

# Chuleta Gráficos en R

*Fernando Villalba Bergado*

*Febrero de 2017*

# Índice

<b>PRIMER VISTAZO A LOS DATOS</b>	<b>1</b>
SELECCIONAR SUBCONJUNTOS DE LOS DATOS . . . . .	1
GENERAR SERIES ALEATORIAS . . . . .	2
<b>HISTOGRAMAS</b>	<b>2</b>
Histograma con ggplot2 . . . . .	5
<b>BOX WHISPER</b>	<b>6</b>
<b>DISPERSIÓN x-y</b>	<b>7</b>
X~Y CON MODELOS DE REGRESION . . . . .	9
GRÁFICOS DE LINEAS . . . . .	10
<b>MULTIGRÁFICAS</b>	<b>11</b>
<b>PALETA DE COLOR PERSONALIZADO</b>	<b>12</b>
Colores con ggplot2: . . . . .	14
<b>PINTAR EN FICHERO</b>	<b>15</b>
PDF . . . . .	15
PNG . . . . .	16
<b>VARIAS ESCALAS</b>	<b>16</b>

# Índice de figuras

1.	Histogramas base system . . . . .	2
2.	Histogramas base system . . . . .	3
3.	Histogramas base system . . . . .	4
4.	Histogramas . . . . .	4
5.	Histogramas . . . . .	5
6.	Histogramas ggplot . . . . .	5
7.	Histogramas ggplot . . . . .	6
8.	Graficos de caja Box-Whisper . . . . .	6
9.	Graficos de caja Box-Whisper . . . . .	7
10.	Graficos de dispersion . . . . .	7
11.	Graficos de dispersion . . . . .	8
12.	Graficos de dispersion . . . . .	8
13.	Graficos de dispersion . . . . .	9
14.	Graficos de lineas con modelo de regresion . . . . .	10
15.	Graficos de lineas . . . . .	11
16.	Uso del color en ggplot . . . . .	15
17.	Uso del color en ggplot . . . . .	15
18.	grafico con dos escalas distintas . . . . .	16



## PRIMER VISTAZO A LOS DATOS

Antes de hacer un gráfico vemos las dimensiones contenido etc. de forma rápida.

Para los ejemplos usaremos alguna veces datos incluidos en R de la librería datasets y otras veces generaremos series aleatorias rápidas.

Lectura de datasets que contiene tablas y dataframes de datos de ejemplo:

```
library(datasets) # cargamos el paquete de datos de muestra
head(cars)
dim(cars) # dimension de la tabla
names(cars) # para saber los nombres de las variables en la tabla cars

summary(cars$speed) # da el min, 25%,mediana=50%, media, 75% y max
quantile(cars$speed)
```

## SELECCIONAR SUBCONJUNTOS DE LOS DATOS

Es muy importante saber seleccionar datos de un data frame. Aparte de librerías específicas como dplyr y plyr, la manera más sencilla suele ser usar el propio lenguaje R:

```
# Seleccionar una columna por numero
x[,1]
# Seleccionar columna por name
x[,"var1"]
# Seleccionar parte de una fila y columna a la vez
x[1:2,"var2"]
# Seleccionar con operadores lógicos
# selecciono los valores que cumplen esta condición en todas las columnas
x[(x$var1<=3 & x$var3 >11),] # AND
x[(x$var1<=3 | x$var3>15),] # OR
# Seleccionar con which
x[which(x$var2>8),]
x[(x$var2>8),] # ver diferencia respecto a NA

# subset selecciona
subset(df, df$var1=="west")

# sumaparcial table suma frecuencias
table(cars$speed)
# split selecciona y agrupa subconjuntos
df_split<-split(df,list(df$year,df$type))

# aggregate suma parciales
sumaparcial<-aggregate(Emissions ~ year + type, data=BaltCity, sum)

# idem con dplyr
library(dplyr)
xta <- nei %>%
  filter(fips == 24510) %>%
```



```
group_by(year, type) %>%
  summarize(total.emissions = sum(Emissions))
```

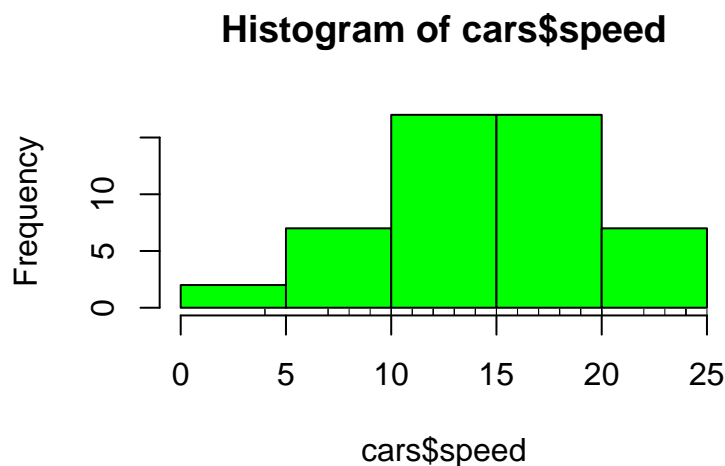
## GENERAR SERIES ALEATORIAS

Para muestra y ejemplos muchas veces necesitamos generar una serie de datos aleatoria rápida. Damos aquí una pequeña muestra de funciones útiles para este fin:

```
# genera una serie de numeros del 1 al 12 en orden aleatorio y sin repetir
  sample(1:12)
# repite 1,2,3,4,5 ---- 3 veces
  rep(c(1:5),3)
# repite cero, 5 veces y tres, 5 veces --> 0 0 0 0 0 3 3 3 3 3
  rep(c(0,3),each=5)
#Generamos datos aleatorios
  x<-1:100      # serie de numeros de 1 a 100
# serie de 100 números aleatorios normales de media 0 y sd=5
  y<-rnorm(100,0,5)
# Creamos undata frame
  df<-data.frame(x,y)
```

## HISTOGRAMAS

```
# GRAFICAS DE HISTOGRAMAS
  hist(cars$speed, col="green")
# añadiendo rug para ver marcas de datos
  rug(cars$speed)
```



**Figura 1:** Histogramas base system



```
#Otro
hist(y, col="tomato", breaks=5)
rug(y)
# añadimos linea vertical
abline(v= 1, lwd=2) # grosor 2)
# añadimos linea vertical en la media
abline(v= median(y), col= "navy", lwd=4)
```

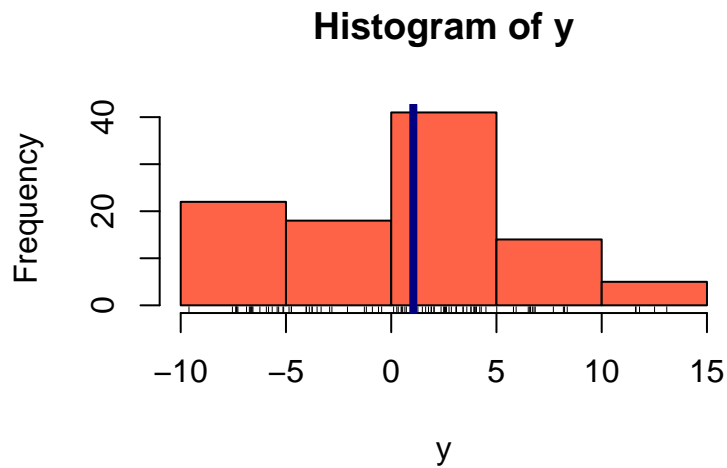
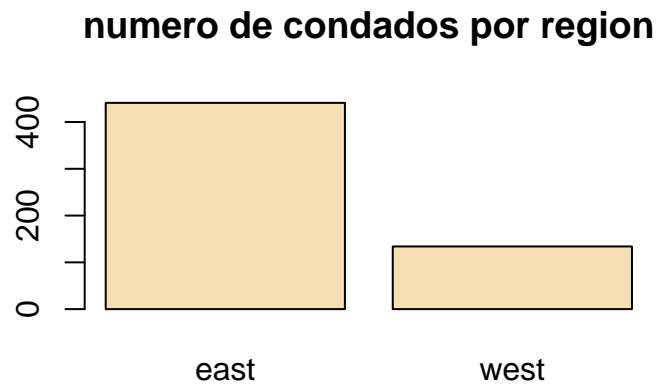


Figura 2: Histogramas base system

```
# lwd=line_width=grosor 2)
# lty=tipo linea

barplot(table(pollution$region), col="wheat", main="numero de condados por region")
```

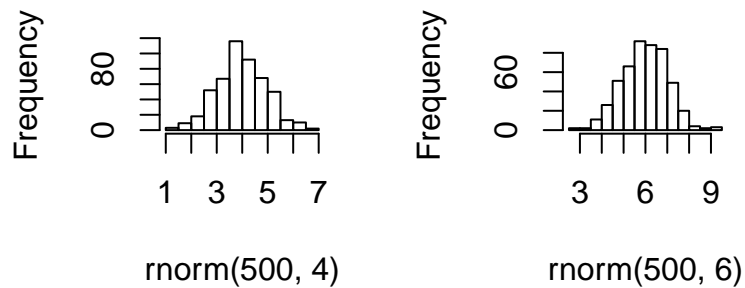


**Figura 3:** Histogramas base system

```
set.seed(42)
par(mfrow=c(1,2))
p1 <- hist(rnorm(500,4))
p2 <- hist(rnorm(500,6))
```

# una fila con dos col de graficos  
# centrado en 4  
# centrado en 6

### Histogram of rnorm(500, 4)      Histogram of rnorm(500, 6)



**Figura 4:** Histogramas

```
plot( p1, col=rgb(0,0,1,1/4), xlim=c(0,10)) # primer histograma

plot( p1, col=rgb(0,0,1,1/4), xlim=c(0,10)) # repetimos para la segunda grafica
plot( p2, col=rgb(1,0,0,1/4), xlim=c(0,10), add=T) # ojo con add añadimos al mismo grafico
```



## Histogram of rnorm(50)Histogram of rnorm(50

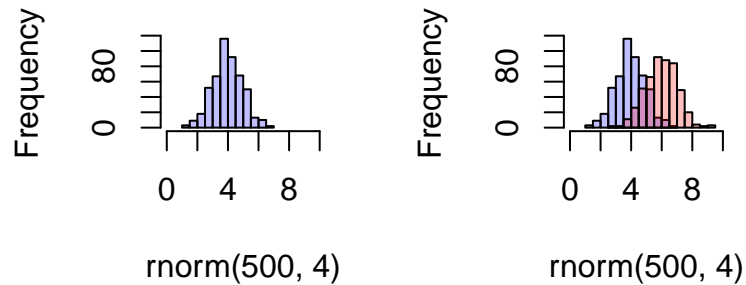


Figura 5: Histogramas

```
dev.off()

## null device
##          1
```

## Histograma con ggplot2

```
library(ggplot2)

ggplot(df, aes(y)) +
  geom_histogram(aes(fill=..count..)) +
  scale_fill_gradient("Count", low = "green", high = "red")+
  geom_density(position = "stack")
```

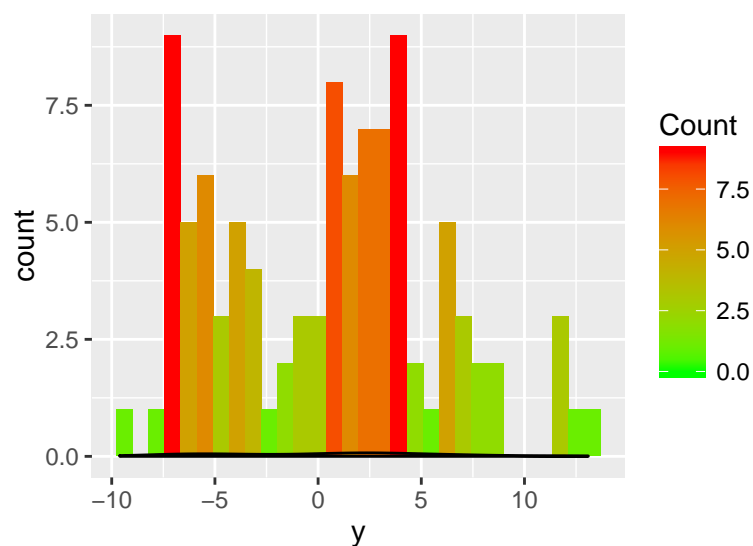


Figura 6: Histogramas ggplot





```
# funcion de densidad
ggplot(df, aes(x=y)) +
  geom_histogram(aes(y = ..density.., fill=..count..)) +
  geom_density(lwd=1,adjust = 1/2)
```

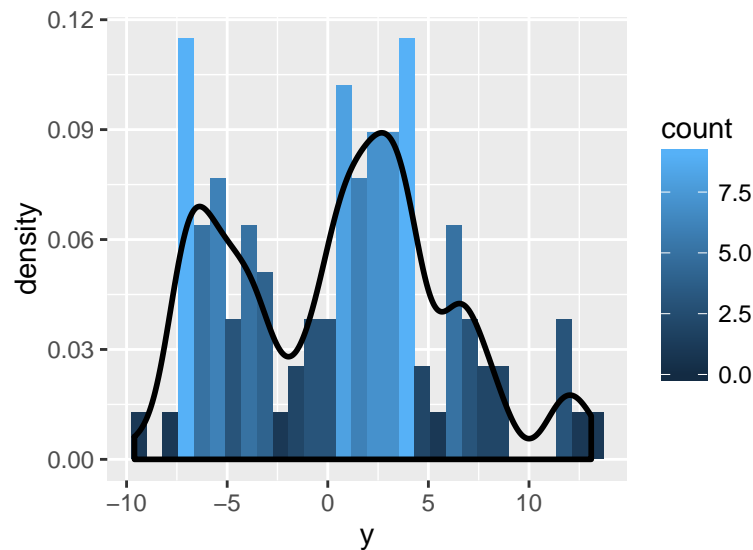


Figura 7: Histogramas ggplot

## BOX WHISPER

Para los graficos de cajas es necesario si queremos hacer categorías convertir la variable de categoría a factor.

```
par(mar=c(3,3,0,2))#mar=c(bottom, left, top, right)
boxplot(cars$speed, col="lightblue", horizontal = TRUE)
#si queremos añadir una linea al boxplot, por ejemplo para marcar el limite de 12
abline(v=median(cars$speed), col= "red", lwd=1)
```

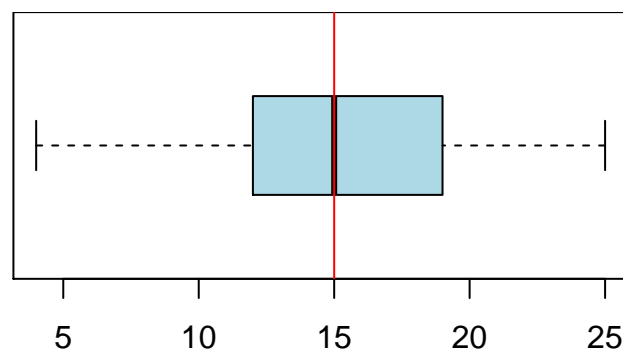


Figura 8: Graficos de caja Box-Whisper



```
# Grafica de cajas boxplot
airquality<-transform(airquality,Month=factor(Month))
boxplot(Ozone ~ Month, airquality, xlab="mes", ylab="ozono ppb",
        col=c("slategray","sienna","seagreen","tomato2","royalblue3"))
```

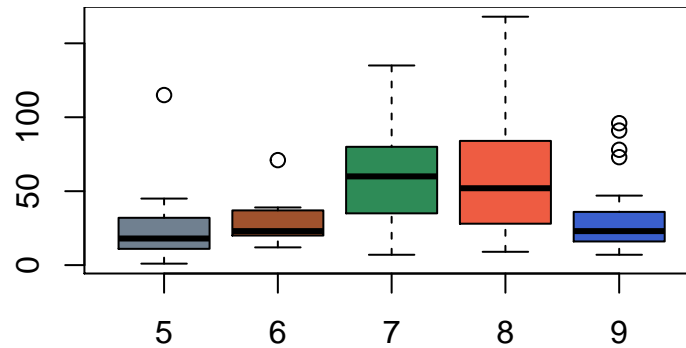


Figura 9: Graficos de caja Box-Whisper

## DISPERSIÓN x-y

```
library(datasets)
# head(cars)
# pintamos de la datatable cars, dos variables
par(mar=c(3,3,3,2))#mar=c(bottom, left, top, right)
with(cars, plot(speed,dist))
```

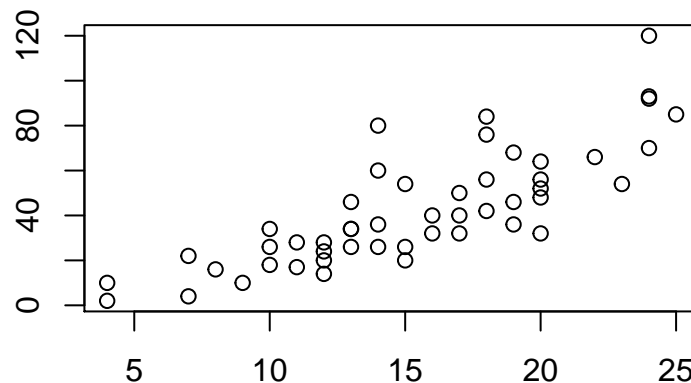


Figura 10: Graficos de dispersion

```
# lo mismo con conlor transparente 0.3 : cex cambia las fuentes multiplica escala
with(cars, plot(speed,dist,col=rgb(0,.5,.5,0.3),pch=19, cex=0.8))
```



```
# Añadimos un texto al grafico
# lo situamos en la mitad (media), parte alta
  text(mean(cars$speed),max(cars$dist),"aquí va")
# añadimos etiquetas a cada punto
  with(cars, text(speed,dist, labels = dist,pos=4))#row.names(cars), pos = 4))
```

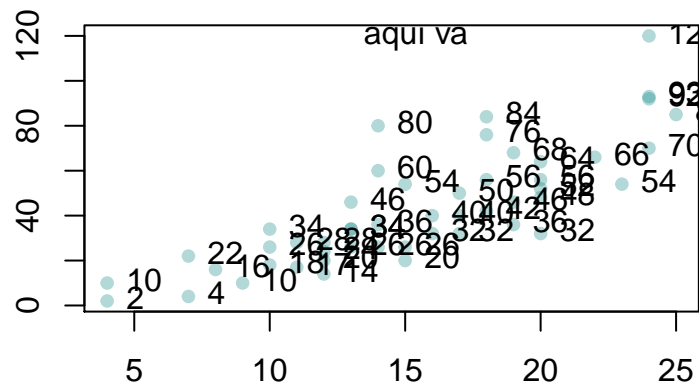


Figura 11: Graficos de dispersion

```
# Añadir label de datos
#text(x = speed, y = dist, labels = rownames(dataFrame), pos=4, col="red")
#-----otra -----
with(pollution,plot(latitude,pm25, col=region ))
  abline(h=12,lwd=2,lty=2)
```

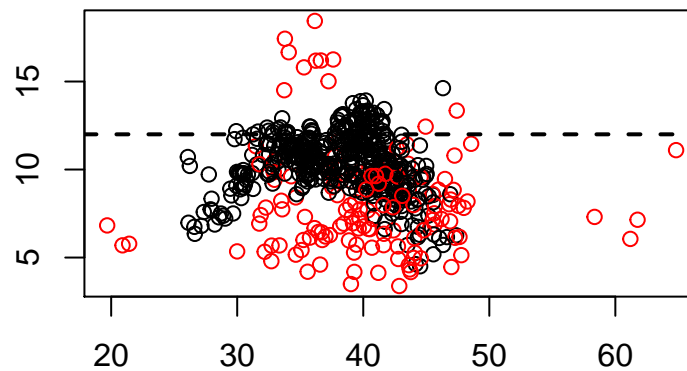


Figura 12: Graficos de dispersion



```
with(airquality,plot(Wind,Ozone,type = "n"))
  may <- subset(airquality, Month==5)
#pintamos ese conjunto
  points(may$Wind,may$Ozone,col="blue",pch=17)
# ahora los meses que no son mayo
  notmay <- subset(airquality, Month!=5)
  points(notmay$Wind,notmay$Ozone,col="red",pch=8)
#añadimos la legenda
  legend("topright",pch=c(17,8),col=c("blue","red"),legend=c("May", "Other Months"))
#añadimos un titulo:
  title(main="Ejemplo de grafico x-y")
  mtext("Ozone y viento en New York",outer=TRUE)
```

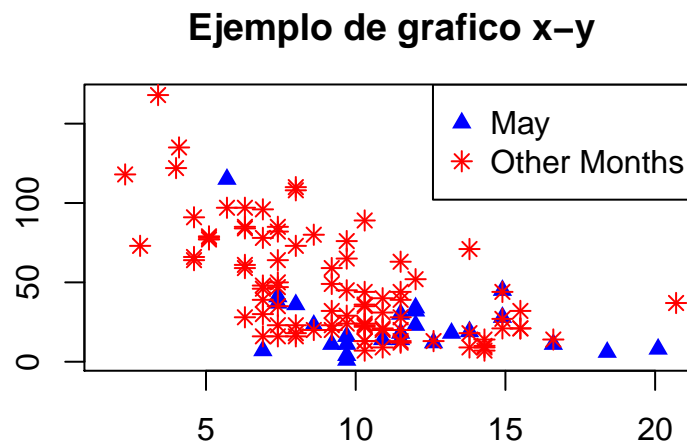


Figura 13: Graficos de dispersion

## X~Y CON MODELOS DE REGRESION

```
with(airquality, plot(Wind,Ozone, main="Ozono y viento en NY",pch=1))
# Añado que pinte los del mes de mayo en azul
with(subset(airquality, Month==5), points(Wind,Ozone,col="blue",pch=3))
# Añadimos una leyenda
legend("topright", pch =c(1,3), col=c("blue","black"),legend=c("Mayo","otros"))
#añadimos linea de regresion
model<-lm(Ozone ~ Wind,airquality)
abline(model,lwd=2)
```

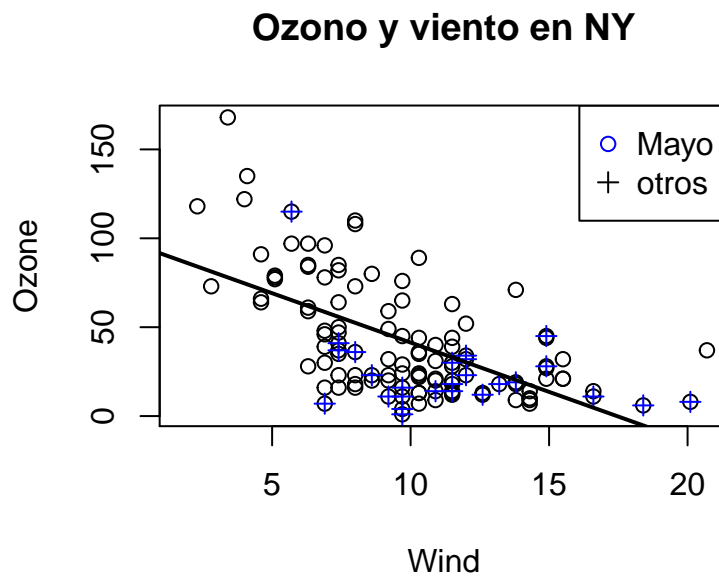


Figura 14: Graficos de lineas con modelo de regresion

## GRÁFICOS DE LINEAS

```
# Graficos de lineas
par(mar=c(3,3,0,2))#mar=c(bottom, left, top, right)
# cumsum es suma acumulada de la serie
plot(df$x, cumsum(df$y), ylab="eje y", xlab=" Eje X", type="n") # type n es que no lo pinta
lines(df$x, cumsum(df$y), type= "l", col="grey")
lines(df$x, cumsum(df$y)+10, type= "l", col="red")
lines(df$x, cumsum(df$y)-20, type= "l", col="blue")
legend("topright", pch =20, col=c("grey", "red", "blue"), legend=c("gris", "rojo", "azul"))
```

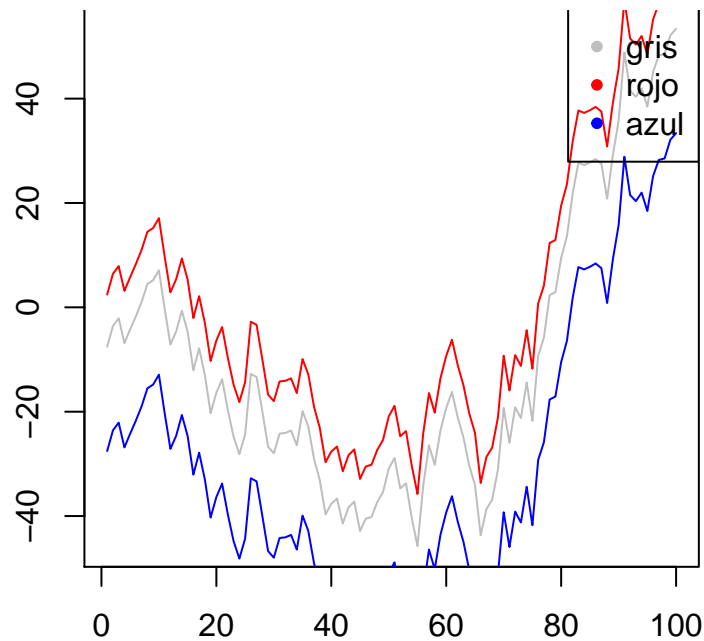
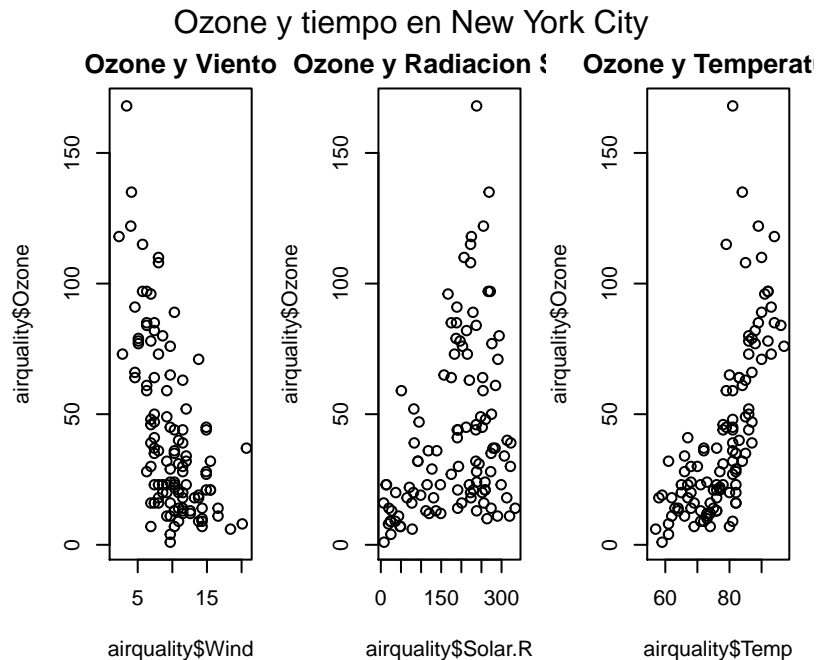


Figura 15: Graficos de lineas

## MULTIGRÁFICAS

```
# MULTIGRÁFICAS
library(datasets)
# grafico de 3 graficas
par(mfrow = c(1, 3), mar = c(4, 4, 2, 1), oma = c(0, 0, 2, 0))
plot(airquality$Wind, airquality$Ozone, main="Ozone y Viento")
plot(airquality$Solar.R, airquality$Ozone, main="Ozone y Radiacion Solar")
plot(airquality$Temp, airquality$Ozone, main="Ozone y Temperatura")
mtext("Ozone y tiempo en New York City", outer=TRUE)
```



## PALETA DE COLOR PERSONALIZADO

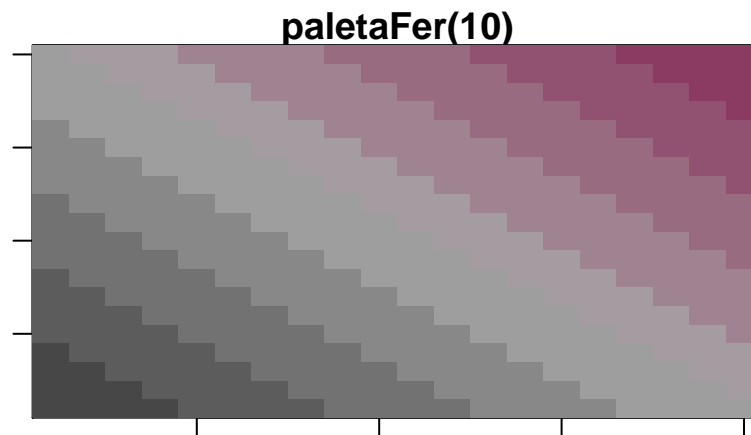
Los colores con nombre son los almacenados en el vector `colors()`, hay 627. Las paletas predefinidas están en `grDevices` color palette existen unas cuantas como: *cm.colors*, *topo.colors*, *terrain.colors*, *heat.colors*, *rainbow*.

```
# generar 10 colores aleatorios de la muestra colors
sample(colors(),10)

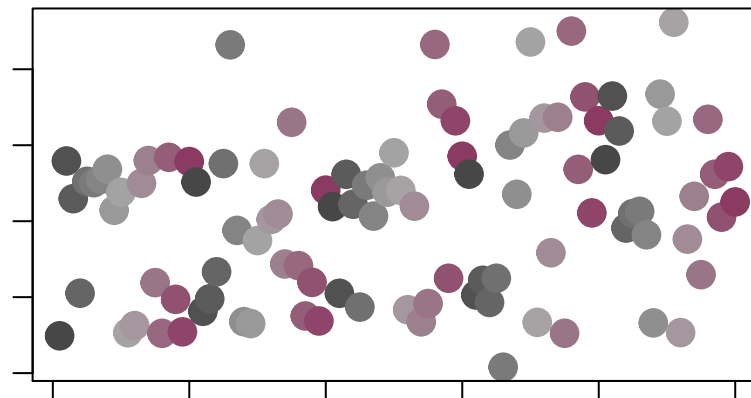
## [1] "whitesmoke"    "grey27"        "maroon4"       "seashell3"
## [5] "plum4"         "mistyrose"     "lightpink3"    "purple1"
## [9] "darkseagreen2" "gray83"

par(mar = c(1, 1, 1, 1))
# FUNCION para ver las escalas en un gráfico de color
verEscalas<-function(cv){
  myarg <- deparse(substitute(cv))
  z<- outer( 1:20,1:20, "+")
  obj<- list( x=1:20,y=1:20,z=z )
  image(obj, col=cv, main=myarg )
}

# Generamos una paleta personalizada aleatoria a partir de 3 colores
coloresorigen<-sample(colors(),3)
paletaFer<-colorRampPalette(coloresorigen,alpha=0.3)
# vemos la paleta
verEscalas(paletaFer(10))
```

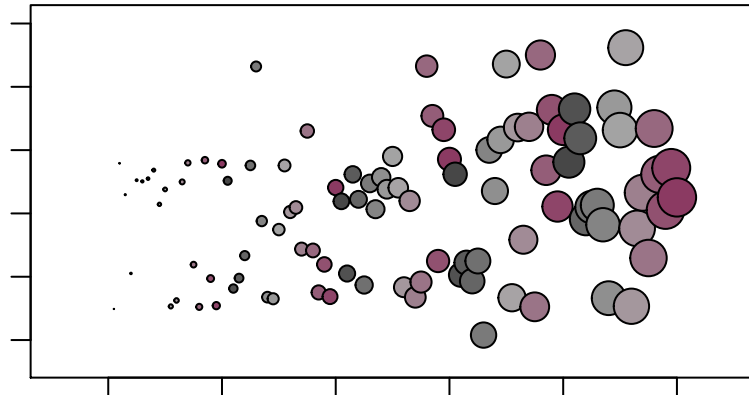


```
# Ejemplo de uso de la paleta en un gráfico  
plot(x,y,col=paletaFer(20),pch=20,lwd=10)
```



```
#cambiando tamaño de los puntos según su valor  
colorfondo<-paletaFer(20)[3]  
with(df,  
  symbols(x=x, y=y, circles=x, inches=1/10,ann=F,  
    bg=colorfondo, fg="black")) # fg="NULL"
```





```
#para generar colores a partir de una de las paletas persnalizadas:  
rainbow(3) # genera 3 colores de la paleta rainbow
```

```
## [1] "#FF0000FF" "#00FF00FF" "#0000FFFF"
```

```
terrain.colors(5) # genera 5 colores de la paleta terrain.colors
```

```
## [1] "#00A600FF" "#E6E600FF" "#EAB64EFF" "#EEB99FFF" "#F2F2F2FF"
```

```
topo.colors(4, alpha = 0.5) # # genera 5 colores de la paleta topo.colors con alfa .5
```

```
## [1] "#4C00FF80" "#00E5FF80" "#00FF4D80" "#FFFF0080"
```

```
#heat.colors
```

## Colores con ggplot2:

```
# Uso del color en ggplot  
library(ggplot2)  
par(mar = c(4, 4, 3, 1))  
qplot(x, y, data = df, colour=y) +  
  scale_colour_gradientn(colours=colorfondo)# cambia el color de gradiente
```

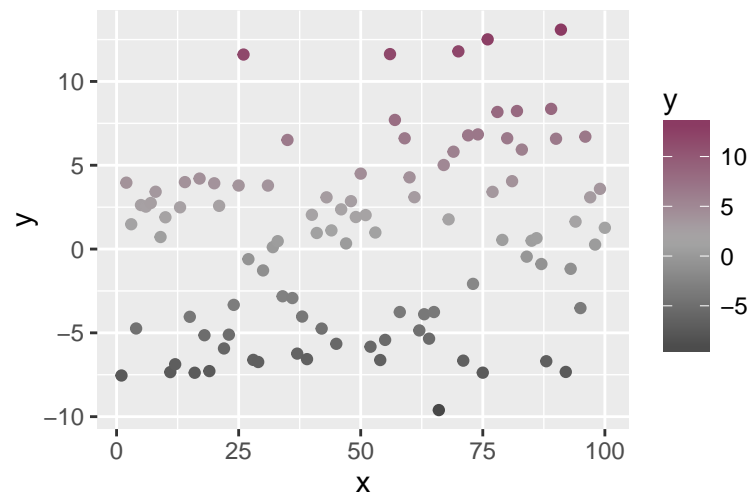


Figura 16: Uso del color en ggplot

```
ggplot(df, aes(y)) +  
  geom_histogram(aes(fill=..count..)) +  
  scale_colour_gradientn(colours = terrain.colors(8))
```

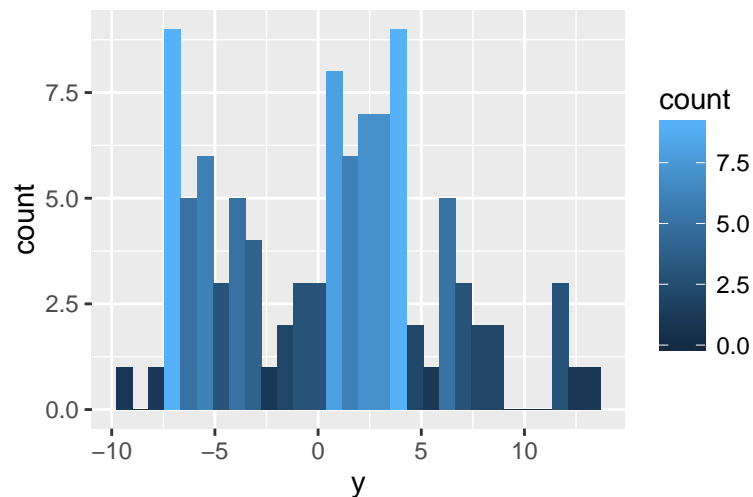


Figura 17: Uso del color en ggplot

## PINTAR EN FICHERO

### PDF

```
# pdf  
pdf(file = "myplot.pdf") # abrimos el dispositivo gráfico  
# creamos un grafico y lo mandamos al dispositivo pdf
```



```
with(faithful,plot(eruptions,waiting))
title(main="Datos de daños Geyser") # sobreescribimos el título
dev.off()# cerramos el dispositivo, para volver a dejar la pantalla por defecto
```

## PNG

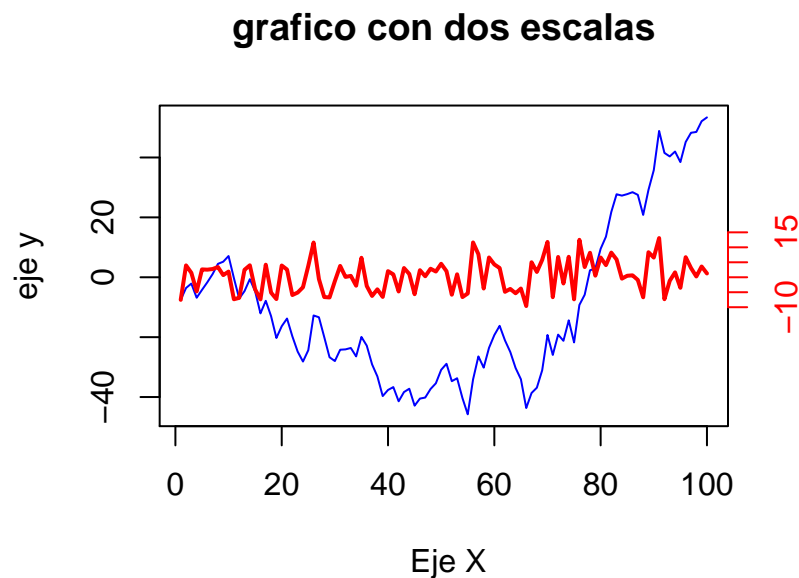
```
png('plot1.png',width = 480, height = 480, units = "px")
hist(df$y, xlab="eje X", main="titulo del grafico", col="red")
dev.off()
```

## VARIAS ESCALAS

Para pintar un gráfico con dos escalas distintas:

```
# grafico con dos escalas distintas
df$z<-df$y*10 # añadido columna nueva a la df

with(df,plot(x, cumsum(y), ylab="eje y", xlab=" Eje X", type="n"))
# type n es que no lo pinta
lines(x, cumsum(y), type= "l", col="blue")
title("grafico con dos escalas")
# añadido otro grafico
par(new = T)
lines(df$x, y, type= "l",lwd=2, col="red", xlab=NA, ylab=NA)
axis(side=4,xlab="eje 2", at = pretty(range(y)),col="red",col.axis = "red")
```



**Figura 18:** grafico con dos escalas distintas



```
# at puede omitirse y se calcula
```