

La función `sample()` en R: muestreo aleatorio y permutaciones

Adaptado del tutorial de R-Coder

Introducción

La función `sample()` en R permite crear *muestras aleatorias* o *permutaciones* de un vector o lista, ya sea con o sin reemplazo, e incluso basadas en probabilidades asignadas a cada elemento. :contentReference[oaicite:2]index=2 Es una herramienta fundamental en simulación, muestreo estadístico y análisis de datos en el que la aleatoriedad juega un papel clave.

1. Sintaxis de `sample()`

La forma general de la función es:

```
sample(x, size, replace = FALSE, prob =  
       NULL)
```

Donde los argumentos significan:

- **x**: vector o lista que contiene los elementos de los que se va a seleccionar. :contentReference[oaicite:3]index=3
- **size**: número de elementos a seleccionar. Si `replace = TRUE`, puede ser mayor que la longitud de `x`. :contentReference[oaicite:4]index=4
- **replace**: valor lógico que indica si el muestreo será con reemplazo (`TRUE`) o sin reemplazo (`FALSE`). Por defecto es `FALSE`. :contentReference[oaicite:5]index=5
- **prob**: un vector de pesos de probabilidad opcionales para cada elemento de `x`. :contentReference[oaicite:6]index=6

2. Muestreo sin reemplazo

Por defecto (`replace = FALSE`), `sample(x)` devuelve una *permutación* aleatoria de los elementos de `x`, es decir, los mismos valores en otro orden. :contentReference[oaicite:7]index=7

```
x <- 1:10  
sample(x)
```

También puede especificarse un tamaño menor usando el argumento `size`:

```
sample(x, size = 1)
```

Si se pide un tamaño mayor que la longitud de `x` sin reemplazo, se generará un error:

```
sample(x, size = 15)  
# Error in sample.int(length(x), size,  
#   replace, prob) :  
#   cannot take a sample larger than the  
#   population when 'replace = FALSE'
```

:contentReference[oaicite:8]index=8

3. Muestreo con reemplazo

Cuando `replace = TRUE`, el muestreo permite que los elementos seleccionados puedan repetirse. Por tanto el tamaño de la muestra puede exceder la longitud de `x`. :contentReference[oaicite:9]index=9

```
x <- 1:10  
sample(x, replace = TRUE)  
sample(x, size = 15, replace = TRUE)
```

4. Muestreo ponderado (weighted sampling)

Al especificar el argumento `prob`, cada elemento de `x` puede tener una probabilidad distinta de ser seleccionado. Esto permite modelar escenarios más realistas en los que los elementos no tienen probabilidades iguales. :contentReference[oaicite:10]index=10

```
sample(c("A", "B"), size = 10, replace =  
TRUE, prob = c(0.8, 0.2))
```

A medida que el tamaño de la muestra crece, las proporciones observadas tienden a acercarse a las probabilidades especificadas.

5. Casos de uso prácticos adicionales

Muestras reproducibles

Para garantizar que los resultados sean reproducibles, es buena práctica fijar una semilla antes de usar `sample()`.

```
set.seed(1)
sample(1:10)
```

:contentReference[oaicite:11]index=11

Selección de filas en un data frame

Es habitual usar `sample()` para seleccionar aleatoriamente filas de un data frame. Por ejemplo:

```
set.seed(21)
df <- data.frame(x = 1:10, y = rnorm(10))
df[sample(1:nrow(df), size = 5), ]
```

:contentReference[oaicite:12]index=12

Conclusión

La función `sample()` es una herramienta versátil para generar permutaciones o muestras aleatorias, ya sea con o sin reemplazo, y con posibilidad de ponderación. Su uso correcto —junto con la fijación de semilla cuando se requiere reproducibilidad— es esencial para estudios de simulación, análisis estadístico y muestreo en R.