

# GGPLOT2. Gráfico de barras II - Fill y colores

*Raúl Ortiz*

*Saturday, July 11, 2015*

## Tutorial sobre el paquete ggplot2(), en R. Creación de gráficas de barras - Fill y colores.

Puedes seguir el tutorial por vídeo en <https://youtu.be/uC0TDAEW570>

### Empezamos introduciendo nuestros datos y creando el DataFrame

```
Tratamiento = factor(c(1,2,1,2,1,2), labels=c("Testigo", "Tratado"))
Color = factor(c(1,1,2,2,3,3), labels=c("Verde", "Envero", "Negra"))
Aceitunas = c(72,33,11,8,17,59)

df=data.frame(Tratamiento,Color,Aceitunas)
df
```

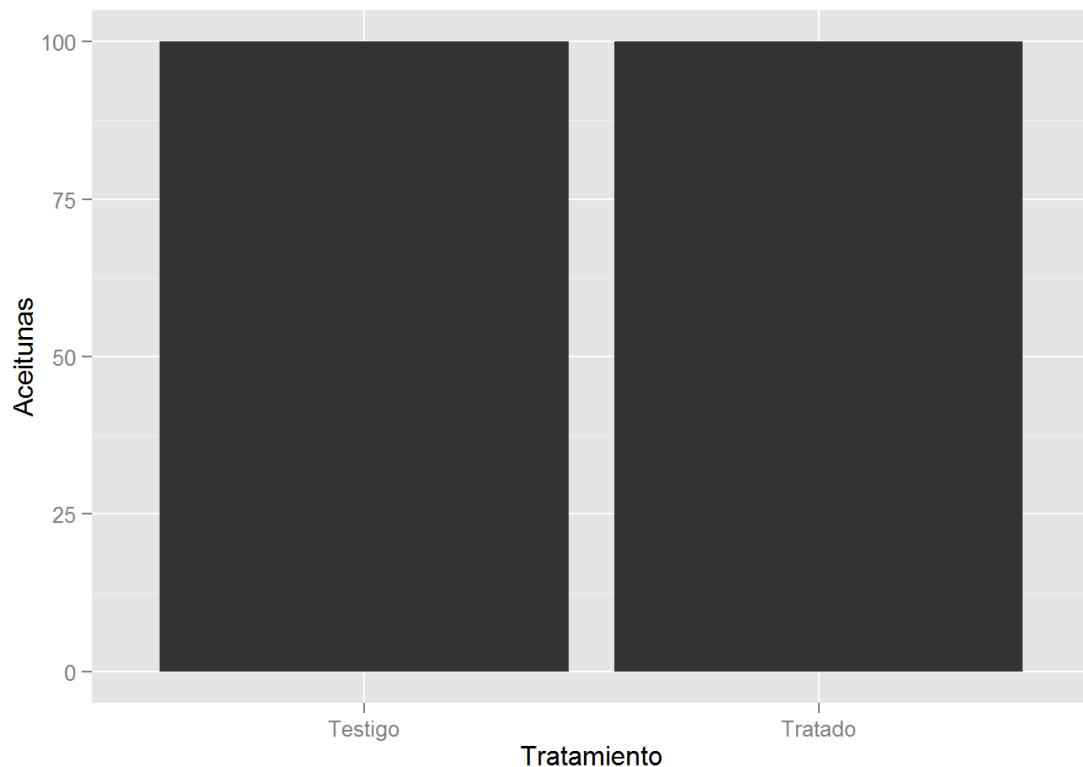
##	Tratamiento	Color	Aceitunas
## 1	Testigo	Verde	72
## 2	Tratado	Verde	33
## 3	Testigo	Envero	11
## 4	Tratado	Envero	8
## 5	Testigo	Negra	17
## 6	Tratado	Negra	59

```
#install.packages("ggplot2") # Si es la primera vez que lo usas, tendr
ás que instalar primero el paquete.

library(ggplot2) # Después hay que cargarlo cada vez que inicies sesi
ón.
```

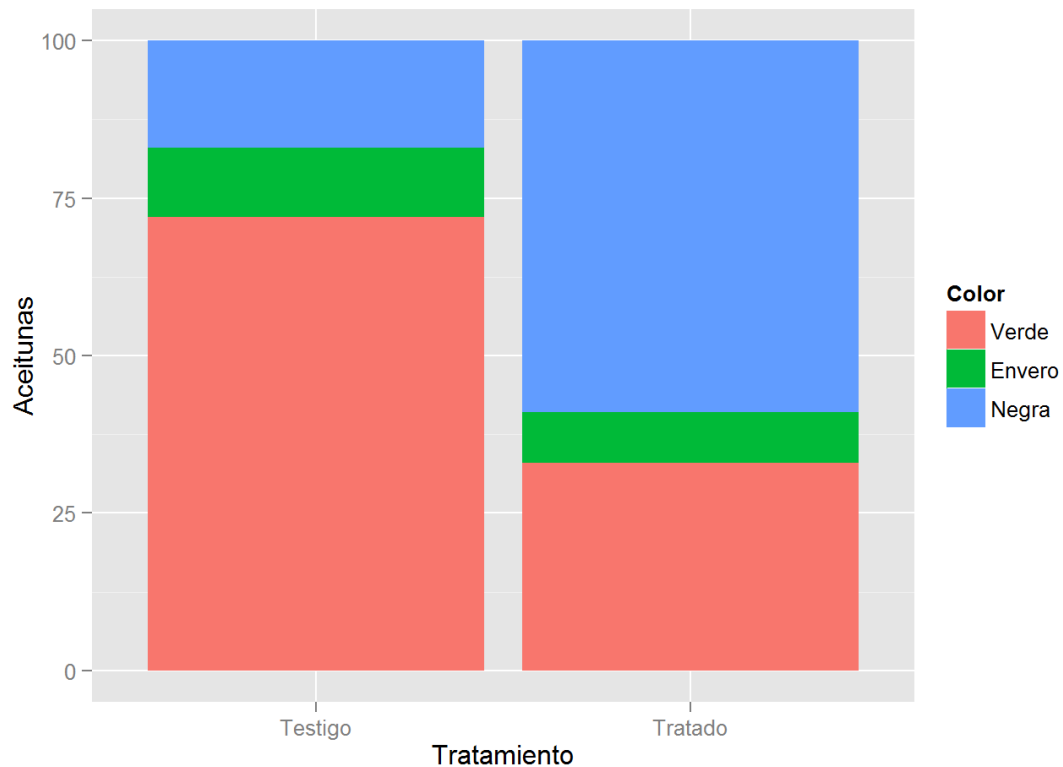
Representamos el número de aceitunas que se han muestreado en cada tratamiento.

```
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas)) + geom_bar(stat="identity")
```



Podemos hacer que el relleno en las columnas varíe según el color de las aceitunas, recogido nuestra variable “Color”, utilizando el comando “fill”

```
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas, fill=Color)) +  
  geom_bar(stat="identity")
```

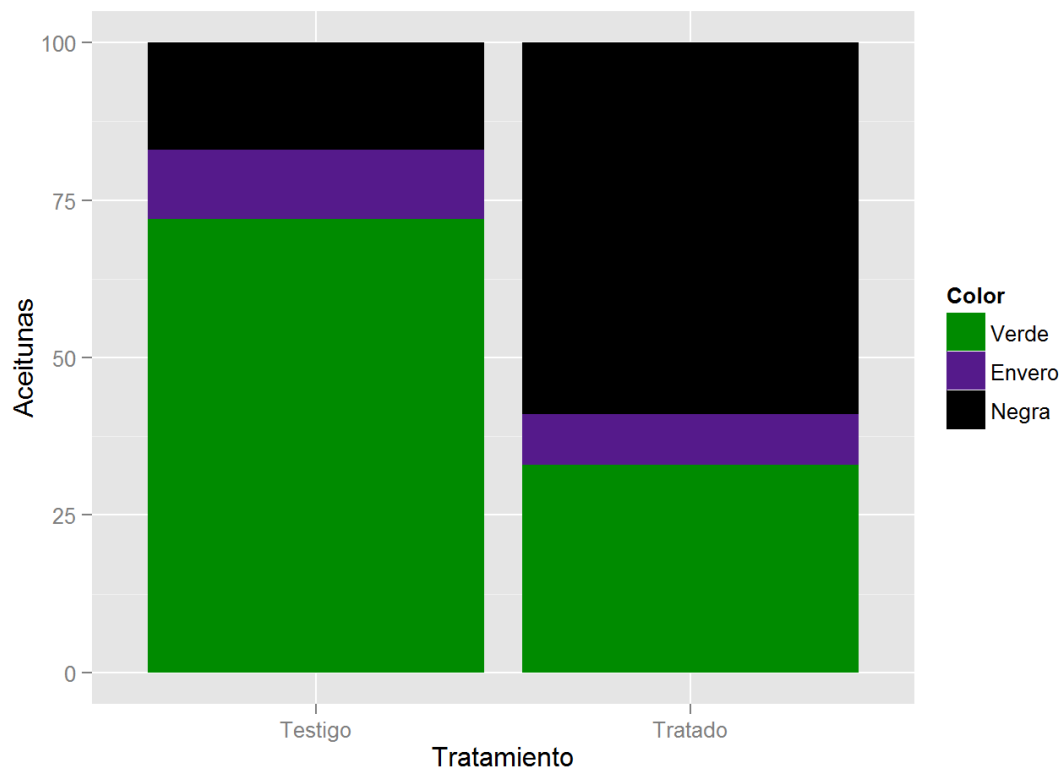


Podemos además modificar los colores, si preferimos usar otros para que la comprensión del gráfico sea mas intuitiva.

El color se puede definir especificando bien el nombre en inglés (por ejemplo “black”) o bien según el código hexadecimal (p.e.: “#FF0033”)

Ejemplo usando nombres.

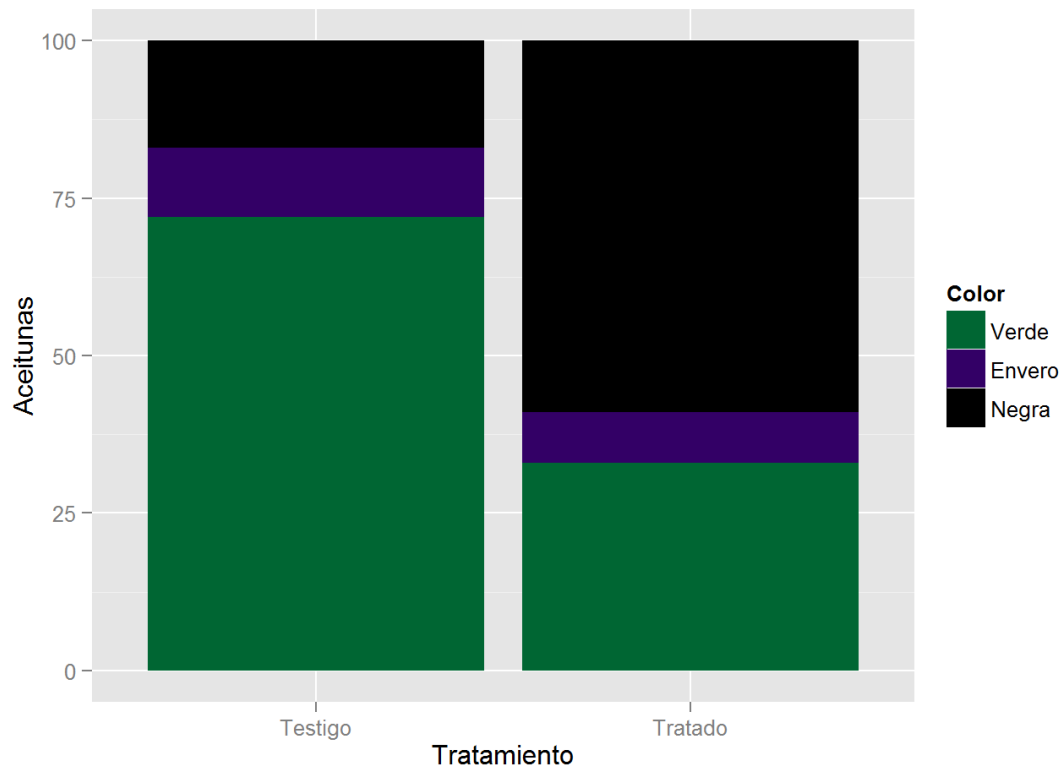
```
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas, fill=Color)) +  
  geom_bar(stat="identity") +  
  scale_fill_manual(values=c("green", "purple", "black"))
```



*# Para saber todos los nombres disponibles, se puede ejecutar la siguiente función.*

```
showCols <- function(cl=colors(), bg = "grey",
                     cex = 0.75, rot = 30) {
  m <- ceiling(sqrt(n <- length(cl)))
  length(cl) <- m*m; cm <- matrix(cl, m)
  require("grid")
  grid.newpage(); vp <- viewport(w = .92, h = .92)
  grid.rect(gp=gpar(fill=bg))
  grid.text(cm, x = col(cm)/m, y = rev(row(cm))/m, rot = rot,
            vp=vp, gp=gpar(cex = cex, col = cm))
}
showCols(cl= colors(), bg="gray33", rot=30, cex=0.75)
## Loading required package: grid
```

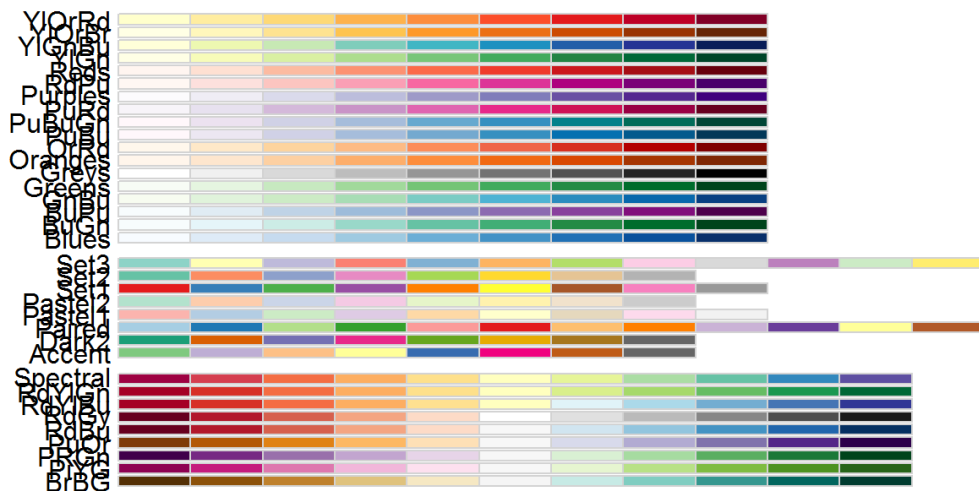




*# La correspondencia entre colores y códigos se puede consultar en la página <http://www.visibone.com>*

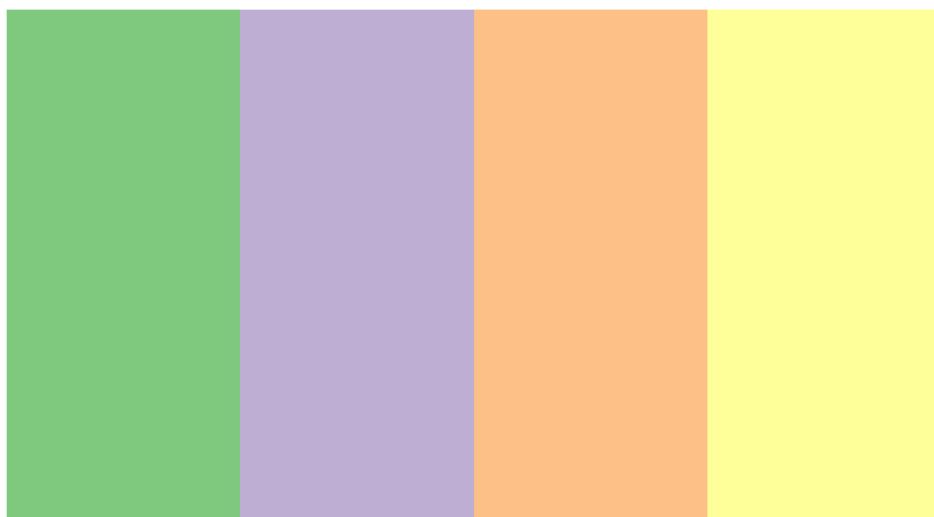
Se pueden utilizar paletas de colores RColorBrewer.

```
#install.packages("RColorBrewer")  
library("RColorBrewer")  
display.brewer.all()
```



*# Si quieres ver una de las paletas con mas detalles, puedes hacerlo e especificando su nombre y el número de franjas que quieres visializar ( mínimo 3 y máximo variable)*

```
display.brewer.pal(n = 4, name = 'Accent')
```



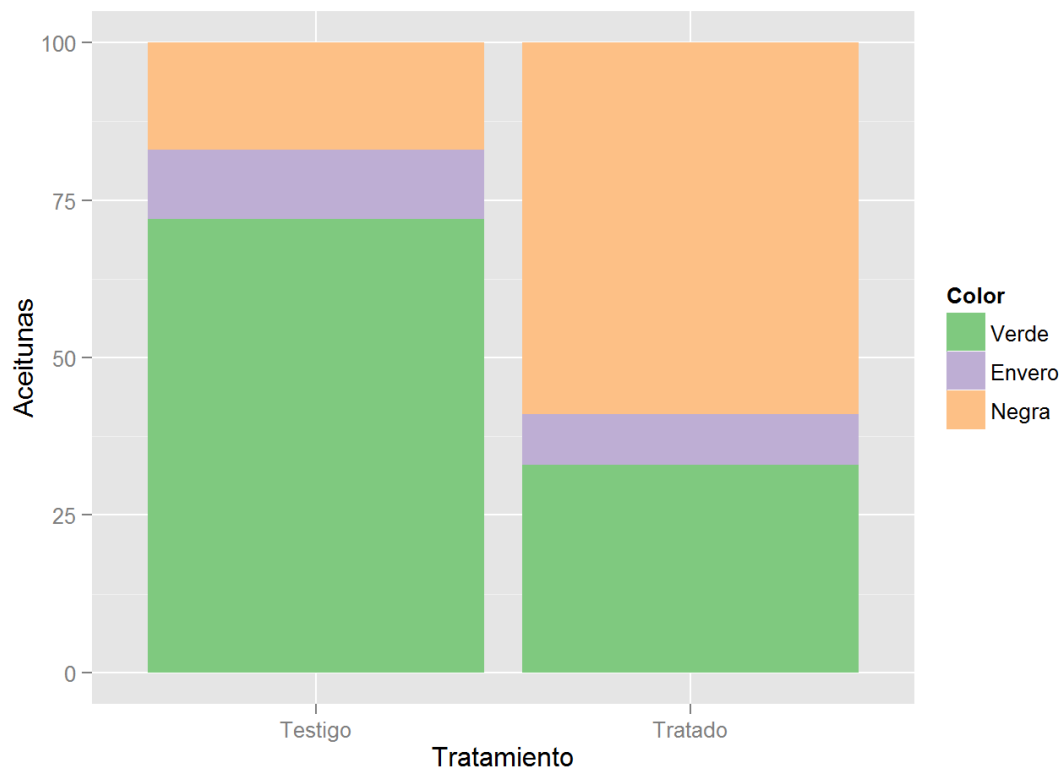
Accent (qualitative)

```
# En el código de la gráfica se incluiría de la siguiente manera.
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas, fill=Color)) +
  geom_bar(stat="identity") +
  scale_fill_manual(values=brewer.pal(n = 3, name = "Accent"))
```



```
# De manera uno poco mas abreviada.
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas, fill=Color)) +
  geom_bar(stat="identity") +
  scale_fill_brewer(palette = "Accent")
```





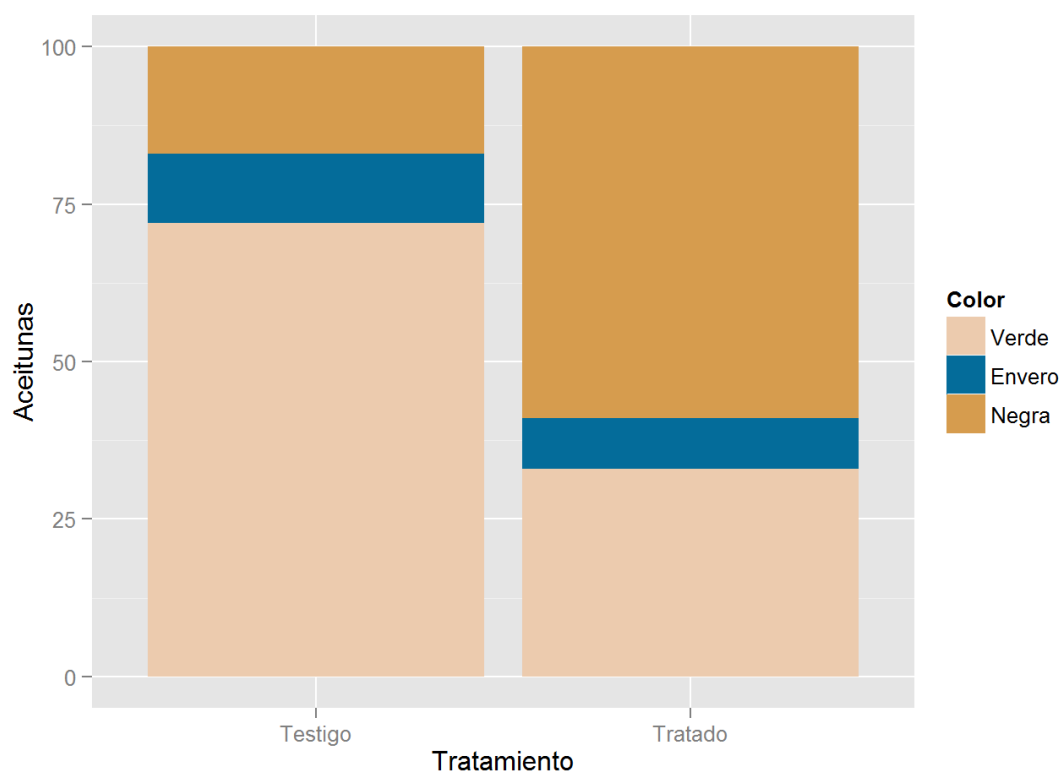
Lo anterior se puede aplicar a la paleta Wes Anderson.

```
#install.packages("wesanderson")
library(wesanderson)
## Warning: package 'wesanderson' was built under R version 3.2.3
wes_palettes # Listado completo de paletas
## $GrandBudapest
## [1] "#F1BB7B" "#FD6467" "#5B1A18" "#D67236"
##
## $Moonrise1
## [1] "#F3DF6C" "#CEAB07" "#D5D5D3" "#24281A"
##
## $Royal1
## [1] "#899DA4" "#C93312" "#FAEFD1" "#DC863B"
##
## $Moonrise2
## [1] "#798E87" "#C27D38" "#CCC591" "#29211F"
##
## $Cavalcanti
```

```
## [1] "#D8B70A" "#02401B" "#A2A475" "#81A88D" "#972D15"
##
## $Royal2
## [1] "#9A8822" "#F5CDB4" "#F8AFA8" "#FDDDA0" "#74A089"
##
## $GrandBudapest2
## [1] "#E6A0C4" "#C6CDF7" "#D8A499" "#7294D4"
##
## $Moonrise3
## [1] "#85D4E3" "#F4B5BD" "#9C964A" "#CDC08C" "#FAD77B"
##
## $Chevalier
## [1] "#446455" "#FDD262" "#D3DDDC" "#C7B19C"
##
## $Zissou
## [1] "#3B9AB2" "#78B7C5" "#EBCC2A" "#E1AF00" "#F21A00"
##
## $FantasticFox
## [1] "#DD8D29" "#E2D200" "#46ACC8" "#E58601" "#B40F20"
##
## $Darjeeling
## [1] "#FF0000" "#00A08A" "#F2AD00" "#F98400" "#5BBCD6"
##
## $Rushmore
## [1] "#E1BD6D" "#EABE94" "#0B775E" "#35274A" "#F2300F"
##
## $BottleRocket
## [1] "#A42820" "#5F5647" "#9B110E" "#3F5151" "#4E2A1E" "#550307" "#0
C1707"
##
## $Darjeeling2
## [1] "#ECCBAE" "#046C9A" "#D69C4E" "#ABDDDE" "#000000"
wes_palette("Moonrise2")
```

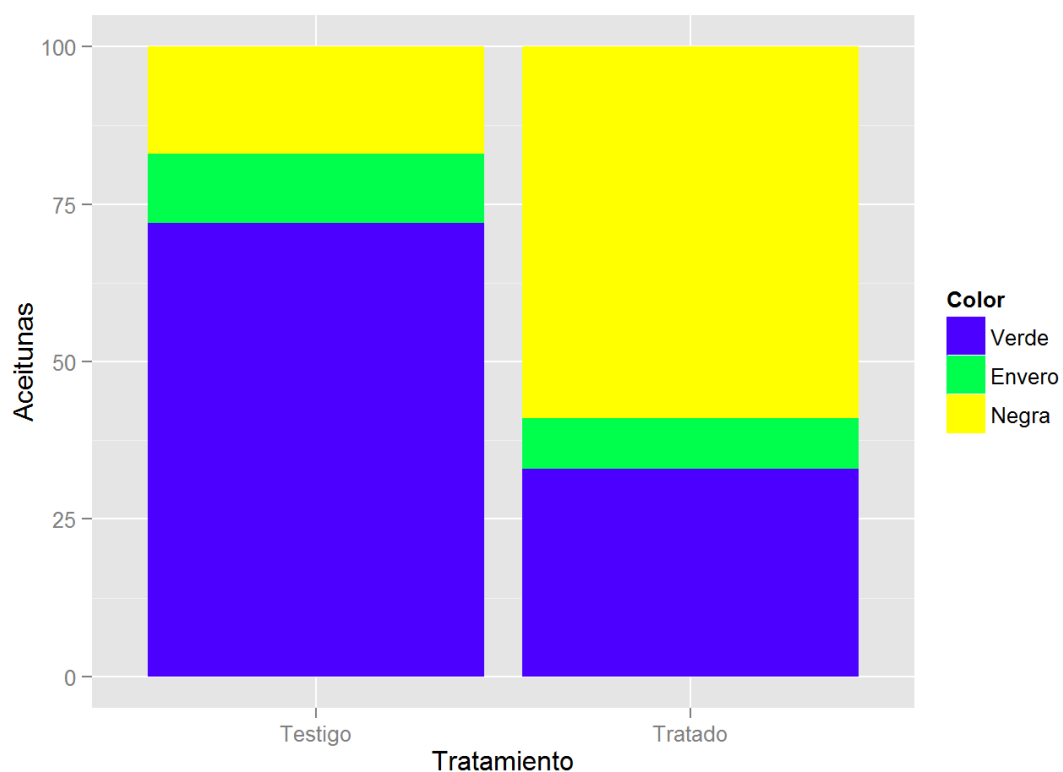


```
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas, fill=Color)) +  
  geom_bar(stat="identity") +  
  scale_fill_manual(values = wes_palette(n=3, name="Darjeeling2"))
```



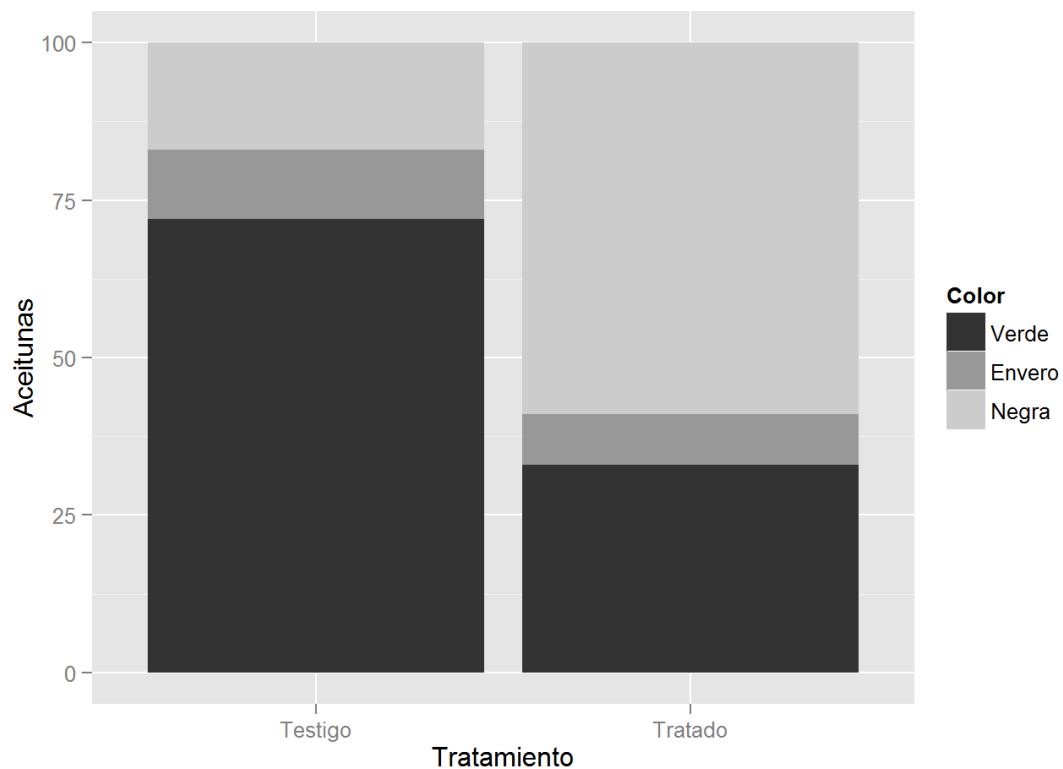
También se pueden hacer escalas de colores a partir de las funciones `rainbow(n)`, `heat.colors(n)`, `terrain.colors(n)`, `topo.colors(n)`, and `cm.colors(n)`

```
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas, fill=Color)) +  
  geom_bar(stat="identity") +  
  scale_fill_manual(values=topo.colors(3))
```



Si queremos una escala de grises.

```
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas, fill=Color)) +  
  geom_bar(stat="identity") +  
  scale_fill_grey()
```



Por último también se puede cambiar la luminiscencia “l” y la intensidad “c” y el tono de los colores “h”.

```
ggplot(data=df, aes(x=Tratamiento, y=Aceitunas, fill=Color)) +  
  geom_bar(stat="identity") +  
  scale_fill_manual(values=rainbow(3)) +  
  scale_fill_hue(l=50)  
  
## Scale for 'fill' is already present. Adding another scale for 'fill'  
, which will replace the existing scale.
```

