

Anatomía de una grafica

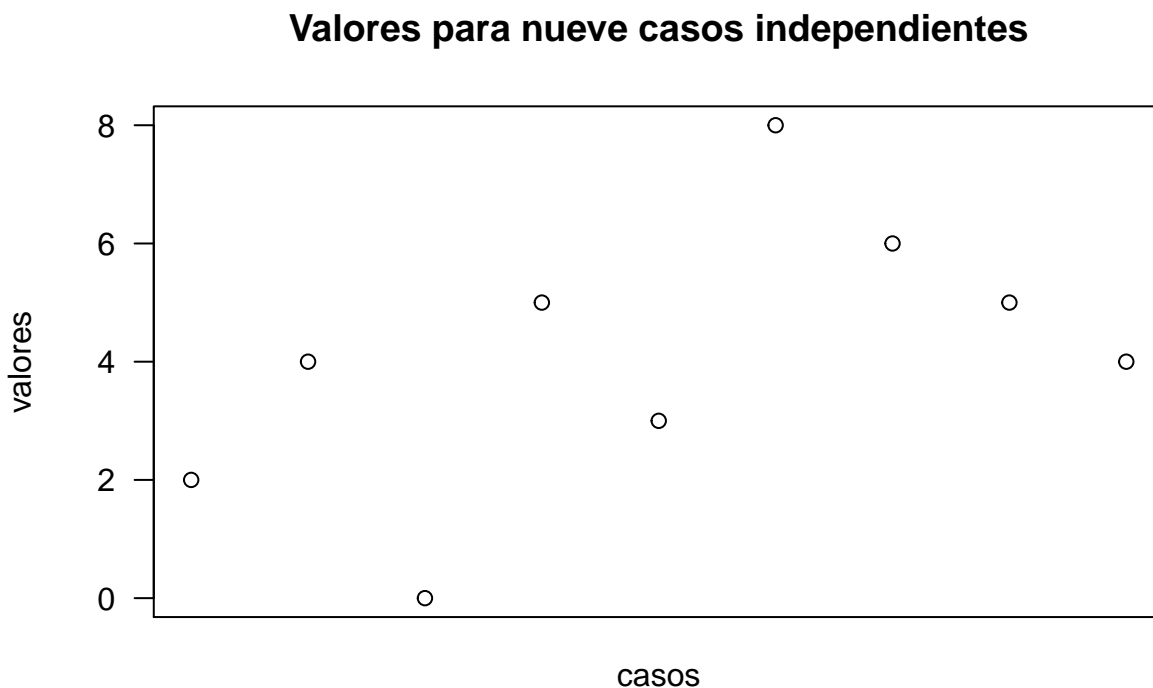
Análisis Exploratorio de Datos

¡Hola! en esta lección, vamos a explorar la anatomía de una gráfica cualquiera mediante los programas R y swirl. Si acabas de leer este texto, ¡ya estás en ellos!

Para esta lección, vamos a usar el sistema de gráficas base del R. Luego, conocerá también a un sistema alternativo denominado GGplot2 que se ofrece opciones poderosas y convenientes para gráficas exploratorias. ¡No vayas a creer lo que producirás!

Sin embargo, empezamos con el sistema base de R. Se puede hacer gráficas igual de poderosa y hermosa con ello, aunque con menos automatización. ¡Vámonos, ya!

```
# una gráfica cualquiera
plot(c(2, 4, 0, 5, 3, 8, 6, 5, 4),
     main = "Valores para nueve casos independientes",
     ylab="valores", xlab="", las=1, xaxt="n")
mtext("casos", side = 1, line = 1)
```



Ojalá haya aparecido una gráfica por allí a la derecha, ¿Si? o ¿Si? ¿Cuales son los elementos que debe llevar una gráfica así?

Empecémanos con los ejes. Los ejes definen el espacio o rango de una gráfica y lo más común es una gráfica con dos ejes, los cual denominamos el X y el Y. El X se ubica por convenio en el horizontal.

Antes de seguir, entonces, ¿como se llama la variable en el eje X de nuestra gráfica acá al lado?

- casos
- z

- valores
- no se lo ve

En este caso, con una variable sola, el X marca un espacio más o menos arbitrario, a lo largo de cual se arregló los 9 casos. El eje Y normalmente lleva la variable destacada, o de respuesta. Esta variable depende o cambia según el caso indicado por el eje X. Antes de seguir, entonces, ¿cómo se llama el eje Y en esta gráfica?

- casos
- z
- valores
- X

Ya vamos a crear una gráfica semejante, paso por paso, para asegurar que compartimos una lenguaje común de la anatomía de una gráfica. Sin embargo, no te preocupes, después de este ejercicio, jamás volveremos a construir una gráfica de tal modo.

Antes de trazar los ejes, necesitamos unos datos. ¿Te acuerdas de la lección inicial de R y swirl verdad? Bien, entonces haga un objeto 'v' compuesta del siguiente conjunto de valores c(2, 4, 0, 5, 3, 8, 6, 5, 4). Este "v" se podría considerar una variable.

```
v <- c(2, 4, 0, 5, 3, 8, 6, 5, 4)
```

Muy bien, vamos a graficarlos. He preparado ya el cuadro para la gráfica. Ya se la ve en la ventanilla de Plots. Añadé etiquetas que muestran como el R refiere el espacio fuera del cuadro.

```
plot(v, type="n", axes=F, xlab = "", ylab="")
box(lty = 3)
mtext(c("side = 1", "side = 2", "side = 3", "side = 4"),
      side = c(1, 2, 3, 4), col = "light grey", line = 2, cex = 2)
```



Acuérdete que los datos son 9 observaciones del valor de ... algo. Puesto que ya se ha creado el cuadro a la derecha, añadimos los datos escribiendo `points(v)`. Házlo ahora mismo y pulse Intro.

```
points(v)
```

Muy bien ¿no? Ahora vamos a dibujar los ejes, empezando con el X. Teclee `'axis(side = 1)'` para añadir el eje X. ¡Pruebalo ahora!

```
axis(side = 1)
```

¿Todo bien hasta aquí? Hmmmm, no lo creo. Ya mencioné que el eje X es algo arbitrario en este caso. Piénselo, ¿cuántos tipos de información lleva nuestra variable `v`? Teclee `v` para verlo en la consola.

```
v
```

```
## [1] 2 4 0 5 3 8 6 5 4
```

¿Ves que nuestro eje X representa nada más que la posición relativa de cada caso en la lista de valores '`v`'? La identidad del caso no nos importa, ¿cierto? Sin embargo, los valores en el X pueden distraernos. De hecho, no hay que añadir valores al eje X en este caso.

Perdóneme, pero mejor que empecemos de nuevo. Haya aparecido de nuevo el cuadro para la gráfica.

```
plot(v, type="n", axes=F, xlab = "", ylab="")
box(lty = 3)
mtext(c("side = 1", "side = 2", "side = 3", "side = 4"),
      side = c(1, 2, 3, 4), col = "light grey", line = 2, cex = 2)
```

side = 3

side = 2



side = 1

side = 1

Añadir los puntos nuevamente como hiciste hace unos pocos pasos. ¿Te acuerdas del comando? Si no, hay un atajo útil que uso mucho y que hubieras conocido en la primera lección. Pulse la flecha arriba hasta aparezca el comando que necesitas para trazar los puntos.

```
points(v)
```

Muy bien, ahora ponemos un eje más adecuado para esta situación en el X. Requiere unas instrucciones más para anular la opción que pone R por defecto. Teclee `axis(side = 1, at = 1:9, labels = FALSE)`.

```
axis(side = 1, at = 1:9, labels = FALSE)
```

Ahora, más fácil, ponemos el eje Y. En este caso las opciones por defecto están en nuestro favor. Escriba `axis(side = 2)` y pulse Intro.

```
axis(side = 2)
```

¿Ya hemos llegado? Pues, no, pero casi. Antes de seguir, entonces, ¿qué falta nuestra gráfica para cumplir con los requisitos mínimos?

- Etiquetas para los ejes más un título o subtítulo
- Etiquetas para cada eje
- Un subtítulo (o ‘caption’ en inglés)
- Un título

¡Vamonos pues! Puesto que estamos construyendo la gráfica paso por paso, el comando que necesitamos existe para añadir texto al margen. La función se denomina `mtext()`, una sigla para ‘margin text’. Seguro que reconoces un argumento de esta función `mtext`. Para etiquetar el eje X, teclee `mtext(text='Casos', side = 1, line = 2)` y pulse Intro.

```
mtext(text='Casos', side = 1, line = 2)
```

Y para el eje Y, repite el paso previo, cambiando la etiqueta ‘text’ a Valores, y el lado para el número adecuado para el eje Y.

```
"mtext(text = 'Valores', side=2, line=2)"
```

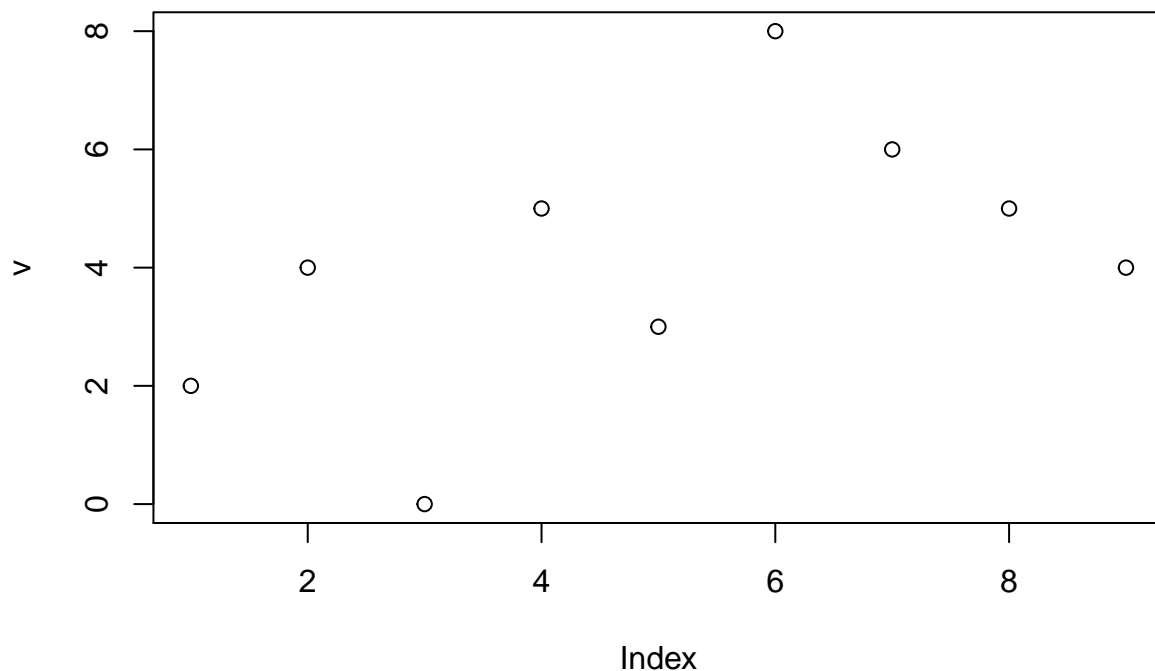
Acabas de etiquetar el eje Y, el dependiente. Lo llamamos así porque el valor en Y depende o pertenece al caso correspondiente en X.

Ahora el título. Vamos a llamar la función ‘`title()`’, así `title(main = ‘Valores de nueve casos independientes’)`. Pruébalo ahora por favor.

```
title(main = 'Valores de nueve casos independientes')
```

Qué piensas, un maratón? Pues, la buena noticia es que no hay que construir gráficas así. Normalmente, aquí en el sistema base de gráficas en R, usamos la función ‘`plot()`’. Para ver como sale con el comando más mínimo teclee `plot()` ingresando nuestro objeto `v`.

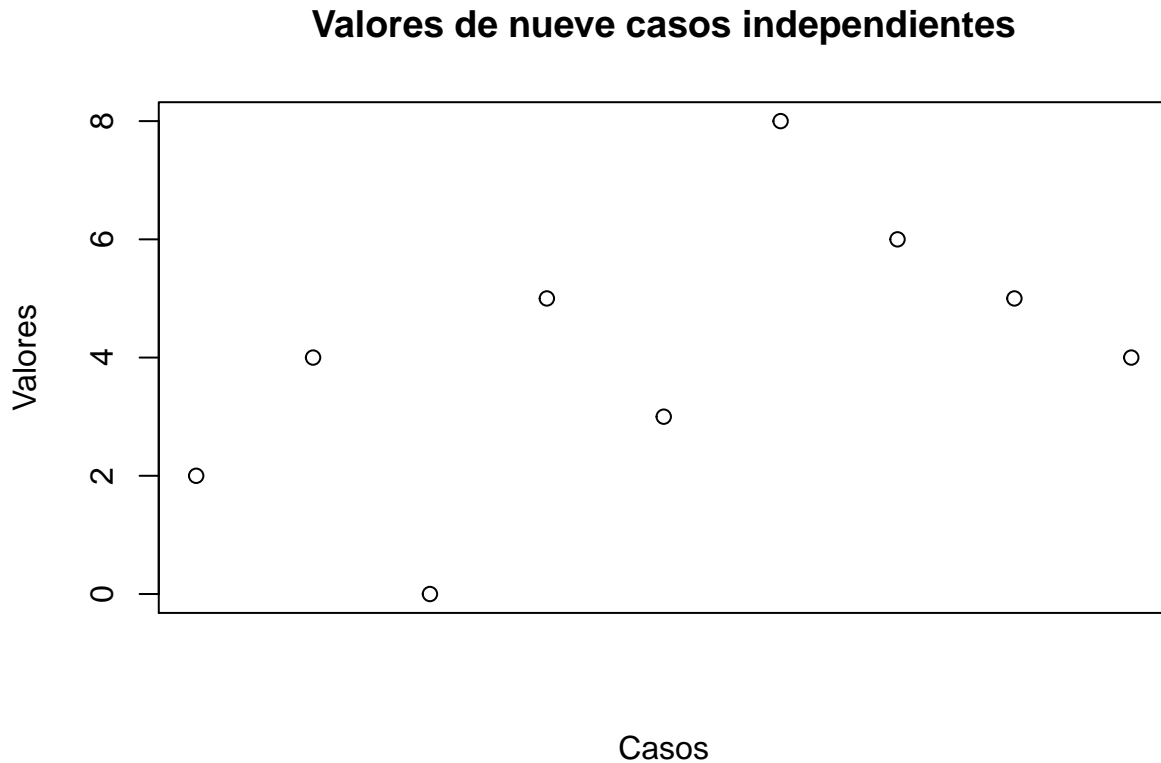
```
plot(v)
```



Fíjete que R haya etiquetado el eje Y con el nombre del objeto `v`, y la palabra ‘Index’ (índice) como etiqueta para el X. Tampoco proporcionó ningún título, aunque depende la gráfica R suele hacerlo.

No aceptamos estas opciones, así que vamos a ajustar la llamada a la función `plot()` para reemplazar algunas opciones con elementos de los últimos pasos. Teclee el siguiente `plot(v, main = ‘Valores de nueve casos independientes’, ylab=‘Valores’, xlab=‘Casos’, xaxt=‘n’)`

```
plot(v, main = 'Valores de nueve casos independientes', ylab='Valores', xlab='Casos', xaxt='n')
```



¡Buenazo! ¿no? Si yo fuera tu, guardaría este último pedazo de código en mi álbum de recortes.

Te explico los elementos o argumentos que hayas especificado para lograr la gráfica. Ya sabes que `v` contiene los valores, es nuestra variable o objeto. Reconocerías el argumento `'main='` del paso cuando hubieras puesto el título; `main =` principal. los argumentos `'ylab'` y `'xlab'` detallan el texto para etiquetar los ejes Y y X respectivamente. Por último, incluí `xaxt='n'` para que R no dibujara los números arbitrarios en el eje X. Hay muchísimas más opciones con las cuales se puede crear gráficas a medida, pero las opciones por defecto suelen hacer gráficas adecuadas.

Antes de terminar con este ejercicio, revisamos un poco. ¿Es necesario que siempre etiquetáramos los ejes, y proporcionáramos un título para cualquier gráfica?

- Si
- No

Y normalmente, antes de seguir, entonces, ¿en cuál eje se pone la variable de respuesta o la dependiente?

- No importa
- X
- Y