Función merge en R

La función **merge** en R **permite fusionar o unir dos data frames por columnas comunes o por nombres de fila**. Esta función permite realizar diferentes combinaciones de bases de datos (SQL):

- ✓ unión izquierda (left join),
- ✓ unión interna (inner join),
- ✓ unión derecha (right join)
- ✓ unión completa (full join)
- ✓ otras

Sintaxis de la función merge en R

La sintaxis de la función **merge** en R con una breve descripción de sus argumentos se muestra en el siguiente bloque de código:

merge (x, y, ...)

```
# Para data frames:
merge(x, y, # Data frames u objectos a ser transformados
    by = intersect(names(x), names(y)), # Columnas usadas para unir
    by.x = by, by.y = by, # Columnas usadas para unir
    all = FALSE, # Si TRUE, all.x = TRUE y all.y = TRUE
    all.x = all, all.y = all, # Si TRUE, añade filas para cada fija en x (y) que no coincide con una en y (x).
    sort = TRUE, # Ordenar (TRUE) las columnas por las columans usadas en el argumento 'by'
    suffixes = c(".x",".y"), # Sufijos para crear nombres de columna únicos
    no.dups = TRUE, # Evitar nombres de columnas duplicados (TRUE) añadiendo más sufijos
    incomparables = NULL, # Como tratar los valores que no se pueden unir
    ...) # Argumentos adicionales
```

Ten en cuenta que el **método principal de la función merge es para data frames**. Sin embargo, merge es una función genérica que **también** se puede usar con otros objetos (como **vectores** o **matrices**), pero serán transformados a la clase data.frame.

Unir data frames en R

Para crear un ejemplo reproducible que muestre cómo unir dos data frames en R, vamos a utilizar los siguientes conjuntos de datos de muestra denominados **df_1**, que representan el id de identificación, el nombre y el salario mensual de algunos empleados de una empresa y **df_2**, que muestra la identificación, nombre, edad y cargo de algunos empleados.

```
Conjuntos de datos de ejemplo
set.seed(61)

empleado_id <- 1:10
empleado_nombre <- c("Andrew", "Susan", "John", "Joe", "Jack",  "Jacob", "Mary", "Kate", "Jacqueline",
"Ivy")
#10 numeros aleatorios con media de 1500 y desviación estándar de 200
empleado_salario <- round(rnorm(10, mean = 1500, sd = 200))
```

```
#10 numeros aleatorios con media de 50 y desviación estándar de 8
empleado_edad <- round(rnorm(10, mean = 50, sd = 8))

empleado_puesto <- c("CTO", "CFO", "Administrativo", rep("Técnico", 7))
df_1 <- data.frame(id = empleado_id[1:8], nombre = empleado_nombre[1:8],
                   salario_mensual = empleado_salario[1:8])
#Se excluye el 5 elemento de cada vector
df_2 <- data.frame(id = empleado_id[-5], nombre = empleado_nombre[-5],
                   edad = empleado_edad[-5], position = empleado_puesto[-5])

df_1
df_2
```

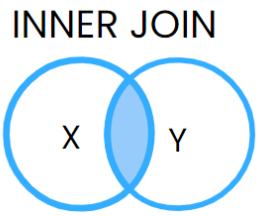
Output

```
 # df_1
  id  nombre  salario_mensual
  1  Andrew      1424
  2   Susan      1425
  3    John      1156
  4     Joe      1570
  5    Jack      1223
  6   Jacob      1462
  7    Mary      1641
  8    Kate      1603
```

```
 # df_2
 id      nombre   edad         puesto
 1       Andrew    40            CTO
 2        Susan    38            CFO
 3         John    54  Administrativo
 4          Joe    66         Técnico
 6        Jacob    38         Técnico
 7         Mary    53         Técnico
 8         Kate    56         Técnico
 9   Jacqueline    55         Técnico
 10         Ivy    43         Técnico
```

Nótese que en un ejemplo real, todas las identificaciones serán únicas, pero los nombres pueden repetirse. También ten en cuenta que 'Jack' falta en la segunda tabla (ni su edad ni su posición están disponibles) y 'Jacqueline' y 'Ivy' faltan en la primera (sus salarios mensuales no están disponibles con los datos actuales).

Inner join



Un inner join (en realidad un natural join), es la **unión de conjuntos de datos más habitual** que se puede realizar. Consiste en fusionar dos data frames en uno que contenga los elementos comunes de ambos, como se describe en la siguiente ilustración:
XYINNER JOIN

Para **fusionar o unir** los dos conjuntos de datos de muestra, solo tienes que pasarlos a la función **merge**, sin la necesidad de cambiar otros argumentos, debido a que, de manera predeterminada, la función combina los conjuntos de datos por los nombres de las columnas comunes. En consecuencia, en este caso, la función fusiona los datos por dos columnas (id y nombre).

```
merge(x = df_1, y = df_2)

merge(x = df_1, y = df_2, by = c("id", "nombre")) # Equivalente
```

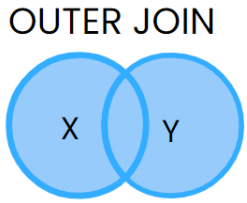
Output

```
 id   nombre  salario_mensual  edad        puesto
  1  Andrew      1424          40           CTO
  2   Susan      1425          38           CFO
```

```
3   John     1156       54   Administrativo
4    Joe     1570       66        Técnico
6  Jacob     1462       38        Técnico
7   Mary     1641       53        Técnico
8   Kate     1603       56        Técnico
```

Tal y como dijimos antes, 'Jack', 'Ivy' y 'Jacqueline' no están en ambos conjuntos de datos. En consecuencia, sus datos no están presentes en la salida resultante de esta unión.

Full (outer) join



El outer join, o **unión completa**, combina todas las columnas de ambos conjuntos de datos en uno para todos los elementos:
XYOUTER JOIN

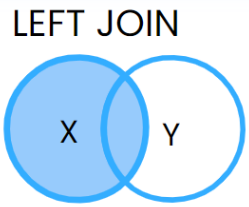
Para crear el full outer join de dos data frames en R tienes que establecer el argumento **all = TRUE**:

```
merge(x = df_1, y = df_2, all = TRUE)

Output
 id     nombre  salario_mensual  edad        puesto
  1     Andrew     1424         40           CTO
  2      Susan     1425         38           CFO
  3       John     1156         54   Administrativo
  4        Joe     1570         66        Técnico
  5       Jack     1223         NA           <NA>    # <-- valores NA
  6      Jacob     1462         38        Técnico
  7       Mary     1641         53        Técnico
  8       Kate     1603         56        Técnico
  9  Jacqueline      NA         55        Técnico    # <-- valores NA
 10        Ivy      NA         43        Técnico    # <-- valores NA
```

Como no todas las filas en el primer data frame coinciden con todas las filas en el segundo, en la salida aparecen valores NA en esos casos.

Left (outer) join



El left join en R consiste en **unir todas las filas del primer data frame con los valores correspodientes del segundo**. Recuerda que 'Jack' estaba en el primer conjunto de datos pero no en el segundo.
XYLEFT JOIN

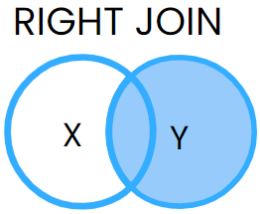
Para crear la unión, tienes que establecer **all.x = TRUE** como sigue:

```
merge(x = df_1, y = df_2, all.x = TRUE)
id  nombre  salario_mensual  edad        puesto
 1  Andrew     1424         40           CTO
 2   Susan     1425         38           CFO
```

```
3   John    1156     54  Administrativo
4    Joe    1570     66       Técnico
5   Jack    1223     NA          <NA>    # <-- Valores NA
6  Jacob    1462     38       Técnico
7   Mary    1641     53       Técnico
8   Kate     603     56       Técnico
```

Como el empleado que se corresponde al id = 5 está en el primer data frame pero no en el segundo, los correspondientes valores faltantes se representan como NA.

Right (outer) join



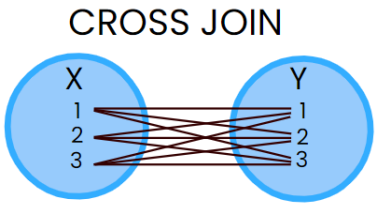
El **right join** en R es lo opuesto al left outer join. En este caso, la combinación consiste en unir todas las filas del segundo data frame con las correspondientes en el primero.
XYRIGHT JOIN

En consecuencia, necesitarás establecer el argumento **all.y = TRUE** para unir los data frames de esta manera.

```
merge(x = df_1, y = df_2, all.y = TRUE)
 id      nombre  salario_mensual  edad        puesto
  1      Andrew     1424          40           CTO
  2       Susan     1425          38           CFO
  3        John     1156          54  Administrativo
  4         Joe     1570          66       Técnico
  6       Jacob     1462          38       Técnico
  7        Mary     1641          53       Técnico
  8        Kate     1603          56       Técnico
  9  Jacqueline       NA          55       Técnico    # <-- Nótese la diferencia
 10         Ivy       NA          43       Técnico    # <-- respecto al left join
```

Como 'Jacqueline' y ' Ivy' están en el segundo conjunto de datos pero no en el primero, los valores correspondientes de salario mensual no están disponibles.

Cross join



El cross join o unión cruzada, realiza el producto cartesiano de los conjuntos de datos, en este caso de df_1 y df_2:
XYCROSS JOIN112323

Puedes crear un cross join en R estableciendo **como NULL el argumento by** de la función merge. Ten en cuenta que mostramos las primeras filas del output porque el resultado completo es muy grande.

```
Merged <- merge(x = df_1, y = df_2, by = NULL)
head(Merged)
```

```
Output
 id.x  nombre.x  salario_mensual  id.y  nombre.y  edad  puesto
    1   Andrew        1424         1    Andrew     40    CTO
    2    Susan        1425         1    Andrew     40    CTO
    3     John        1156         1    Andrew     40    CTO
    4      Joe        1570         1    Andrew     40    CTO
    5     Jack        1223         1    Andrew     40    CTO
    6    Jacob        1462         1    Andrew     40    CTO
```

Unir filas con la función merge en R

También puedes **unir data frames por los nombres de las filas**. Como ejemplo, considera los siguientes conjuntos de datos:

```
Data frames de ejemplo
df1 <- data.frame(var = c("uno", "dos", "tres", "cuatro", "cinco"),
                  datos = c(1, 5, 1, 6, 8))

rownames(df1) <- c("A", "B", "C", "D", "E")
df1

df2 <- data.frame(var = c("tres", "uno", "ocho", "dos", "nueve"),
                  datos = c(1, 5, 1, 6, 8))

rownames(df2) <- c("E", "A", "B", "D", "C")
df2

Output
# df1                    # df2
     var  datos                 var  datos
A    uno     1             E   tres     1
B    dos     5             A    uno     5
C   tres     1             B   ocho     1
D cuatro     6               D    dos     6
E  cinco     8             C  nueve     8
```

En este caso, para unir los data frames según los nombres de las filas tienes que establecer el argumento **by como 0** o como "**row.names**".

```
merge(df1, df2, by = 0, all = TRUE)
merge(df1, df2, by = "row.names", all = TRUE) # Equivalente

Output
 Row.names  var.x  datos.x  var.y  datos.y
1         A    uno        1    uno        5
2         B    dos        5   ocho        1
3         C   tres        1  nueve        8
4         D cuatro        6    dos        6
5         E  cinco        8   tres        1
```

Como puedes observar, la salida contiene tantas filas como nombres de fila distintos hay. Ten en cuenta que se ha aplicado outer join (ya que en este caso es equivalente a un left o right join), pero puedes unir los datos como quieras.

Unir más de dos data frames en R

Por último, cabe mencionar que puedes unir iterativamente data frames en R, concatenando la función **merge**. Considera, por ejemplo, los siguientes conjuntos de datos:

```
x <- data.frame(id = 1:4, año = 1995:1998)
x
y <- data.frame(id = c(4, 1, 3, 2),
                año = c(1998, 1995, 1997, 1996),
                edad = c(22, 25, 23, 24))
y
z <- data.frame(id = c(1, 2, 3),
                año = 1995:1997,
                salario = c(1000, 1200, 1599))
z
```

Conjuntos de datos

```
# x        # y             # z
 id  año     id  año edad    id  año  salario
  1 1995     4 1998   22     1 1995   1000
  2 1996     1 1995   25     2 1996   1200
  3 1997     3 1997   23     3 1997   1599
  4 1998     2 1996   24
```

Puede unir los tres data frames fusionando primero dos y uniendo la salida con el tercer conjunto de datos.

```
merge(x, merge(y, z))
```

Output

```
id  año   edad salario
1   1995  25   1000
2   1996  24   1200
3   1997  23   1599
```

Ten en cuenta que puedes especificar los argumentos que prefieras para cada combinación y que puedes concatenar tantos **merge** como necesites.

```
merge(x, merge(y, z, all = TRUE), all = TRUE)
```

Output

```
id año  edad  salario
1  1995  25   1000
2  1996  24   1200
3  1997  23   1599
4  1998  22    NA
```

Una alternativa más limpia es usar la función **Reduce**, de manera que en lugar de concatenar las funciones **merge**, se pasan los data frames en una lista. Sin embargo, en este caso tendrás que establecer los mismos argumentos para todas las uniones.

```
Reduce(function(x, y) merge(x, y), list(x, y, z))
```

Output

```
id  año   edad salario
1   1995  25   1000
2   1996  24   1200
3   1997  23   1599
```