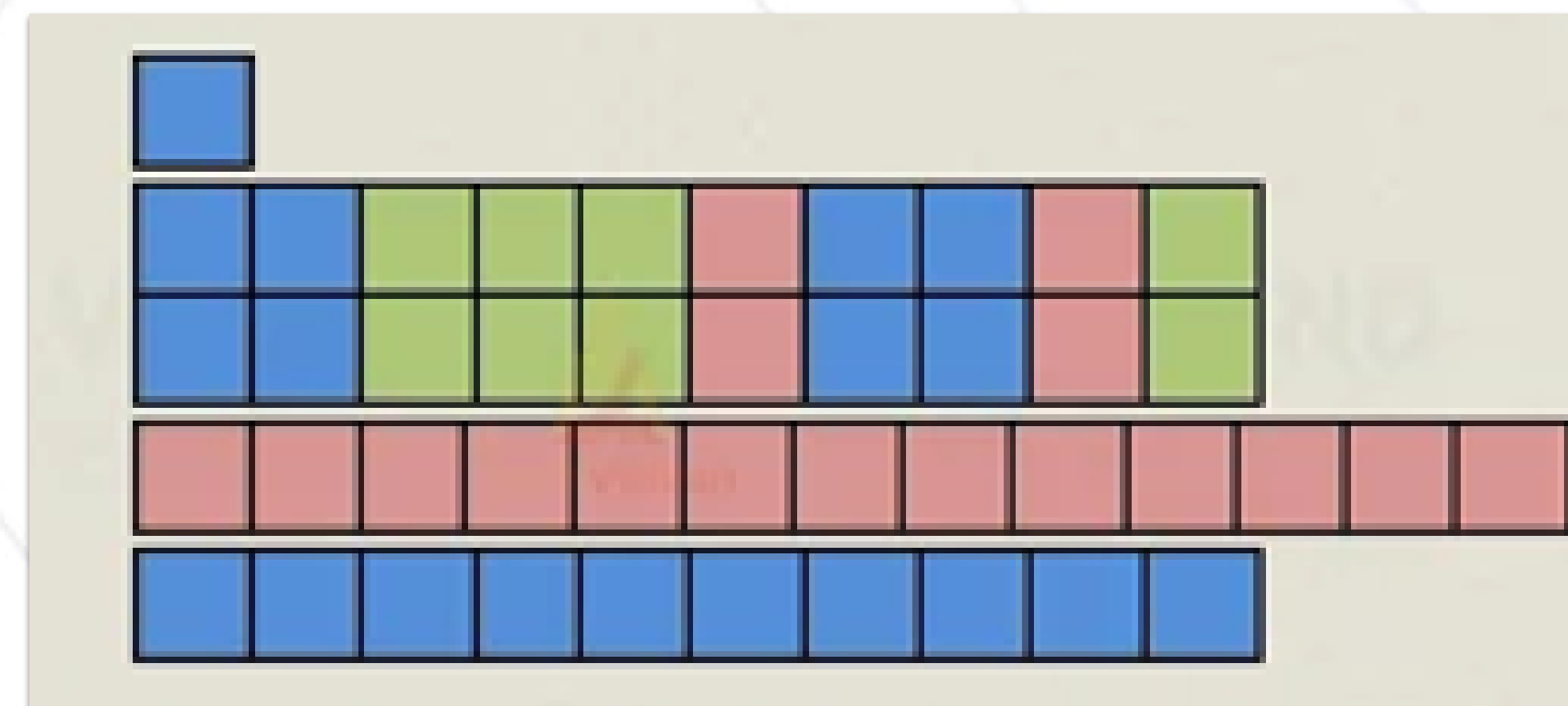
Dimensión	Homogéneo	Heterogéneo
1	Vector	Lista
2	Matriz	Data frame
N	Array	

¿Que es una lista?

Un **contenedor de objetos** que pueden ser de cualquier clase: **números, vectores, matrices, funciones, data frames**, incluso otras listas.

Puede contener **varios objetos a la vez**, que pueden ser además de **distintas dimensiones**.

Cada elemento de una lista tiene un nombre (o índice) asociado, que puede ser numérico o una cadena de caracteres.



Una lista es unidimensional, sólo tiene largo.

Estructura Basica de una lista

Se pueden crear listas utilizando la función `list()`

```
mi_lista <- list(1, "a", TRUE, c(2, 3))
```

Ejemplo:

Los elementos de la lista pueden ser nombrados durante la creación:

```
mi_lista <- list(num = 1, char = "a",  
                logico = TRUE, vec = c(2, 3))
```

Operadores para lista

1. $[i]$
2. $[[i]]$
3. $x\$y$

Sintaxis	Objetos	Descripción
$x[i]$	Vectores, Listas	Selecciona elementos del objeto x , descritos en i . i puede ser un vector de tipo integer, character (de nombres de los objetos) o lógico. Cuando es usado con listas, devuelve una lista. Cuando es usado en vectores devuelve un vector.
$x[[i]]$	Listas	Devuelve un único elemento de x que se encuentra en la posición i . i puede ser un vector de tipo integer o character de longitud 1.
$x\$n$	Listas, Dataframes	Devuelve un objeto con nombre n del objeto x .
$[i, j]$	Matrices	Devuelve el objeto de la fila i y columna j . i y j pueden ser un vector de tipo integer o character (de nombres de los objetos)

Familia `apply()`

Colección de funciones muy potentes que permiten **aplicar operaciones a estructuras de datos** como vectores, listas, *data frames* y *arrays*

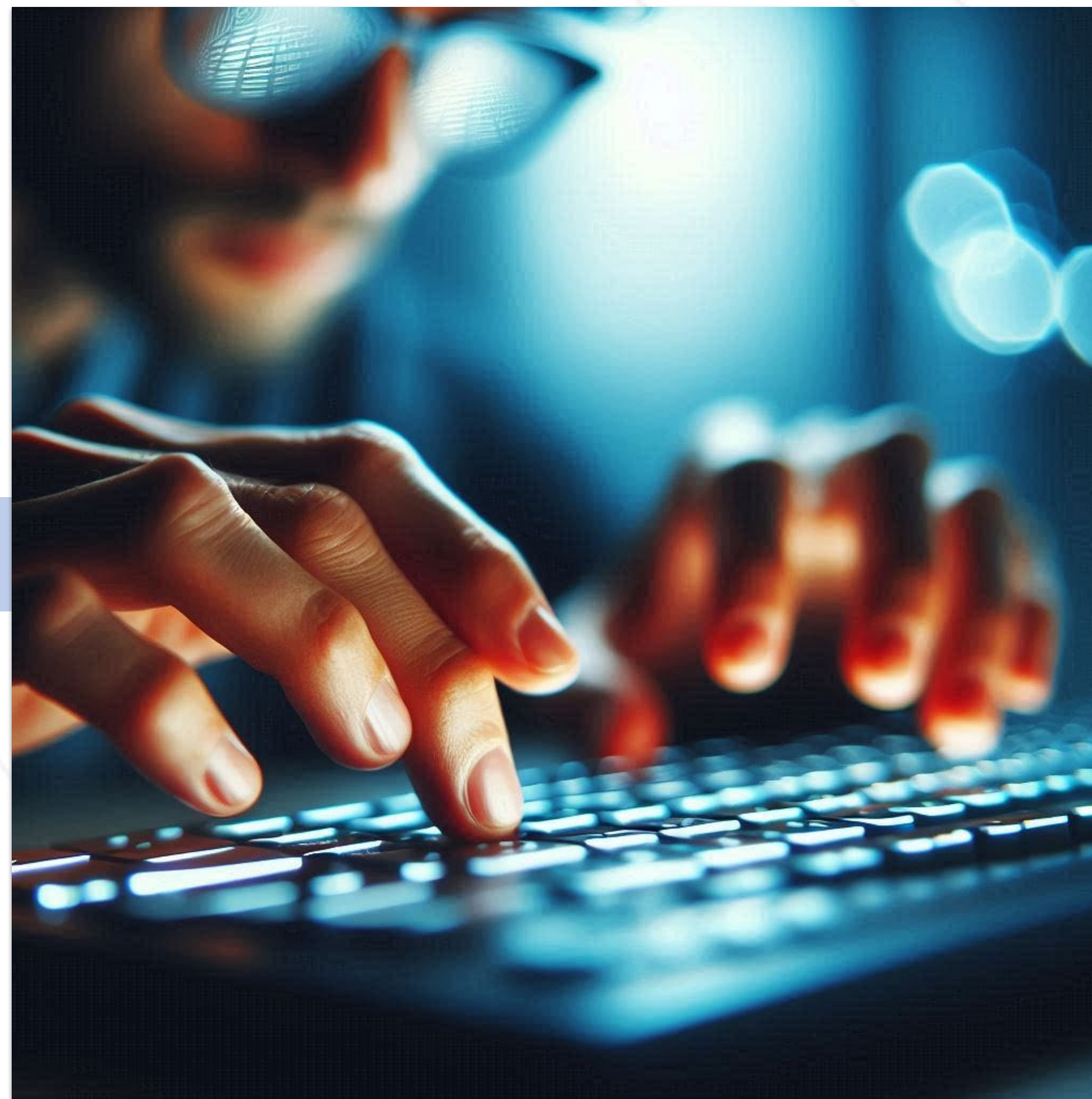
- **Sintaxis:** `apply(X, MARGIN, FUN, ...)`
 - `MARGIN = 1` aplica la función por **filas**,
 - `MARGIN = 2` por **columnas**.
- **Ejemplo:**
 - `apply(matriz, 1, sum)` suma por filas.

Familia `apply()`

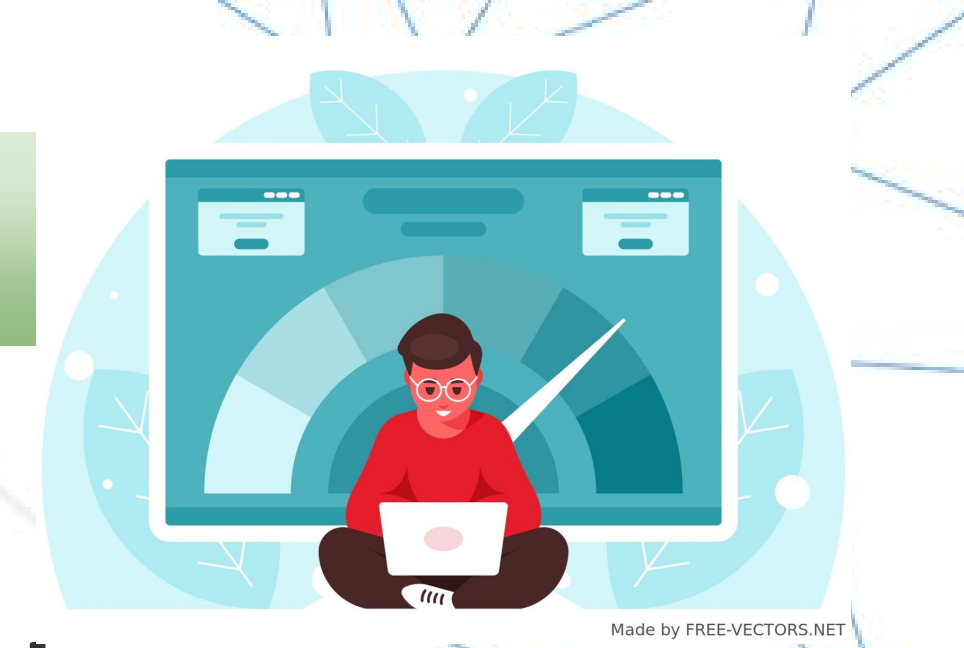
Gene	Control	Tratamiento1	Tratamiento2
Gene1	10	12	15
Gene2	5	8	6
Gene3	20	25	22
Gene4	0	1	0
Gene5	50	55	60

Función	Código_R	Output	Explicación
<code>apply()</code>	<code>apply(exp_mat, 1, mean)</code>	Gene1=12.3, Gene2=6.3, ...	Media por gen
<code>apply()</code>	<code>apply(exp_mat, 2, sd)</code>	Control=19.3, Trat1=20.8, ...	Desviación estándar por muestra
<code>lapply()</code>	<code>lapply(as.data.frame(exp_mat), max)</code>	Control=50, Trat1=55, Trat2=60	Máximo por muestra
<code>sapply()</code>	<code>sapply(as.data.frame(exp_mat), min)</code>	Control=0, Trat1=1, Trat2=0	Mínimo por muestra
<code>mapply()</code>	<code>mapply(mean, exp_mat[,1], exp_mat[,2])</code>	10.5, 6.5, 22.5, 0.5, 52.5	Media entre Control y Tratamiento1 por gen

Manos al teclado



Ejercicio - LISTA



1. Crea una lista llamada `mi_lista` que contenga los siguientes elementos:

- a. Un vector numérico con los números del 1 al 5
- b. Una cadena de caracteres con tu nombre
- c. Un vector lógico con los valores TRUE, FALSE, TRUE
- d. Una matriz de 2x2 con los números del 1 al 4

2. Imprime la lista `mi_lista` en la consola para ver su estructura, además imprime el tipo de elemento `mi_lista`

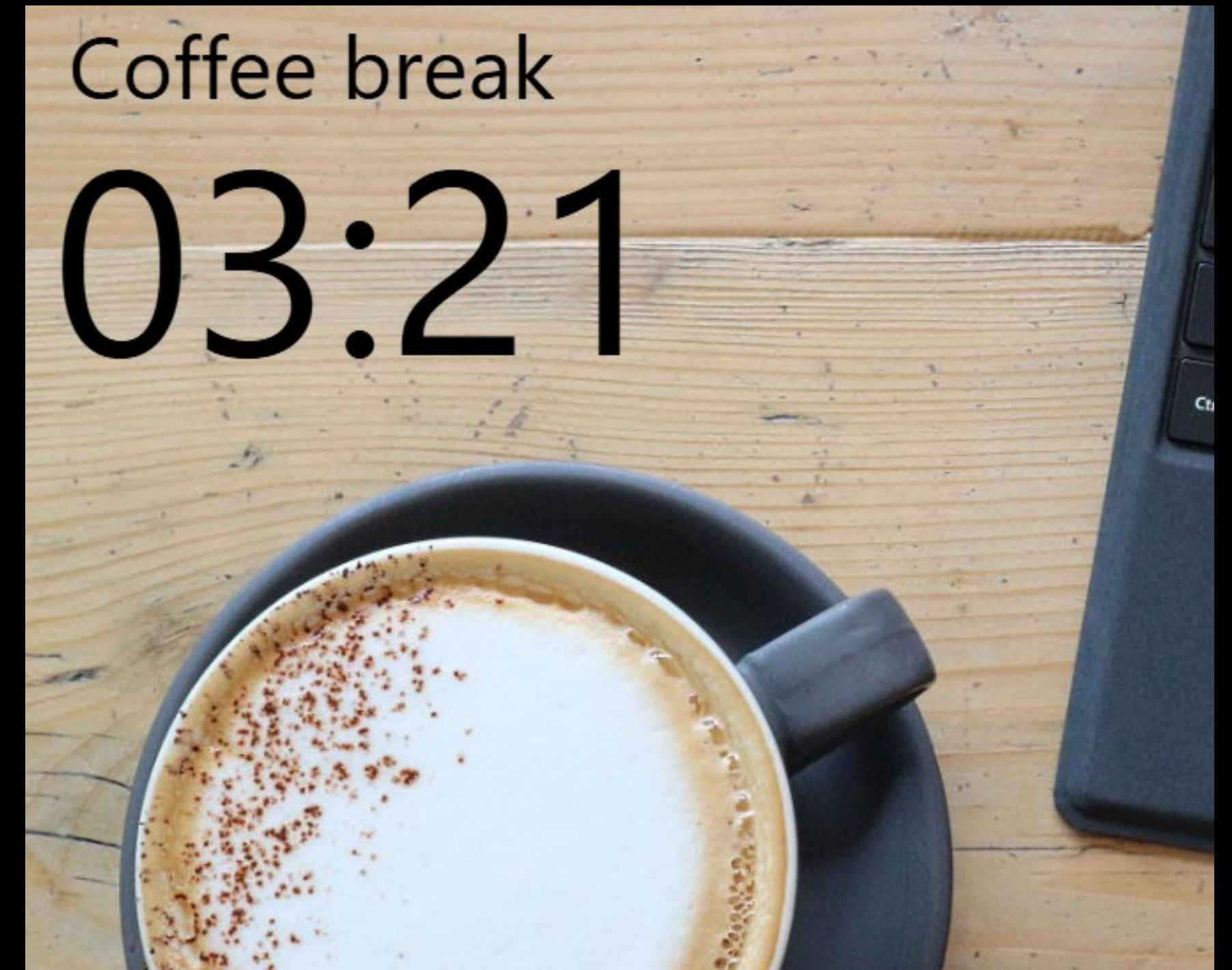
3. Accede al segundo elemento de la lista (tu nombre) y almacénalo en una variable llamada `mi_nombre`.

Me after coding for 5 minutes..



Coffee break

03:21



Es un conjunto de paquetes en R diseñado para facilitar el análisis de datos de forma coherente, legible y eficiente. Todos los paquetes del tidyverse comparten una misma filosofía: trabajar con datos en formato "ordenado" (*tidy data*)

N°	APellidos y Nombres	SEXO	PRUEBA INSTRUMENTAL	PRUEBA INSTRUMENTAL	PARTIAL	PARCIAL O PARCIAL	PRÁCTICA 1	PRÁCTICA 2	PRÁCTICA 3	PRUEBA O PRÁCTICA	NOTA PARCIAL	EXAMEN FINAL	PONDERAD O EXAMEN	FINAL	CONDICION
1	CONTORI MAMANT DANIELA	MUJER	45	50	50	48	45	35	25	35					APROBADO
2	CHINO GONZALEZ EMANUEL	VARON	40	45	35	40	40	40	26	35					APROBADO
3	HERRERA CALANT ELIANA	MUJER	36	40	35	37	35	40	35	37					APROBADO
4	HERRERA FERNANDEZ VERONICA	MUJER	36	45	40	40	36	45	48	43					APROBADO
5	JALDIN TASTACA EDSON	VARON	51	56	45	51	42	30	60	44					APROBADO
6	LOPEZ LINA GLADIS	MUJER	52	55	46	51	60	25	51	45					APROBADO
7	MAMANT MIRANDA WILFREDO	VARON	45	48	36	43	6	56	52	38					APROBADO
8	MARQUEZ ROSAS FRANKLIN	VARON	40	48	25	38	60	59	55	58					APROBADO
9	RODRIGUES CRUZ PABLO	VARON	25	40	35	33	45	56	54	52					REPROBADO
10	TECINA ORELLANA ALEX	VARON	16	36	36	29	25	42	56	41					
11	APAJA AJACOPA BERLIZ	MUJER	45	50	42	46	63	51	23	46					
12	CARO LIEDEZMA DANIELA	MUJER	40	50	40	43	25	50	52	42					
13	REINAGA LOPEZ SHIRLEY	MUJER	36	40	45	40	42	48	48	46					
14	SIQUE GARCERAN LUCIA	MUJER	40	45	40	42	56	45	45	49					
15	SANTOS PANTAGUA ISRAEL	VARON	41	45	45	44	50	12	12	25					

Tidyverse

Packages Blog Learn Help Contribute



R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

Lo veremos en el tema Analisis de datos en R (semana 8)

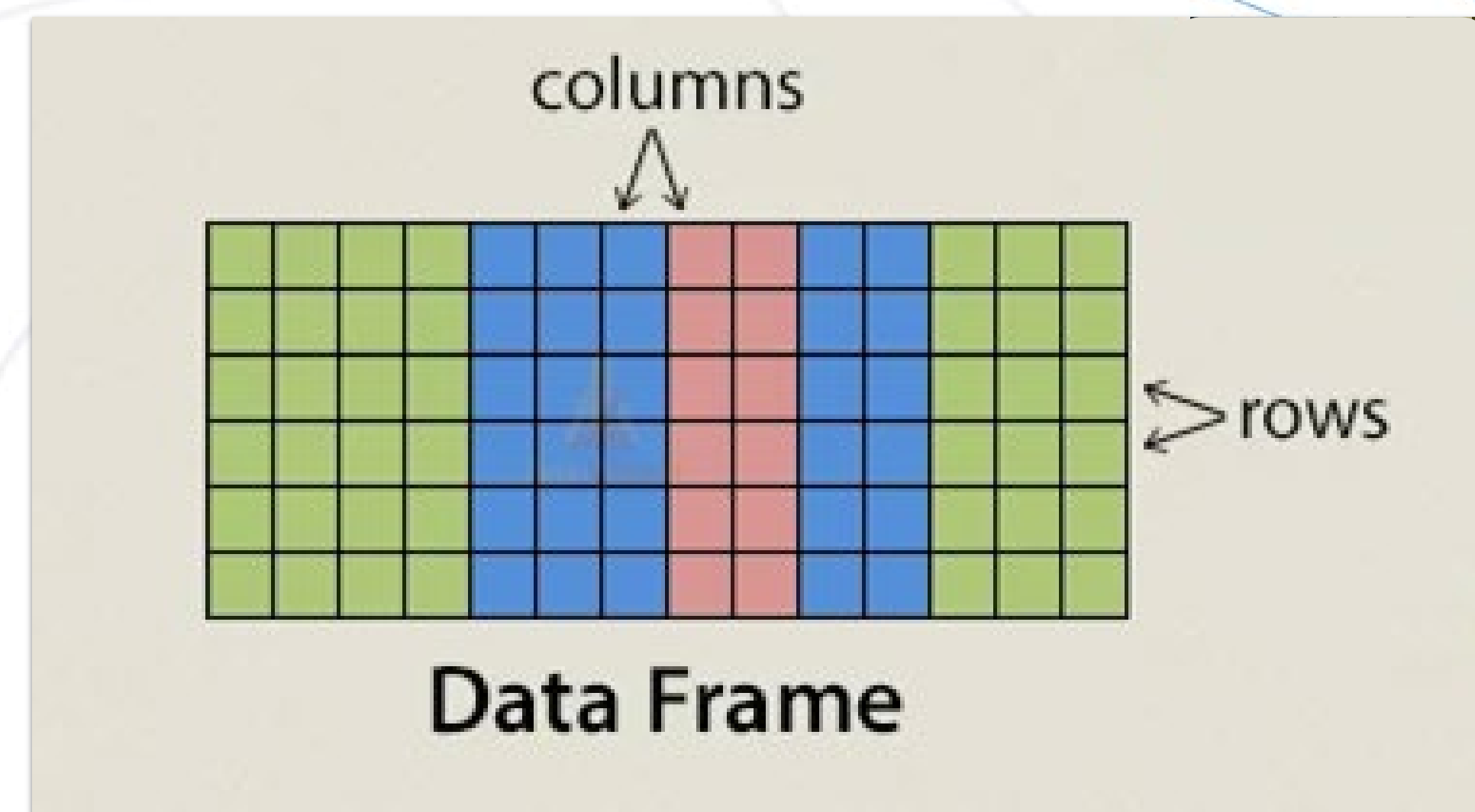
Que es un *Data Frame* (Marco de Datos)

Es una **estructura de datos bidimensional** en R, similar a una hoja de cálculo o una tabla.

Está compuesto por **filas y columnas**, donde cada columna debe contener el mismo tipo de datos (numérico, carácter, lógico, etc.).

Columnas se denominan “variables”
Filas representan las “observaciones”

Similar a una matriz, pero puede tener columnas con diferentes tipos de datos (numérico, carácter, lógico).



country	year	cases	population
Afghanistan	1999	1815	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	30737	17206362
Brazil	2000	80488	17404898
China	1999	210258	1272015272
China	2000	210766	128008583

variables

country	year	cases	population
Afghanistan	1999	1815	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	30737	17206362
Brazil	2000	80488	17404898
China	1999	210258	1272015272
China	2000	210766	128008583

observations

country	year	cases	population
Afghanistan	1999	1815	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	30737	17206362
Brazil	2000	80488	17404898
China	1999	210258	1272015272
China	2000	210766	128008583

values

Estructura Basica de un *Data Frame*

Se pueden crear data frames utilizando la función `data.frame()`

```
data.frame(col1 = vector1, col2 = vector2, ...)
```

Los DF pueden ser importados desde archivos externos, como CSV, Excel, bases de datos, entre otros (veremos en la Sesión 8, Análisis de datos en R)

Ejemplo:

```
df <- data.frame(nombre = c("Ana", "Juan", "Pedro"),  
                  edad = c(25, 32, 18),  
                  genero = c("F", "M", "M"))
```

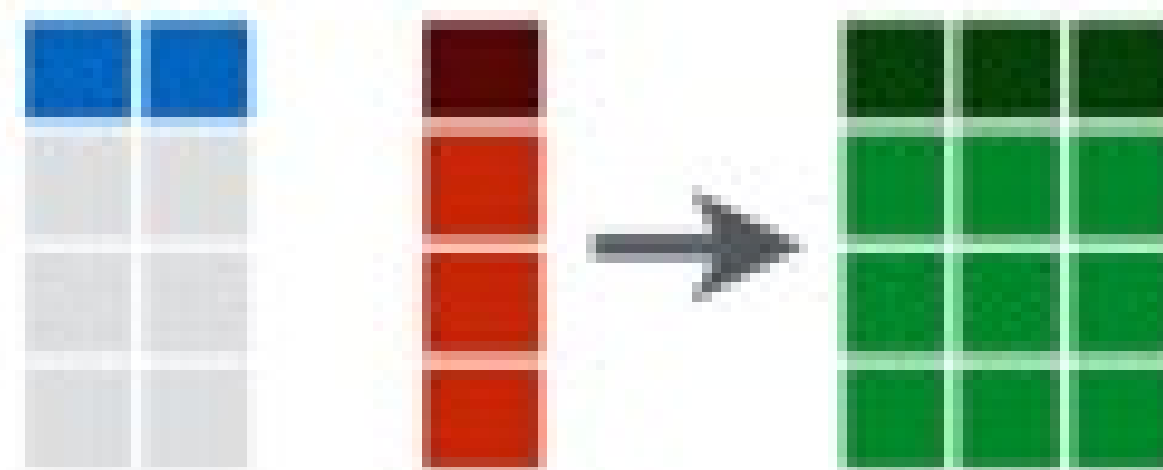
`cbind` y `rbind` en *Data Frame*

La función **`cbind()`** se utiliza para combinar dataframes o vectores por **columnas**.

Ejemplo:

```
cbind(df, altura = c(170, 180, 150))
```

`cbind` - Bind columns.



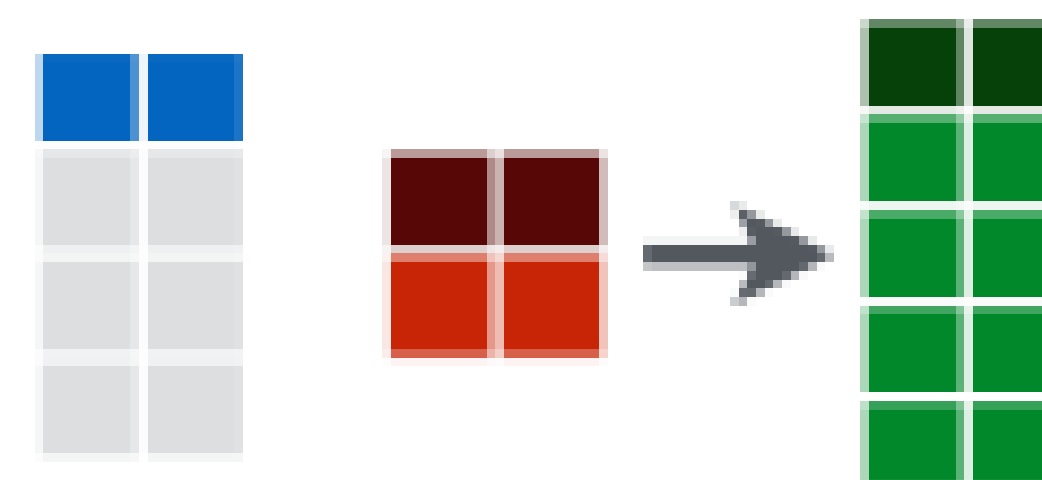
`cbind` y `rbind` en *Data Frame*

La función `rbind()` se utiliza para combinar dataframes o vectores por **filas**.

Ejemplo:

```
rbind(df, pais = c("Jamaica", "Italia",  
"Venezuela"))
```

`rbind` - Bind rows.



NOTA: al combinar dataframes con `cbind()` o `rbind()`, las columnas deben tener **nombres y tipos de datos compatibles**. Si los nombres de las columnas no coinciden, *R* asignará nombres generados automáticamente.

Estas funciones son muy útiles cuando necesitas agregar nuevas columnas o filas a un dataframe existente, o **combinar varios *data frames* en uno solo**.

Subconjuntos en *Data Frame*



	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4			X							
5						X	X			
6		X						X		X
7				X						X
8	X	X						X		
9										
10										

`df[, 2]`

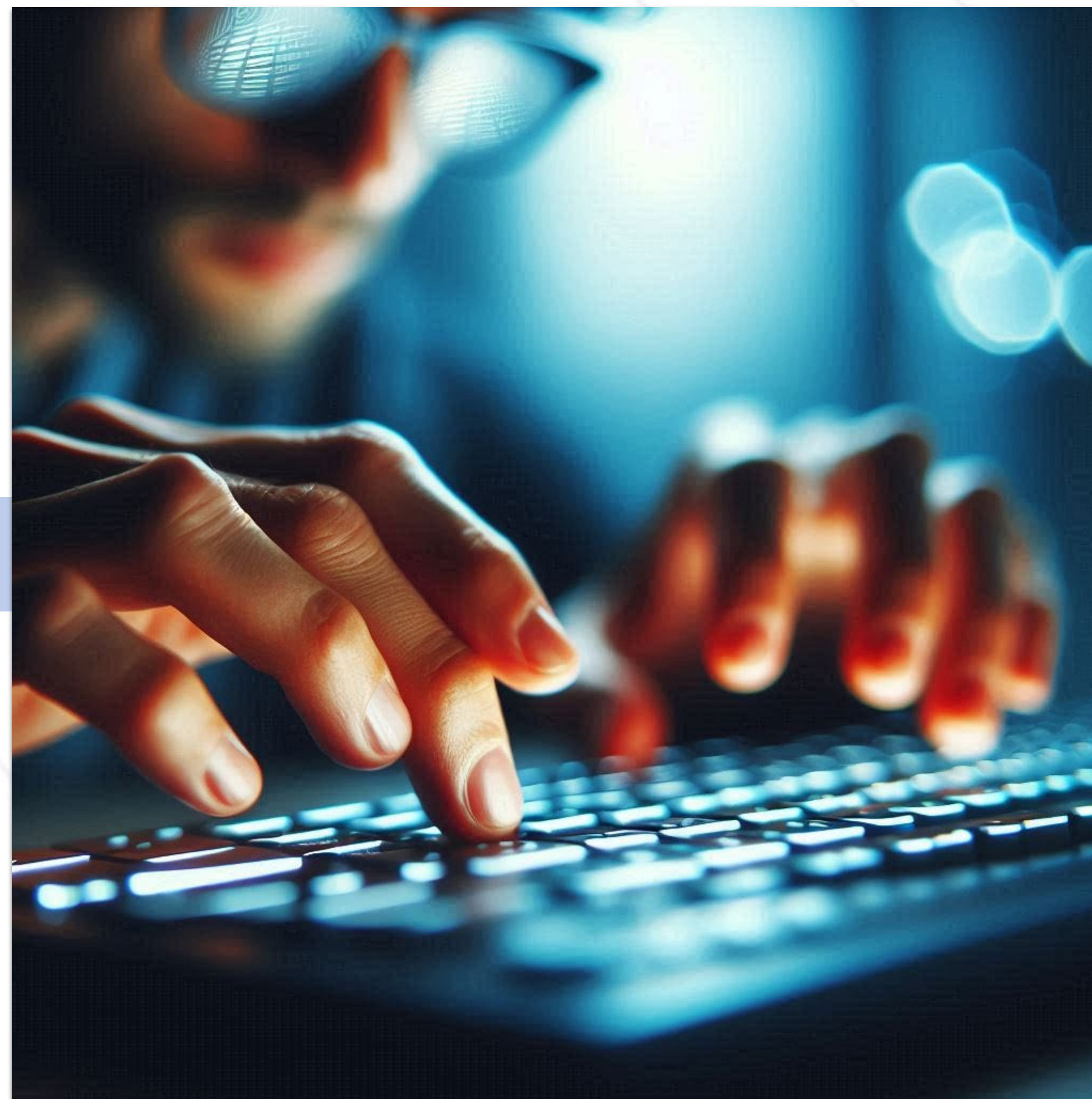
`df[2,]`

`df[2, 2]`

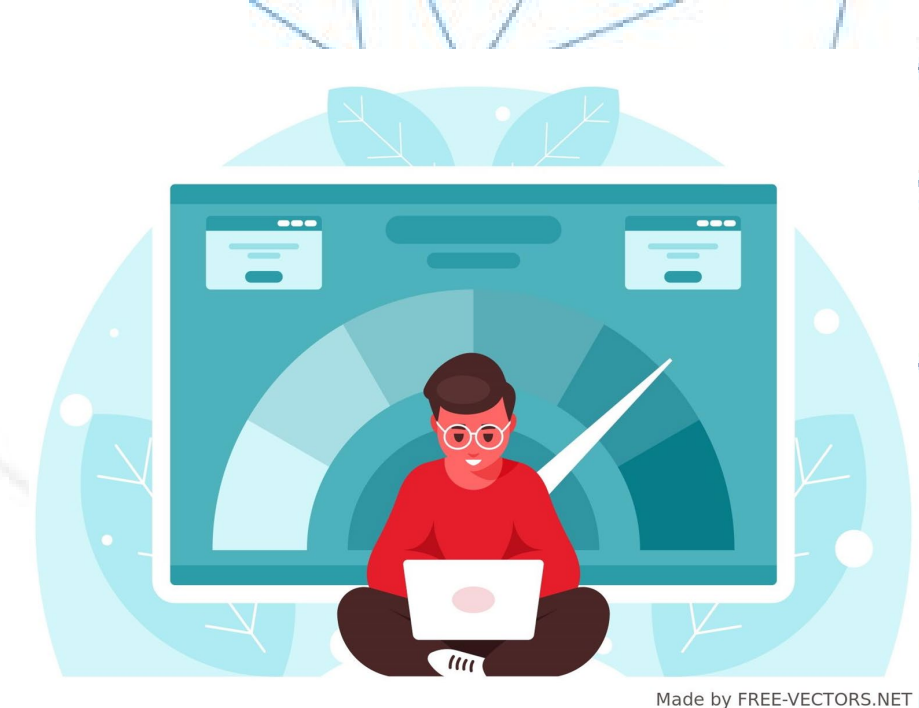
Subconjuntos en *Data Frame*

Operador	Descripción	Ejemplo
[]	Seleccionar filas y columnas específicas de un dataframe.	<code>datos[1:3, c("nombre", "edad")]</code> selecciona las filas 1 a 3 y las columnas "nombre" y "edad".
\$	Seleccionar una columna específica de un dataframe por su nombre.	<code>datos\$nombre</code> selecciona la columna "nombre".
==	Seleccionar filas donde los valores de una columna son iguales a un valor específico.	<code>datos[datos\$edad == 30,]</code> selecciona las filas donde la edad es igual a 30.
!=	Seleccionar filas donde los valores de una columna son diferentes a un valor específico.	<code>datos[datos\$genero != "M",]</code> selecciona las filas donde el género no es "M".
>, <, >=, <=	Seleccionar filas donde los valores de una columna cumplen una determinada condición numérica.	<code>datos[datos\$edad > 30,]</code> selecciona las filas donde la edad es mayor que 30.

Manos al teclado


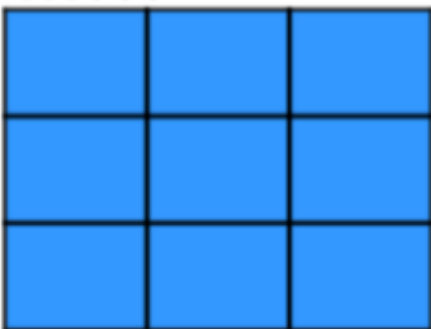
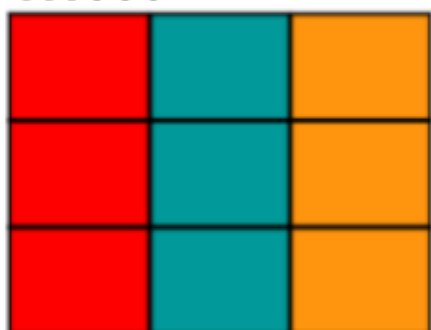
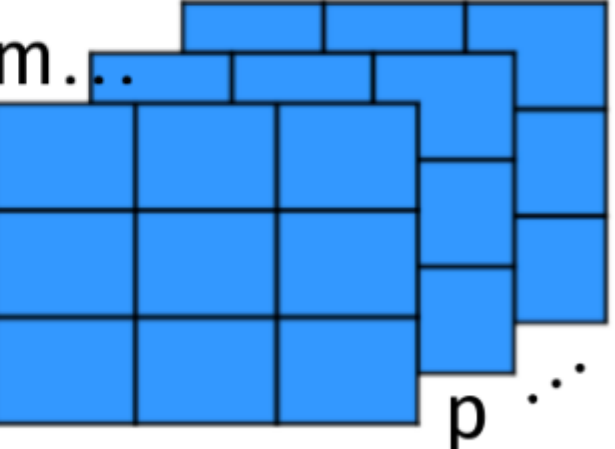


Ejercicio - DATA FRAME



1. Crea un **dataframe** llamado **pacientes** con las siguientes columnas: **nombre**, **edad**, **genero**, **diagnostico** y **tratamiento**. Ingresa los datos de al menos 5 pacientes diferentes.
2. Agrega una nueva columna llamada **fecha_ingreso** al dataframe **pacientes** utilizando la función **cbind**.
3. Crea un nuevo dataframe llamado **nuevos_pacientes** con las columnas **nombre**, **edad**, **género** y **diagnostico**. Ingresa los datos de al menos dos pacientes nuevos.
4. Combina los dataframes **pacientes** y **nuevos_pacientes** utilizando la función **rbind**.
5. Selecciona las filas del dataframe **pacientes** donde el diagnóstico sea "Diabetes".
6. Selecciona las columnas **nombre**, **edad** y **tratamiento** del dataframe **pacientes**.
7. Selecciona las filas del dataframe **pacientes** donde el género sea "F" y la edad sea mayor o igual a 50.

En resumen...

	Dimensions	Mode (data "type")	Example
Vector	1 	Identical	<code>c(10,0.2,34,48,53)</code>
Matrix		Identical	<code>matrix(c(1,2,3, 11,12,13), nrow = 2, ncol = 3)</code>
Data frame		Can be different	<code>data.frame(x = 1:3, y = 5:7)</code>
Array		Identical	<code>array(data = 1:3, dim = c(2,4,2))</code>
List	$\left\{ \begin{array}{l} \text{Vector} \\ \text{Matrix} \\ \text{Data frame} \\ \text{Array} \end{array} \right\}$	Can be different	<code>list(x = cars[,1], y = cars[,2])</code>

¿Más
preguntas?

rpucheq@gmail.com

Twitter: [@rpucheq](https://twitter.com/rpucheq)

