Dividir datos en R

La función split permite **dividir datos en R** basándose en niveles de factores. En este tutorial mostraremos cómo dividir datos en R con diferentes ejemplos y revisaremos todos los argumentos de la función.

La función split en R

La función split divide los datos de entrada (x) en diferentes grupos (f). El siguiente bloque resume los argumentos de la función y su descripción.

```
split(x,              # Vector o data frame
    f,              # Grupos de clase factor, vector o lista
    drop = FALSE,     # Si eliminar los grupos no usados (TRUE) o no (FALSE)
    sep = ".",        # Cadena de caracteres para separar los grupos cuando f es una lista
    lex.order = FALSE, # Si la concatenación de factores debe ser ordenada léxicamente (TRUE) o no (FALSE)
    ...)            # Argumentos adicionales
```

Dividir un vector en R

Supón que tienes un vector cuyos elementos tienen asignado un nombre y donde el nombre de cada elemento corresponde al grupo al que pertenece el elemento. Puedes dividir el vector en dos vectores donde los elementos pertenezcan al mismo grupo, pasando los nombres del vector con la función names al argumento f.

```
Vector de ejemplo
a <- c(x = 3, y = 5, x = 1, x = 4, y = 3)
split(a, f = names(a))

Output
$`x`
x x x
3 1 4

$y
y y
5 3
```

También puedes pasar un vector de caracteres como parámetro del argumento f para indicar los grupos correspondientes de cada elemento, o directamente un objeto de tipo factor.

```
grupos <- c("Grupo 1", "Grupo 1", "Grupo 2", "Grupo 1", "Grupo 2")
split(a, f = grupos)
split(a, f = factor(grupos)) # Equivalente
Output
$`Grupo 1`
x y x
3 5 4

$`Grupo 2`
x y
1 3
```

Además, puedes dividir tus datos en múltiples grupos, generando interacciones entre ellos. Para tal propósito la entrada del argumento `f` debe ser una lista.

```
grupos_2 <- c("Tipo 1", "Tipo 1", "Tipo 1", "Tipo 2", "Tipo 1")
split(a, f = list(grupos, grupos_2))

# Equivalent
f1 <- factor(c("Grupo 1", "Grupo 1", "Grupo 2", "Grupo 1", "Grupo 2"),
             levels = c("Grupo 1", "Grupo 2"))
f2 <- factor(c("Tipo 1", "Tipo 1", "Tipo 1", "Tipo 2", "Tipo 1"),
             levels = c("Tipo 1", "Tipo 2"))
split(a, f = list(f1, f2))

Output
$`Grupo 1.Tipo 1`
x y
3 5

$`Grupo 2.Tipo 1`
x y
1 3

$`Grupo 1.Tipo 2`
x
4

$`Grupo 2.Tipo 2`
named numeric(0)
```

Nótese que, por defecto, las interacciones entre grupos se separan con un punto y que la salida contiene todos los grupos posibles, incluso cuando no hay observaciones en algunos de ellos. Sin embargo, puedes personalizar esto con los argumentos **sep** y **drop**, respectivamente.

```
vec_split <- split(a, f = list(f1, f2), drop = TRUE, sep = ": ")
vec_split
Output
$`Grupo 1: Tipo 1`
x y
3 5

$`Grupo 2: Tipo 1`
x y
1 3

$`Grupo 1: Tipo 2`
x
4
```

Cabe mencionar que con la función **unsplit** puedes recuperar el vector original, pero los nombres de los elementos se perderán.

```
unsplit(vec_split, list(f1, f2))
Output

<NA> <NA> <NA> <NA> <NA>

 3  5  1  4  3
```

Dividir un data frame en R

Puedes dividir un conjunto de datos en subconjuntos en función de una o más variables que representan grupos de datos. Considera el siguiente data frame:

```
Data frame de ejemplo
set.seed(3)
df <- CO2[sample(1:nrow(CO2), 10), ] #muestreo aleatorio de 10 elementos del dataframe CO2
head(df)
Output
   Plant       Type  Treatment conc uptake
15   Qn3     Quebec nonchilled   95   16.2
68   Mc1 Mississippi    chilled  500   19.5
32   Qc2     Quebec    chilled  350   38.8
27   Qc1     Quebec    chilled  675   35.4
49   Mn1 Mississippi nonchilled 1000   35.5
48   Mn1 Mississippi nonchilled  675   32.4
```

Puedes usar la función split para dividir el data frame en grupos basados, por ejemplo, en la variable **Treatment**.

```
Output
split(df, f = df$Treatment)
$`nonchilled`
   Plant       Type  Treatment conc uptake
15   Qn3     Quebec nonchilled   95   16.2
49   Mn1 Mississippi nonchilled 1000   35.5
48   Mn1 Mississippi nonchilled  675   32.4
10   Qn2     Quebec nonchilled  250   37.1
44   Mn1 Mississippi nonchilled  175   19.2

$chilled
   Plant       Type Treatment conc uptake
68   Mc1 Mississippi   chilled  500   19.5
32   Qc2     Quebec   chilled  350   38.8
27   Qc1     Quebec   chilled  675   35.4
23   Qc1     Quebec   chilled  175   24.1
79   Mc3 Mississippi   chilled  175   18.0
```

Puedes dividir un data frame en subconjuntos que cumplan diferentes combinaciones de grupos al mismo tiempo. Por ejemplo, puedes crear la división del data frame de muestra con las columnas **Type** y **Treatment**.

Esto creará cuatro subconjuntos con todas las combinaciones posibles de los grupos. Ten en cuenta que el número total de divisiones es la multiplicación del número de niveles de cada grupo.

```
dfs <- split(df, f = list(df$Type, df$Treatment))
dfs
Output
$`Quebec.nonchilled`
   Plant   Type  Treatment conc uptake
15   Qn3 Quebec nonchilled   95   16.2
10   Qn2 Quebec nonchilled  250   37.1

$Mississippi.nonchilled
   Plant        Type  Treatment conc uptake
49   Mn1 Mississippi nonchilled 1000   35.5
48   Mn1 Mississippi nonchilled  675   32.4
44   Mn1 Mississippi nonchilled  175   19.2

$Quebec.chilled
   Plant   Type Treatment conc uptake
32   Qc2 Quebec   chilled  350   38.8
27   Qc1 Quebec   chilled  675   35.4
23   Qc1 Quebec   chilled  175   24.1

$Mississippi.chilled
   Plant        Type Treatment conc uptake
68   Mc1 Mississippi   chilled  500   19.5
79   Mc3 Mississippi   chilled  175   18.0
```

Recuerda que puedes recuperar el data frame original con la función **unsplit**, pasando el data frame dividido y el grupo o grupos que utilizaste para crear la división.

```
unsplit(dfs, f = list(df$Type, df$Treatment))
```