



Tidy data: Función `pivot_wider()`

Contenidos

01. Función pivot_wider()



01.

Función pivot_wider()





Introducción

- En ocasiones necesitamos transformar la distribución o formato de nuestros *dataframes*, desde **wide** a **long**, o viceversa.
- Para esto el paquete **tidyr** nos ofrece las funciones **pivot_longer()** y **pivot_wider()**.
- En esta presentación, revisaremos la función **pivot_wider()**.

wide

id	x	y	z
1	a	c	e
2	b	d	f

long

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f



Utilidades de pivot_wider()

- Esta función, en contraste con pivot_longer(), ordena un dataframe **permitiendo mostrar categorías de una variable como columnas de un *dataframe***.
- Incrementa el número de las columnas y disminuye el número de las filas.
- El formato **wide** es útil para la presentación de cuadros resumen con doble entrada y es más legible para los humanos.

wide

id	x	y	z
1	a	c	e
2	b	d	f



Ejecutando pivot_wider()

Creamos objetos "df1" y "df2" a partir de nuestro dataframe y ejecutamos:

```
df1 <- data.frame(region = c(1, 2),  
                  hombres = c(100, 200),  
                  mujeres = c(50, 300))
```

```
df2 <- df1 %>%  
  pivot_longer(cols = -region ,  
               names_to = "sexo", values_to =  
               "total_sexo")
```

```
df2
```

Visualización de consola:

```
## # A tibble: 4 × 3  
##   region sexo      total_sexo  
##   <dbl> <chr>      <dbl>  
## 1     1 1 hombres      100  
## 2     1 1 mujeres       50  
## 3     2 2 hombres      200  
## 4     2 2 mujeres      300
```

Continuación





Ejecutando pivot_wider()

Llamamos a la librería tidyr, utilizamos función pivot_wider() y ejecutamos código:

```
library(tidyr)
df2 %>%
  pivot_wider(names_from
              = "sexo", values_from =
              "total_sexo")
```

Visualización de consola:

```
## # A tibble: 2 × 3
##   region hombres mujeres
##   <dbl>   <dbl>   <dbl>
## 1     1     100     50
## 2     2     200    300
```

Dataframe con pivot_longer

Región	Sexo	Total sexo
1	hombres	100
1	mujeres	50
2	hombres	200
2	mujeres	300

Dataframe con pivot_wider

Región	Hombres	Mujeres
1	100	50
2	200	300



Ejecutando pivot_wider()

Como pudimos notar, las observaciones correspondientes a las categorías "hombres" y "mujeres", se agrupan, a partir de la utilización de la función **pivot_wider()**. Ello permite disminuir **la cantidad de filas resultantes en el dataframe, lo que facilita el posterior análisis de datos.**

Los argumentos más comunes de esta función son:

- **names_from:** categorías que se quiere convertir en columnas.
- **values_from:** columna desde la cual extraer los valores.



En algunos casos, no contamos con una variable que contenga texto, para hacer el "pivoteo"...

```
## # A tibble: 32 × 3
## # Groups:   region [16]
##   region sexo    n
##   <dbl> <dbl> <int>
## 1     1     1  1187
## 2     1     2  1288
## 3     2     1  1536
## 4     2     2  1601
## 5     3     1  1409
## 6     3     2  1524
## 7     4     1  2481
## 8     4     2  2679
## # i 22 more rows
```



En esos casos, podemos usar el parámetro **names_prefix**



Ejecutamos código...

```

calidad::ene %>%
  group_by(region, sexo) %>%
  summarise(n = n()) %>%
  pivot_wider(names_from = sexo,
              values_from = n,
              names_prefix = "sexo" )

```



Visualización de consola:

```

## # A tibble: 16 × 3
## # Groups:   region [16]
##   region sexo1 sexo2
##   <dbl> <int> <int>
## 1     1     1  1187  1288
## 2     2     2  1536  1601
## 3     3     3  1409  1524
## 4     4     4  2481  2679
## 5     5     5  5296  5829
## 6     6     6  2611  2806
## 7     7     7  3121  3313
## 8     8     8  4975  5742
## 9     9     9  2204  2420
## 10    10    10  3128  3226
## 11    11    11  1003  1014
## 12    12    12   915   947
## 13    13    13  7184  8117
## 14    14    14  1771  1920
## 15    15    15  1481  1543
## 16    16    16  1687  1884

```

¡Muchas gracias!





www.ine.gob.cl