

A light blue dotted world map serves as the background for the central text.

# DATAFRAME EN > R CON RSTUDIO >

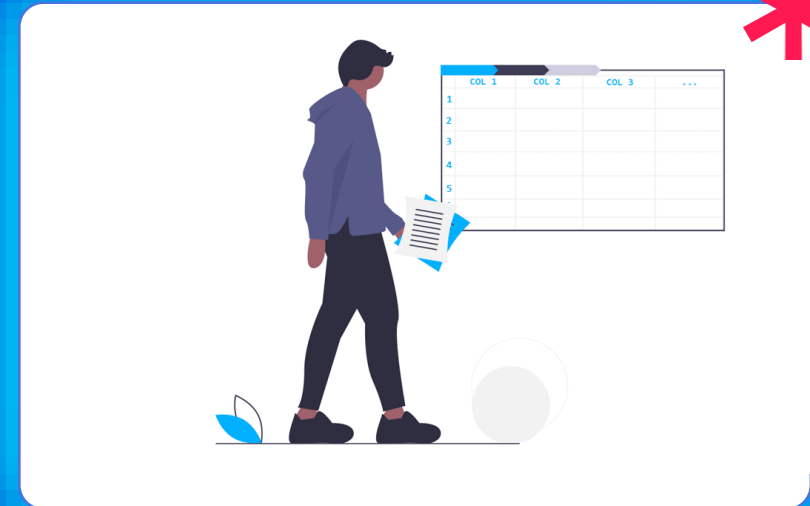
En esta sección abordaremos los siguientes tópicos con relación a las Data Frames en R usando Rstudio:



- Crear data frames en R.
- Añadir y eliminar columnas y filas de un data frame en R.
- Seleccionar elementos de un data frame en R.
- Ordenar un data frame en R.

# \* ¿QUÉ ES UN DATAFRAME EN R?

Los data frames (marcos de datos) son el objeto más habitual para almacenar datos en R, en este tipo de objeto, cada individuo corresponde a una fila y cada columna corresponde a una variable. Dentro de este tipo de estructura puedes almacenar diferentes tipos de datos. Es decir que en un dataframe por ejemplo podemos tener una columna con caracteres, otra con números y otra con variables lógicas. Por lo tanto, son importante para hacer gráficas, análisis de modelos que tienen muchas variables estadísticas





## ***DATAFRAME O MATRIZ***

Una cuestión habitual es preguntarse en qué casos se debe de usar un data frame o una matriz en R. Los data frames son estructuras de datos muy similares a las matrices, pero en el caso de los data frames puedes tener diferentes tipos de datos dentro de las columnas. En consecuencia, la diferencia es que las matrices almacenan tipos de datos homogéneos mientras que los data frames almacenan tipos de datos heterogéneos.



## ***¿CÓMO CREAR UN DATAFRAME EN R?***

En R es muy sencillo crear un nuevo data frame, hay varias formas de crear DataFrame, lo mas sencillo es crear uniendo las variables haciendo uso de la función `data.frame()` para convertir los datos a la estructura de datos de tipo data frame.

Primero, necesitamos tener algunas variables almacenadas para crear el marco de datos en R.





## CREAR UN DATAFRAME VACÍO EN R?

- En ocasiones puede resultar interesante inicializar un data frame vacío y rellenarlo dentro de un bucle o añadir las filas mediante algún procedimiento. En este caso, la forma más recomendada de crear una estructura de datos vacía es utilizando la función `data.frame` y creando variables vacías.



## ACCEDIENDO A LOS DATOS DEL DATAFRAME

Hay varias formas de acceder a las columnas almacenadas en los data frames:

1. Usando el signo del dollar (\$) y el nombre de la columna.
2. Usando corchetes con el índice de la columna después de una coma.

***Acceso directo utilizando la función `attach`-deshabilitar el acceso directo con la función `detach`***





## ***AÑADIR COLUMNAS Y FILAS A UN DATA FRAME***

Muchas veces necesitamos modificar los datos para agregar nuevas filas o columnas, o eliminarlas. Utilizaremos el conjunto de datos cars de R, registrado en la década de 1920, que forma parte de los conjuntos de datos de ejemplo de R.



## ***ELIMINAR COLUMNAS Y FILAS DE UN DATAFRAME***

Ahora, para borrar variables o filas de un data frame, tenemos varias opciones:

1. Usar el signo menos (-) e indicar las columnas o filas que quieras borrar.
2. Crear un subconjunto de los datos que quieras conservar.





## ORDENANDO Y FILTRANDO DATOS DE UN DATA FRAME EN R

Es habitual ordenar o filtrar los datos dentro de los data frames por los valores de alguna variable.

Objetivos de aprendizaje: marcos de datos

Use las siguientes funciones en los data frames:

- `dim()`
- `str()`
- `summary()`
- `colnames()`
- `head()`
- `tail()`
- `View()`
- Aplique técnicas de corte, adición y clasificación con marcos de datos



```
File —> Open File—> Lab03 DF-Rbootcamp —> open
```

Hay ciertas funciones que usa para comprender más sobre el marco de datos que ha creado. Por ejemplo, `dim()` le permite ver la dimensión del marco de datos.

```
d <- cars  
  
h <- d$Height..in.  
w <- d$Weight..lbs.  
df <- data.frame(h, w)  
  
print(dim(df))
```







# STRUCTURE

Puede usar el comando `str()` para ver una descripción más detallada del marco de datos.

```
d <- cars  
  
h <- d$Height..in.  
w <- d$Weight..lbs.  
df <- data.frame(h, w)  
  
print(str(df))
```

```
Resultado:  
'data.frame':  18 obs. of  2  
variables:  
 $ h: int  74 68 70 72 66 66 64 71  
72 69 ...  
 $ w: int  170 166 155 167 124 115  
121 158 175 143 ...  
NULL
```

La descripción le dice que su estructura `data.frame` tiene 18 objetos que se almacenan dentro de 2 variables. También enumera la variable y su tipo de elemento, así como los primeros 10 datos. `NULL` se devuelve por defecto cuando se usa `str()`, así que no te preocupes por eso.





# SUMMARY

`summary()` es una función increíblemente útil que devuelve diferentes tipos de estadísticas con respecto a su marco de datos.

```
d <- cars  
  
h <- d$Height..in.  
w <- d$Weight..lbs.  
df <- data.frame(h, w)  
  
print(summary(df))
```

Resultado:

h	w
Min. :62.00	Min. : 98.0
1st Qu.:66.25	1st Qu.:132.0
Median :69.50	Median :150.0
Mean :69.06	Mean :146.7
3rd Qu.:71.75	3rd Qu.:165.2
Max. :75.00	Max. :176.0

Le proporciona los valores mínimo, primer cuantil, mediana, media, tercer cuantil y máximo para cada columna de datos dentro de su marco de datos. Desde aquí, puede utilizar estos datos para un análisis de datos adicional.





# CUARTIFILES

Los cuantiles a veces se denominan cuartiles. Cada cuantil representa el 25% o una cuarta parte de los datos. En R:

- El primer cuantil representa el punto de datos que se encuentra en la marca del 25 por ciento del conjunto de datos ordenados.
- El segundo cuantil representa el punto de datos que se encuentra en la marca del 50 por ciento del conjunto de datos ordenados (más conocido como la mediana).
- El tercer cuantil representa el punto de datos que se encuentra en la marca del 75 por ciento del conjunto de datos ordenados.
- Y, por último, el cuarto cuantil representa el punto de datos que se encuentra en la marca del 100 por ciento del conjunto de datos ordenados (más comúnmente conocido como el máximo).

Si solo desea calcular cuantiles, puede usar la función `quantile()` en un vector. Por ejemplo:

```
print(quantile(h))  
print(quantile(w))
```





# NOMBRES DE COLUMNA

- Use `colnames()` para devolver todos los nombres de columna o categorías de datos que tiene.

```
d <- cars  
  
h <- d$Height..in.  
w <- d$Weight..lbs.  
df <- data.frame(h, w)  
  
print(colnames(df))
```

Tenga en cuenta que actualmente solo tenemos 2 columnas de datos dentro de nuestro marco de datos, "h" y "w". Siéntase libre de cambiar el nombre de estas variables según sea necesario para que sean más descriptivas o más fáciles de entender.





## RENOMBRAR CON COLNAMES()

De hecho, puede usar `colnames()` para cambiar los nombres de sus columnas. Veamos este ejemplo que tiene la adición de la columna Nombre.

```
d <- cars
h <- d$Height..in.
w <- d$Weight..lbs.
n <- d$Name
df <- data.frame(h, w, n)

print(colnames(df))
```

Para cambiar el nombre de una columna específica, puede usar:

```
colnames(df)[colnames(df)=="h"] <- "height"
print(colnames(df))
```

Para cambiar el nombre de todas las columnas, puede usar:

```
colnames(df) <- c("height", "weight", "name")
print(colnames(df))
```

Tenga en cuenta que si no proporciona a cada columna un nombre nuevo, esa columna tendrá una nueva etiqueta de NA. Para cambiar el nombre de algunas columnas, puede usar:

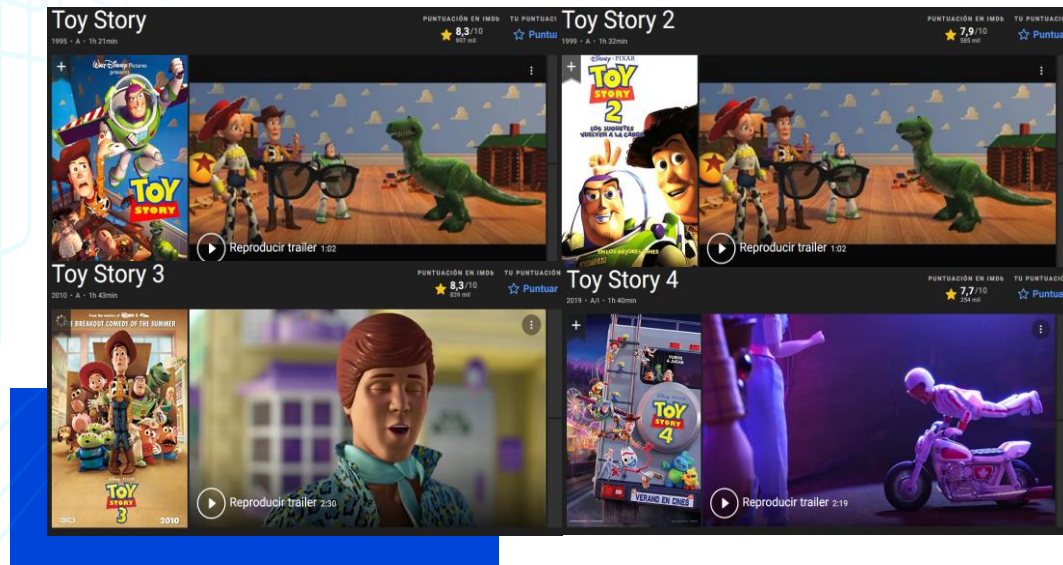
```
colnames(df)[colnames(df) %in% c("h", "n")] <- c("height", "name")
print(colnames(df))
```





## EJEMPLO DE DATAFRAMES EN R

Para practicar, vamos a utilizar los datos de las películas de toy story, en la sección de vectores ya teníamos vectores con la información del nombre, puntuación que da internet y saber si la película es posterior a 2000.



```
# vectores con el nombre de las películas
name <- c("Toy story", "Toy story 2: los juguetes
vuelven a la carga", "Toy story 3", "Toy story 4")
puntuacion <- c(8.3, 7.9, 8.3, 7.7)
posterior_2000 <- c(FALSE, FALSE, TRUE, TRUE)
```





# CÓMO CREAR UN DATAFRAME EN R

## Cómo crear un DataFrame en R?

```
data.frame(vector_1,  
            vector_2,  
            vector_3,  
            vector_etc)
```

```
# crear dataframe de vectores  
películas_df <- data.frame(name,  
                            puntuacion,  
                            posterior_2000)  
  
# mostrar dataframe  
head(películas_df)  
  
# cambiar nombre de dataframe  
names(películas_df) <- c('NAME',  
                          'PUNTUACION',  
                          'POSTERIOR_2000')  
  
# mostrar dataframe  
películas_df
```

## Seleccionar elementos de un DataFrame en R

```
# seleccionar un elemento del dataframe  
películas_df[3, 2]  
películas_df[3, 'PUNTUACION']  
  
# seleccionar más de un elemento del dataframe  
películas_df[c(3,4), c(2,3)]  
películas_df[c(3,4), c('PUNTUACION', 'POSTERIOR_2000')]  
  
# seleccionar una fila o renglón del dataframe  
películas_df[3,]  
  
# seleccionar una columna del dataframe  
películas_df[, 2]  
películas_df[, 'PUNTUACION']  
películas_df$PUNTUACION
```





## ORDENAR UN DATAFRAME EN R

```
#1. Guardar los índices ordenados, que obtenemos con la función order()
#       order(dataframe$columna)
#2. Seleccionar el dataframe con los índices obtenido en el paso uno
#       Indice<- order(dataframe$columna)

# mostrar el indice de la columna de puntuacion con order
order(peliculas_df$PUNTUACION)
# funcion order (menor a mayor)
orden_menor_mayor <- order(peliculas_df$PUNTUACION,
                          decreasing = FALSE)

# mostrar el dataframe ordenado
peliculas_df[orden_menor_mayor, ]

# funcion order (mayor a menor)
orden_mayor_menor <- order(peliculas_df$PUNTUACION,
                          decreasing = TRUE)

# mostrar el dataframe ordenado
peliculas_df[orden_mayor_menor, ]

# guardar el dataframe ordenado
df_ordenado <- peliculas_df[orden_mayor_menor, ]
```



A light blue world map composed of small dots, centered in the background.

***GRACIAS***