
Curso OpenCourseWare

**Aprendizaje del Software Estadístico R: un entorno
para simulación y computación estadística**

Alberto Muñoz García

8. Introducción a los gráficos en R



Introducción

Los gráficos disponibles en R son de gran calidad y de una versatilidad impresionante.

Para hacernos una idea, podemos ejecutar la demo del programa mediante:

```
demo("graphics")
```

Para poder crear gráficos es necesario inicializar un dispositivo gráfico (graphic device), y si hay varios, elegir uno de ellos.

win.graph() x11() windows()	Crea una ventana para gráficos (graphic device) y la deja como dispositivo activo, al cual se enviarán todos los dibujos.
--	---

Para eliminar la ventana gráfica actual:

```
dev.off()
```

En el caso de que haya varias ventanas gráficas, podemos cambiar entre ellas utilizando el comando `dev.set()`. Un ejemplo:

```
> win.graph() # Crea una ventana gráfica (dispositivo 2)
> win.graph() # Crea otra ventana gráfica (dispositivo 3): ésta es ahora la
ventana activa
> dev.set(2) # Hace que la ventana activa sea la dos.
X11
2
```

Para eliminar una ventana gráfica:

```
dev.off(3) # Elimina la ventana número 3. La ventana 2 queda como ventana
activa
```

Gráficos de alto nivel

Entre las funciones de alto nivel para dibujar, podemos destacar por su frecuencia de uso las siguientes:

barplot Diagrama de barras

boxplot Diagrama de caja

contour Gráfico de contorno

hist Histograma

pairs Pares de gráficos de dispersión por variables

persp gráficos 3D

plot Rutina de dibujo general

Veamos algunos ejemplos:

```
> data(VADeaths)
```

```
> VADdeaths
```

```
Rural Male Rural Female Urban Male Urban Female
```

```
50-54 11.7 8.7 15.4 8.4
```

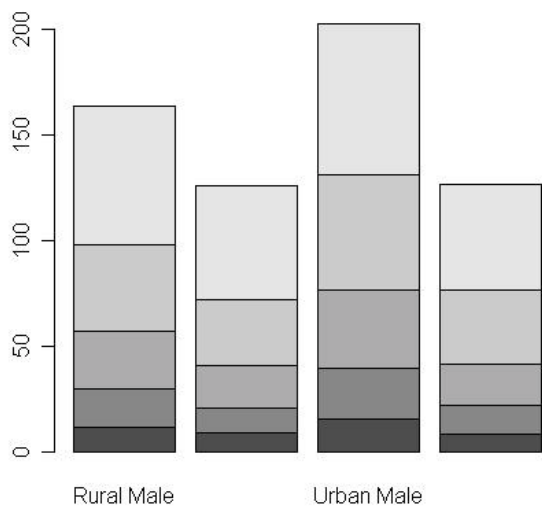
```
55-59 18.1 11.7 24.3 13.6
```

```
60-64 26.9 20.3 37.0 19.3
```

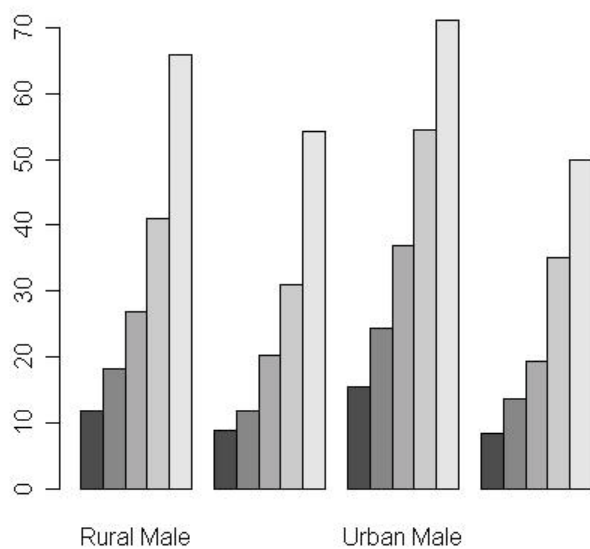
```
65-69 41.0 30.9 54.6 35.1
```

```
70-74 66.0 54.3 71.1 50.0
```

```
> barplot(VADeaths)
```



> barplot(VADdeaths, beside=T)



De ese modo obtendremos diagramas de barras por grupos de edad para cada tipo de población.

> x = rnorm(200) # genera 200 datos de una distribución normal estandarizada

> boxplot(x) # dibuja un diagrama de caja para mostrar la forma de la distribución

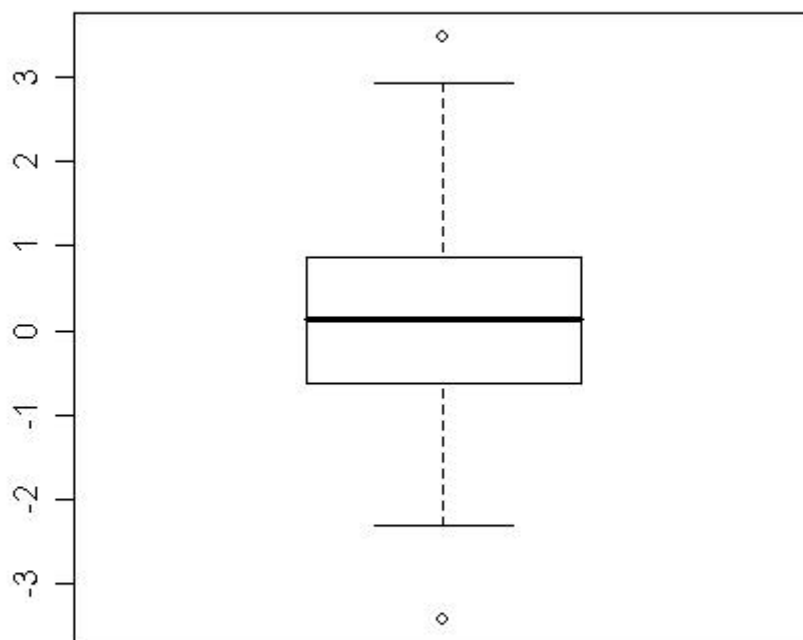
Pero también podemos dibujar diagramas de caja para matrices, uno por cada columna (variable) de la matriz. Obsérvese la diferencia entre las dos órdenes siguientes:

> **boxplot(deaths)** # Trata todos los datos como una sola variable

> **boxplot(data.frame(deaths))** # Un diagrama de caja por cada categoría

> **x = rnorm(1000)** # Genera 1000 datos de una normal

> **hist(x)** # Histograma de los 1000 datos



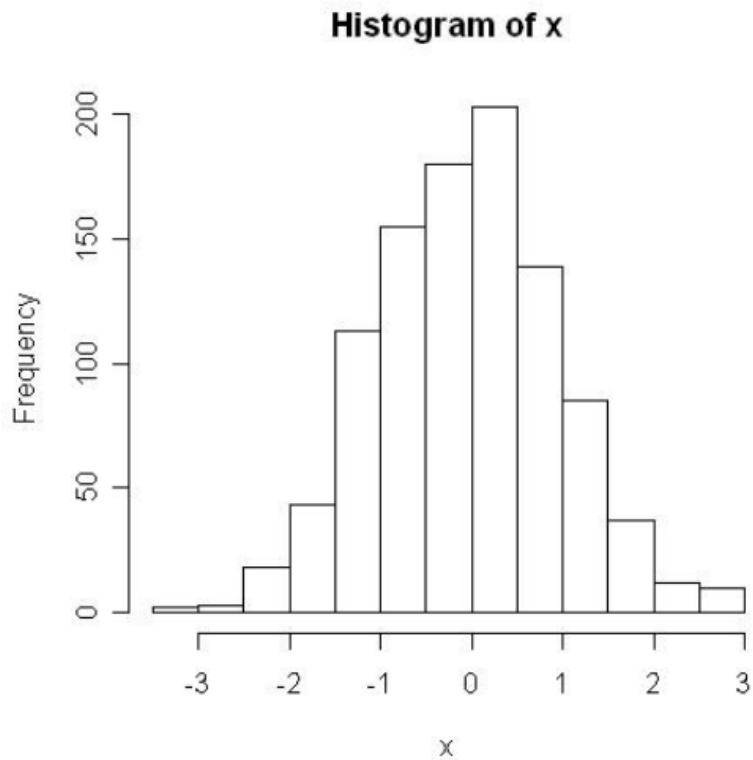
Pero también podemos dibujar diagramas de caja para matrices, uno por cada columna (variable) de la matriz. Obsérvese la diferencia entre las dos órdenes siguientes:

> **boxplot(deaths)** # Trata todos los datos como una sola variable

> **boxplot(data.frame(deaths))** # Un diagrama de caja por cada categoría

> **x = rnorm(1000)** # Genera 1000 datos de una normal

> **hist(x)** # Histograma de los 1000 datos



Vamos a ver un ejemplo de la función pairs sobre el conjunto de datos iris. Este conjunto recoge una matriz de 150 observaciones con 4 variables cada una, relativas a la morfología de varias especies ornamentales. Lo que vamos a hacer a continuación es cargar el conjunto de datos iris en la memoria, mostrar cuatro registros, y graficar las variables dos a dos.

```
> data(iris) # Cargamos el conjunto de datos iris
```

```
> iris[1:4,] # Mostramos 4 registros
```

```
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
```

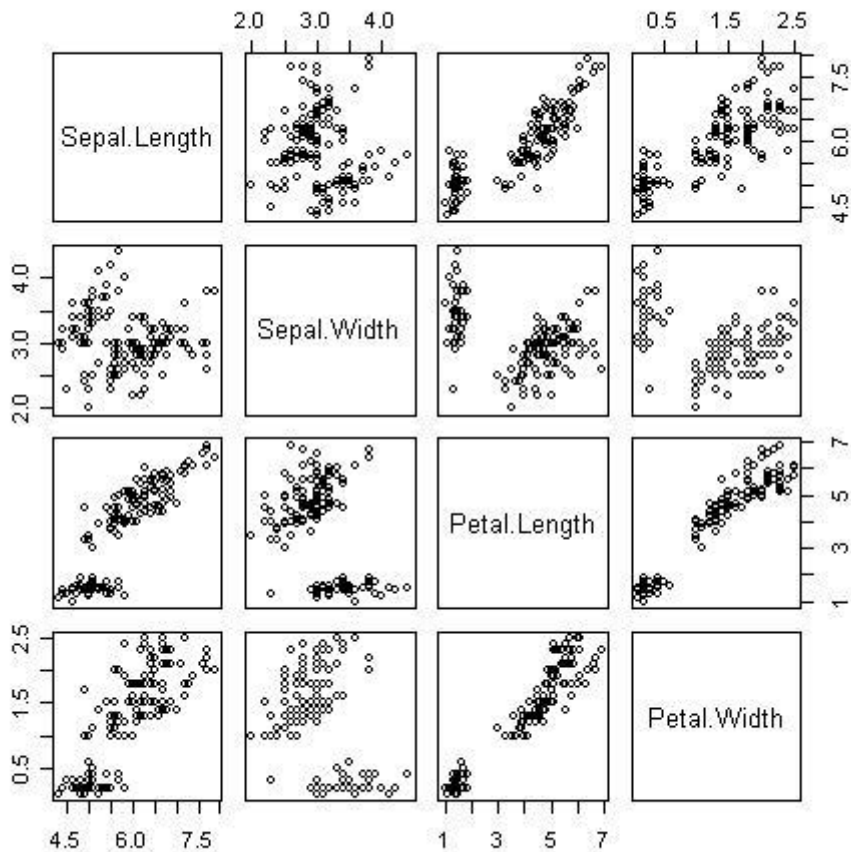
```
1 5.1 3.5 1.4 0.2 setosa
```

```
2 4.9 3.0 1.4 0.2 setosa
```

```
3 4.7 3.2 1.3 0.2 setosa
```

```
4 4.6 3.1 1.5 0.2 setosa
```

```
> pairs(iris[,1:4]) # Mostramos los gráficos de dispersion de las variables
```

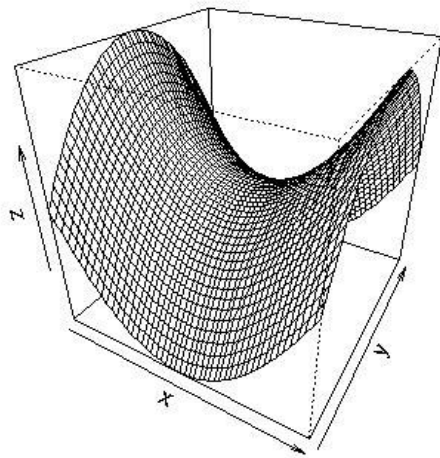


Ejemplo de dibujo de una función $z=f(x,y)$:

```

> x = seq(-10,10,length=50)    # Generamos una malla de puntos (x,y)
> y = x
> f = function(x,y) { x^2 - y^2 } # Definimos la función que dibujaremos
> z = outer(x,y,f)    # La función outer evalua la función f en cada
punto(xi,yj)
> persp(x,y,z)        # Un gráfico en perspectiva
> persp(x,y,z,theta=30,phi=30) # Otro

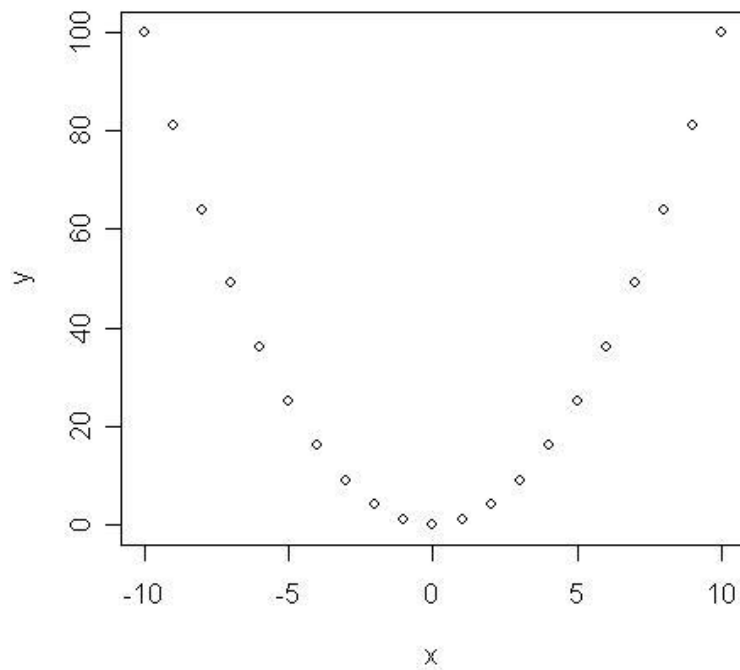
```



La función plot()

El comando **plot** se utiliza para crear una nueva figura.

```
> x = seq(-10,10) # Generamos los números -10, -9, ..., 9, 10  
> y = x^2        # Generamos los cuadrados de dichos números  
> plot(x,y)     # Graficamos
```

axes=F Suprime la generación de los ejes

log="x" Hace que alguno de los ejes se tome en escala logarítmica

log="y"

log="xy"

type="p" Dibuja puntos individuales (opción por defecto)

type="l" Dibuja líneas

type="b" Dibuja puntos y líneas

type="o" Dibuja puntos atravesados por líneas

type="h" Dibuja con líneas verticales

type="s" Dibuja a base de funciones escalera

type="S" Casi lo mismo

type="n" No dibuja nada. Pero deja marcados los puntos para manejos posteriores

`xlab="cadena"` Etiqueta para el eje de las x

`ylab="cadena"` Etiqueta para el eje de las y

`main="cadena"` Título del gráfico

`sub="cadena"` Subtítulo del gráfico

`pch="simbolo"` Se dibuja con el simbolo especificado. Por ejemplo:

`pch=18`

`pch="x"`

`pch="P"`

`col= numero entero` Color para dibujar

`col=2` Color rojo

`col=3` Color verde

Algunos ejemplos:

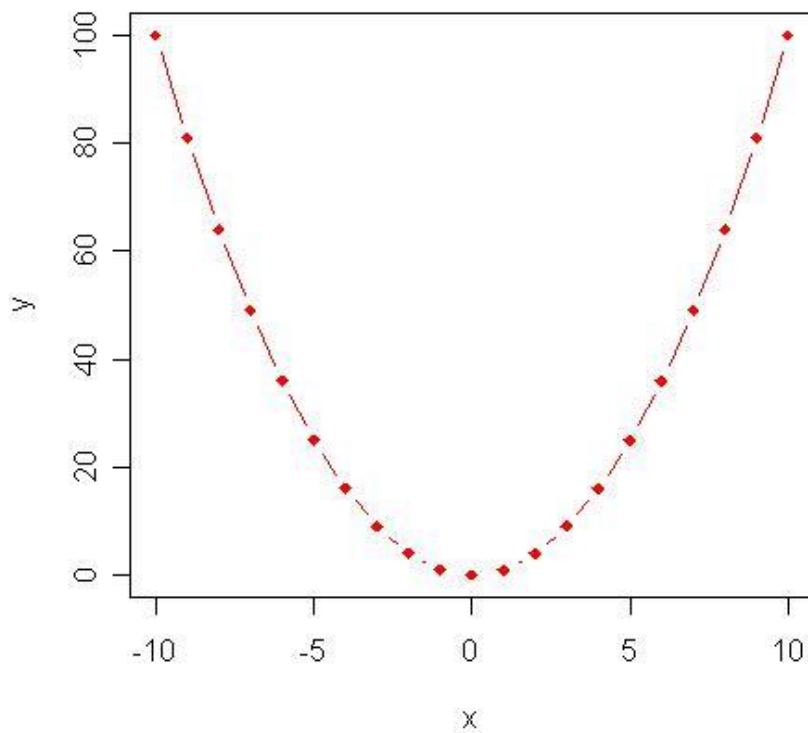
```
x = seq(-10,10)
```

```
y = x^2
```

```
plot(x,y,type="l",xlab="eje de las x",ylab="eje de las y",main="Parabola")
```

```
plot(x,y,type="h",xlab="eje de las x",ylab="eje de las y",main="Parabola",axes=F)
```

```
plot(x,y,pch=18,col=2,type="b")
```



Gráficos de bajo nivel

Los comandos de bajo nivel sirven para añadir información extra a los gráficos que producen los comandos de alto nivel. Por ejemplo, podríamos querer añadir texto a un gráfico, puntos extras, líneas, cosas así. Entre los más importantes podemos destacar:

xlab="cadena" Etiqueta para el eje de las x

ylab="cadena" Etiqueta para el eje de las y

main="cadena" Título del gráfico

sub="cadena" Subtítulo del gráfico

points(x,y) Añade puntos o líneas conectadas al gráfico actual

lines(x,y)

text(x,y,etiquetas) Añade texto al gráfico actual en la posición x,y

abline(a,b) Añade una línea de pendiente a y que corta al origen en b

abline(h=y) Añade línea horizontal que corta al eje y en h=y

abline(v=x) Lo análogo para línea vertical

polygon(x,y) Dibuja un polígono

title(main,sub) Añade título y subtítulo al gráfico actual

axis(side) Añade ejes al gráfico actual (de 1 a 4)

Algunos ejemplos:

```
> x = seq(-10,10)
```

```
> y = x^2
```

```
> plot(x,y,axes=F)
```

```
> axis(1)
```

```
> x = seq(-10,10)
```

```
> y = x^2
```

```
> plot(x,y)
```

```
> abline(h=40,col=3)
```

```
> abline(v=0,col=4)
```

```
> text(-5,60,"grafico raro")
```

```
> x = rnorm(200)
```

```
> plot(x)
```

```
> abline(h=0,col=2)
```

Varios gráficos en una misma ventana

Para conseguir esto se usa el comando `par`, con la opción `mfrow`. Ejemplos:

```
par(mfrow=c(1,1)) # un solo gráfico por ventana: la opción por defecto  
par(mfrow=c(2,1)) # Dibuja una matriz de gráficos 2x1: un gráfico debajo de otro  
par(mfrow=c(2,3)) # Matriz de gráficos 2 x 3 : dos filas por tres columnas
```

Un ejemplo:

```
> x = rnorm(200) # Se generan 200 valores de una normal estandarizada  
> par(mfrow=c(2,2)) # Se crea una matriz de gráficos 2 x 2  
> plot(x) # Dibujo de x frente al índice 1 a 200  
> hist(x) # Histograma de x  
> boxplot(x) # Diagrama de caja de x  
> qqnorm(x) # Gráfico cuantil-cuantil de x frente a la distribución normal
```

