

# Notas sobre R

Francesc Carmona, Jordi Ocaña i Alex Sánchez

Department d'Estadística  
Universitat de Barcelona

23 de gener de 2008

# Outline

## Introducción

Instrucciones de alto nivel

Ajuste de parámetros

Más ejemplos

Identificación de puntos en la región gráfica

Graficos Trellis

## Graficos

# Introducción

- ▶ R tiene grandes posibilidades para hacer gráficos.
  - ▶ `demo(graphics)`
  - ▶ <http://addictedtor.free.fr/graphiques>
- ▶ Idea: la visualización es parte integral del análisis y la comunicación de los resultados
- ▶ Los gráficos deben ser en primer lugar *estadísticos*.
- ▶ También pueden ser *estéticos*.

# Tipos de funciones gráficas en R

- ▶ Funciones de alto nivel
  - ▶ Permiten crear los gráficos básicos
  - ▶ `plot`, `hist`, `boxplot`, `pairs`,...
- ▶ Funciones de bajo nivel
  - ▶ Permiten modificar los gráficos creados
  - ▶ `points`, `lines`, `text`, `axis`,...
- ▶ Gráficos Trellis: Permiten describir situaciones complejas, a menudo multivariantes, con un sólo gráfico organizado en paneles.

# Los dispositivos gráficos

- ▶ Cuando creamos un gráfico en R dirigimos la salida hacia un *dispositivo* o “driver”.
  - ▶ Por defecto es la pantalla.
  - ▶ Hay muchos formatos disponibles:  
'postscript', 'pdf', 'png', 'jpeg', 'bmp', ...
- ▶ La utilización de los distintos drivers aumenta la flexibilidad y potencia de R
  - ▶ Podemos crear gráficos de gran calidad, de manera automática.
  - ▶ Podemos crear grandes cantidades de gráficos mediante scripts para tratamientos masivos de datos.

# Los pasos para crear un gráfico

- ▶ Una forma típica de crear gráficos puede ser:
  - ▶ Llamar el driver hacia el que se dirigirá la salida
  - ▶ Establecer los parámetros por defecto (*Ajustes previos* al gráfico)
  - ▶ Crear el gráfico utilizando funciones de alto nivel (*Ajustes al crear* el gráfico)
  - ▶ Modificar el gráfico con funciones de bajo nivel (*Ajustes posteriores* al gráfico)
  - ▶ Restablecer los parámetros
  - ▶ Cerrar el dispositivo de salida.

# Outline

Introducción

**Instrucciones de alto nivel**

Ajuste de parámetros

Más ejemplos

Identificación de puntos en la región gráfica

Graficos Trellis

## Graficos

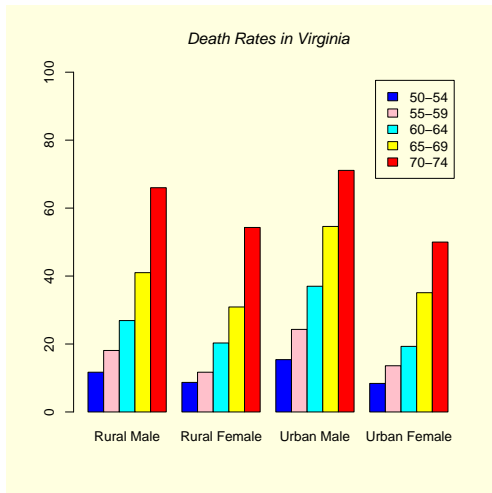
## plot() y compañía

- ▶ La instrucción básica para crear un gráfico es `plot()`
- ▶ Se trata de una *función genérica*: Al ser R un lenguaje orientado a objetos el resultado de la función será diferente según sobre que tipo de datos se aplique.

```
> opt <- par(bg = "lightyellow", mfrow = c(1, 1))
> barplot(VADeaths, beside = T, col = c("blue", "pink",
+   "yellow", "red"), legend = rownames(VADeaths), ylab = "Deaths",
+   100))
> title(main = "Death Rates in Virginia", font.main = 3)
> par(opt)
```



# Ejemplo



# Outline

Introducción

Instrucciones de alto nivel

**Ajuste de parámetros**

Más ejemplos

Identificación de puntos en la región gráfica

Graficos Trellis

## Graficos

## Ajustes previos al gráfico: `par()`

- ▶ La instrucción `par()` permite establecer un grandísimo número de opciones gráficas (`? par`)
- ▶ A diferencia de otros programas, muchos aspectos de configuración deben establecerse *antes* de crear el gráfico.
- ▶ Las opciones deben restaurarse después de su utilización

```
> library(DAAG)
```

```
> attach(elasticband)
```

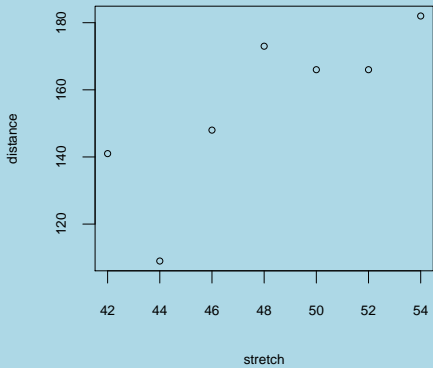
```
> oldpar <- par(cex.main = 1.5, mex = 1.5, bg = "lightblue")
```

```
> plot(distance ~ stretch)
```

```
> par(oldpar)
```

```
> detach(elasticband)
```

```
> detach(package:DAAG)
```



## Ajustes al crear el gráfico

- ▶ La mayoría de instrucciones admiten un número mínimo de parámetros para definir cosas como el título (`main`) o las etiquetas (`xlab`, `ylab`)

```
> library(DAAG)
```

```
> attach(elasticband)
```

```
> oldpar <- par(cex.main = 1.5, mex = 1.5)
```

```
> plot(distance ~ stretch, main = "Análisis de elasticidad")
```

```
+   ylab = "Distancia", sub = "Tomado de 'Using R...'")
```

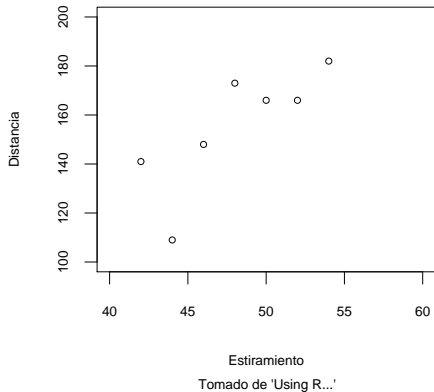
```
+   xlim = c(0, 100), ylim = c(100, 200))
```

```
> par(oldpar)
```

```
> detach(elasticband)
```

```
> detach(package:DAAG)
```

## Analisis de elasticidad



## Ajustes despues del gráfico: funciones de bajo nivel

- ▶ Una vez dibujado el gráfico podemos añadirle líneas o texto fácilmente. `ylab`)

```
> library(DAAG)
```

```
> attach(elasticband)
```

```
> oldpar <- par(cex = 1.5, mex = 1.5)
```

```
> plot(distance ~ stretch, main = "Análisis de elasticidad",  
+       ylab = "Distancia")
```

```
> abline(lm(distance ~ stretch))
```

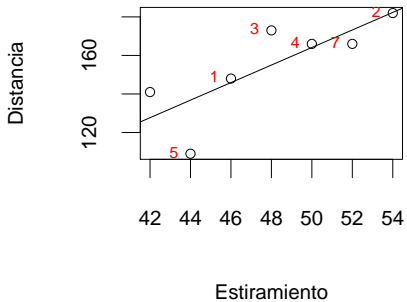
```
> text(stretch, distance, rownames(elasticband), pos = "right",  
+      col = "red")
```

```
> par(oldpar)
```

```
> detach(elasticband)
```

```
> detach(package:DAAG)
```

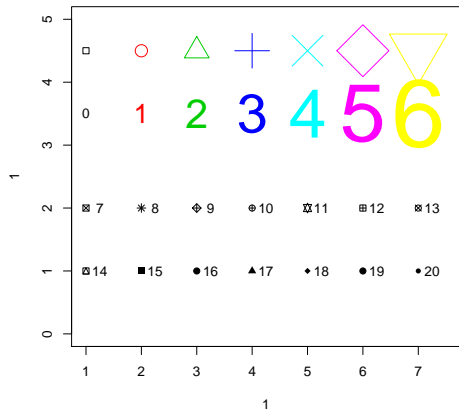
## Analisis de elasticidad





## Símbolos, colores y tamaños para los gráficos

```
> plot(1, 1, xlim = c(1, 7.5), ylim = c(0, 5), type = "n")
> points(1:7, rep(4.5, 7), cex = 1:7, col = 1:7, pch = 0:6)
> text(1:7, rep(3.5, 7), labels = paste(0:6), cex = 1:7, col = 1:7)
> points(1:7, rep(2, 7), pch = (0:6) + 7)
> text((1:7) + 0.25, rep(2, 7), paste((0:6) + 7))
> points(1:7, rep(1, 7), pch = (0:6) + 14)
> text((1:7) + 0.25, rep(1, 7), paste((0:6) + 14))
```

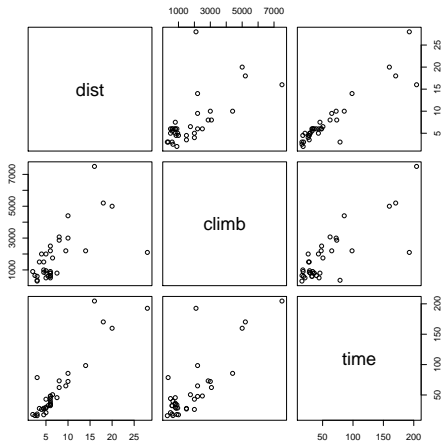


# Gráficos múltiples

## Scatterplots matriciales con `pairs()`

- ▶ Si los datos están en una matriz numérica cuyas columnas representan variables relacionadas podemos representarlas simultáneamente con la instrucción `pairs()`

```
> library(MASS)
> data(hills)
> pairs(hills)
> detach(package:MASS)
```

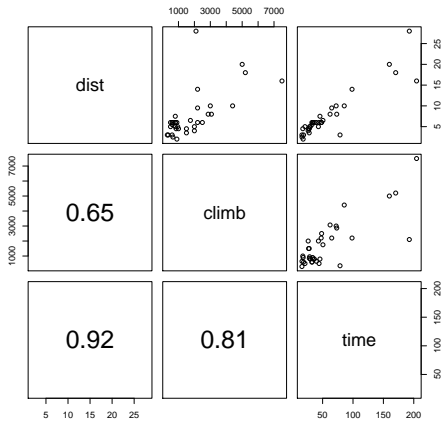


## Scatterplots matriciales panelados pairs()

- ▶ Si utilizamos un panel simétrico puede contener informaciones distintas.
- ▶ Para ello debemos crear una función y utilizarla mediante la función `panel()`

```
> panel.cor <- function(x, y, digits = 2, prefix = "",
+   usr <- par("usr")
+   on.exit(par(usr))
+   par(usr = c(0, 1, 0, 1))
+   r <- cor(x, y, use = "complete.obs")
+   txt <- format(c(r, 0.123456789), digits = digits)
+   txt <- paste(prefix, txt, sep = "")
+   if (missing(cex.cor))
+     cex <- 0.8/strwidth(txt)
+   text(0.5, 0.5, txt, cex = cex * 0.5)
+ }
> pairs(hills, main = "Correlation between genes (alls
+   lower.panel = panel.cor)
```

### Correlation between genes (alls spots)



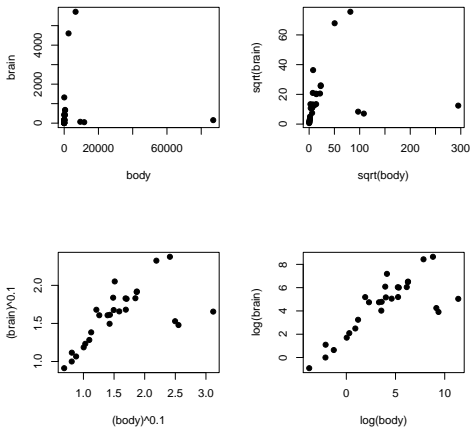
## Varios gráficos por pantalla

- ▶ Probablemente las opciones más utilizadas sean `mfrow()` y `mfcoll()`
- ▶ Permiten representar más de un gráfico por pantalla organizándolos por filas o columnas.

```
> library(MASS)
> attach(Animals)
> par(mfrow = c(2, 2), pch = 16)
> plot(body, brain)
> plot(sqrt(body), sqrt(brain))
> plot((body)^0.1, (brain)^0.1)
> plot(log(body), log(brain))
> par(mfrow = c(1, 1), pch = 1)
> mtext("Varias formas de representar la relacion peso
+     line = 2)
> detach(Animals)
> detach(package:MASS)
```



### Varias formas de representar la relacion peso altura



# Outline

Introducción

Instrucciones de alto nivel

Ajuste de parámetros

**Más ejemplos**

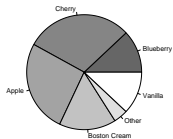
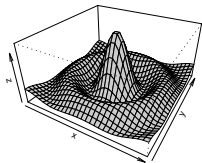
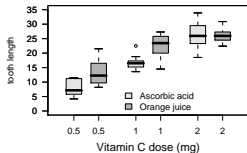
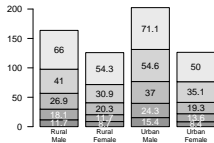
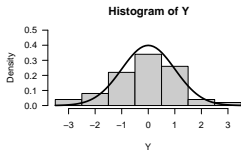
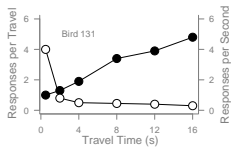
Identificación de puntos en la región gráfica

Graficos Trellis

## Graficos

## Más ejemplos

- ▶ La mejor forma de aprender a utilizar los parámetros es probando que efecto producen sobre los gráficos.
- ▶ Una buena fuente de ejemplos esta en la página web del libro “R graphics” de Paul Murrell <http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html>
- ▶ El código contenido en este script:  
<http://www.stat.auckland.ac.nz/~paul/RGraphics/examples-stdplots.R> realiza los gráficos de la página siguiente.



# Outline

Introducción

Instrucciones de alto nivel

Ajuste de parámetros

Más ejemplos

**Identificación de puntos en la región gráfica**

Graficos Trellis

## Graficos

- ▶ R permite una cierta interacción con interactuar con los gráficos en pantalla.
  - ▶ `identify()` Permite etiquetar puntos, colocando el cursor sobre ellos y haciendo un clic con el botón izquierdo.
  - ▶ `locate()` Permite obtener las coordenadas del punto, colocando el cursor sobre ellos y haciendo un clic con el botón izquierdo.
- ▶ La interacción finaliza si se alcanza el total de puntos solicitados o se pulsa el botón derecho.

```
library(UsingR);attach(florida)
plot(BUSH, BUCHANAN, xlab="Bush", ylab="Buchanan")
identify(BUSH, BUCHANAN, County)
locator()
detach(florida);detach(package:UsingR)
```

# Outline

Introducción

Instrucciones de alto nivel

Ajuste de parámetros

Más ejemplos

Identificación de puntos en la región gráfica

**Graficos Trellis**

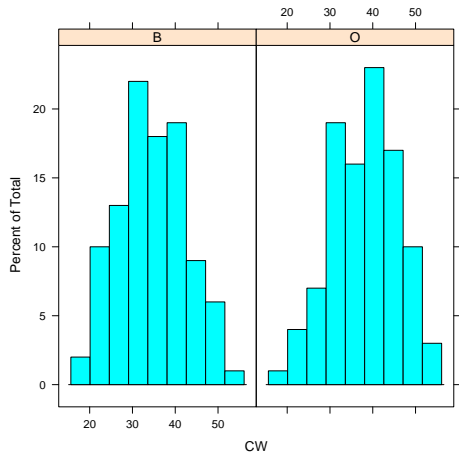
Graficos

# La librería lattice

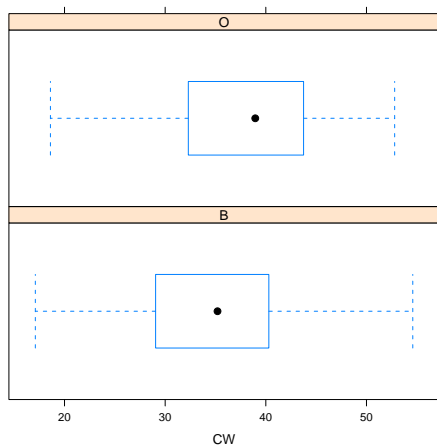
- ▶ El paquete lattice es muy útil para describir gráficamente datos multivariantes.
- ▶ La idea consiste en que el gráfico está formado por un cierto número de paneles.
- ▶ Normalmente cada uno de ellos corresponde a alguno de los valores de una variable que condiciona.
- ▶ Las funciones se escriben con la notación de la fórmula del modelo.
- ▶ En los gráficos univariantes como los histogramas, la variable respuesta, a la izquierda, se deja vacía.



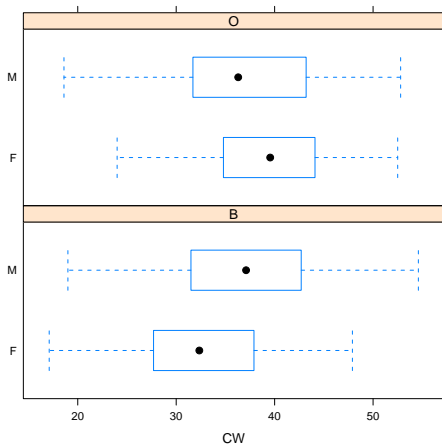
# Un grafico Trellis básico



## Representación de una variable condicionada por los valores de otra



# Notación de fórmula en gráficos Trellis



La última instrucción es muy interesante porque nos permite estudiar comparativamente el sexo separado por especies.