Data Science con R

Instituto de Estadística PUCV - Magister en Estadística

Dae-Jin Lee < dlee@bcamath.org >

Presentación "Data Visualization in Social Sciences"

Chambers et al. (1983)

"No existe una herramienta estadística tan poderosa como un gráfico bien escogido"

Visualización de datos

- Una de las principales razones por las que los analistas de datos recurren a `R` es por su gran capacidad gráfica.
- Esta sección proporciona una introducción completa sobre cómo representar datos mediante el sistema de gráficos por defecto de `R`.
- Las posibilidades gráficas de `R` son enormes (*casi infinitas*).
- Muchas librerías disponen de representaciones gráficas muy útiles para la representación de datos y de modelos.
- Veamos a continuación algunos ejemplos.

Objetivos de este tema

- Conocer las capacidades gráficas básicas de `R`
- Aprender a personalizar gráficos, y conocer los tipos de gráficos más complejos desde el punto de vista estadístico.
- Trabajar con datos reales y realizar análisis descriptivos y gráficos.
- Realizar gráficos con librerías como `ggplot2`
- Guardar gráficos en los diferentes formatos para utilizarlos posteriormente en presentaciones, informes etc ...

Preliminares

- Instalar las siguientes librerías de 'R

```
install.packages("DAAG")
install.packages("calibrate")
install.packages("corrplot")
install.packages("gplots")
install.packages("HSAUR2")
install.packages("sp")
install.packages("maps")
install.packages("maptools")
```

```
install.packages("RColorBrewer")
install.packages("RgoogleMaps")
```

Datos cuantitativos

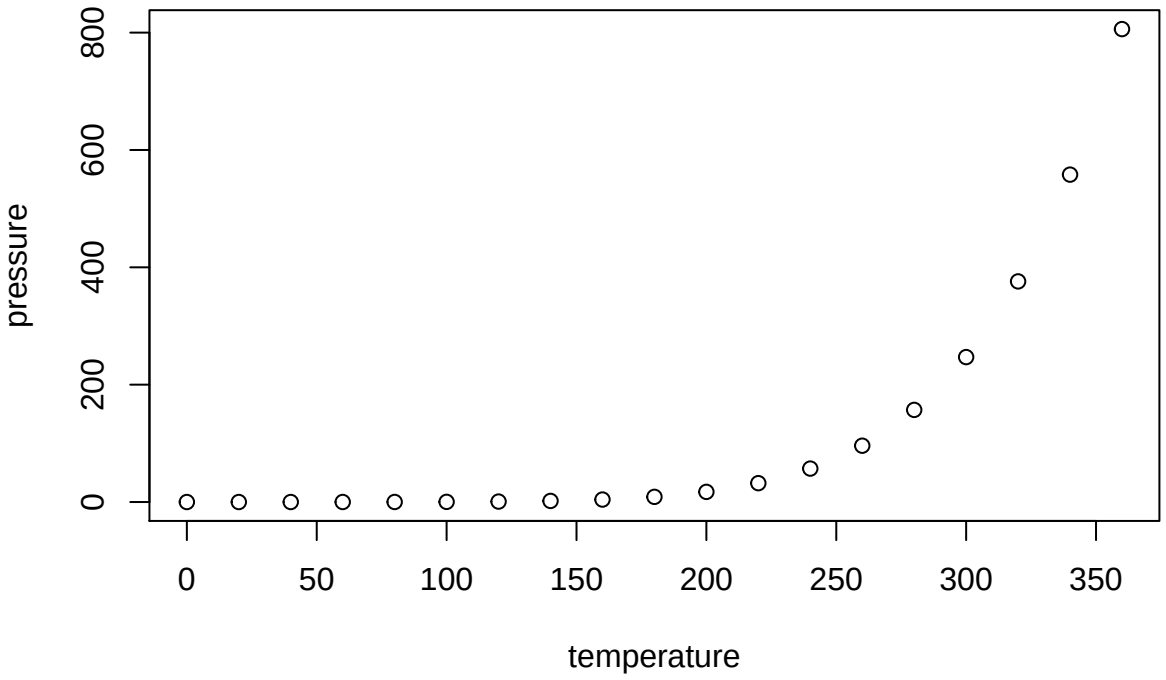
Vamos a comenzar con el conjunto de datos en el `data.frame`: `pressure` (ver `?pressure`)

```
?pressure
head(pressure)
```

```
##   temperature pressure
## 1           0   0.0002
## 2          20   0.0012
## 3          40   0.0060
## 4          60   0.0300
## 5          80   0.0900
## 6         100   0.2700
```

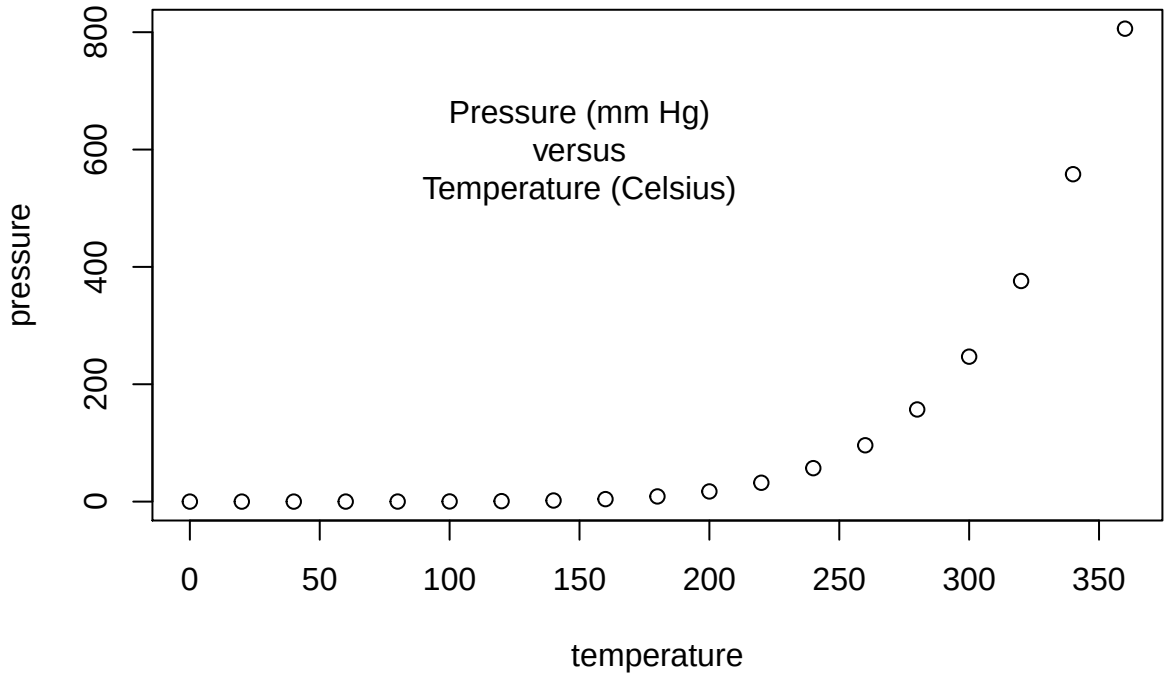
Función `plot`

```
plot(pressure)
```



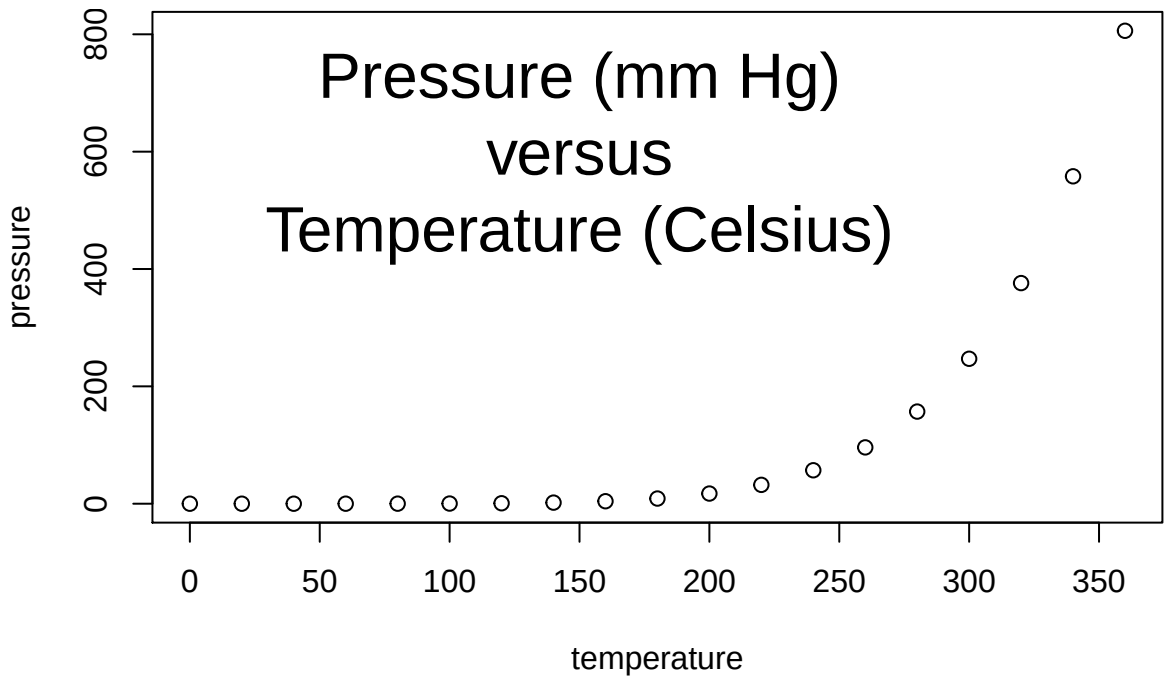
Función `text`

```
plot(pressure)
text(150, 600,
     "Pressure (mm Hg)\nversus\nTemperature (Celsius)")
```



La opción `cex` permite aumentar el tamaño de la fuente

```
plot(pressure)
text(150, 600, cex = 2,
     "Pressure (mm Hg)\nversus\nTemperature (Celsius)")
```

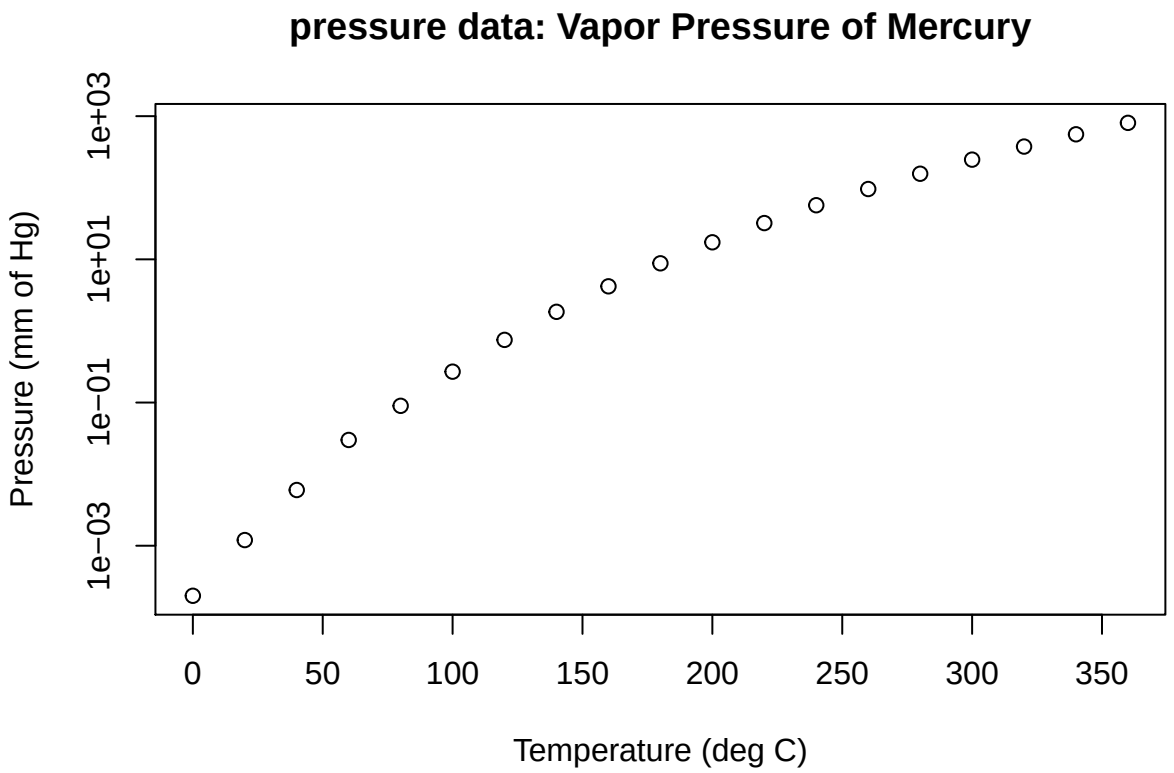


La opción `log = "y"`, representa la variable `y` en escala logarítmica.

`xlab` e `ylab` permite añadir texto a los ejes y `main` el título del gráfico.

```
plot(pressure, xlab = "Temperature (deg C)",  log = "y",
     ylab = "Pressure (mm of Hg)",
```

```
    main = "pressure data: Vapor Pressure of Mercury")
```



Datos `mtcars`

```
?mtcars
```

```
head(mtcars)
```

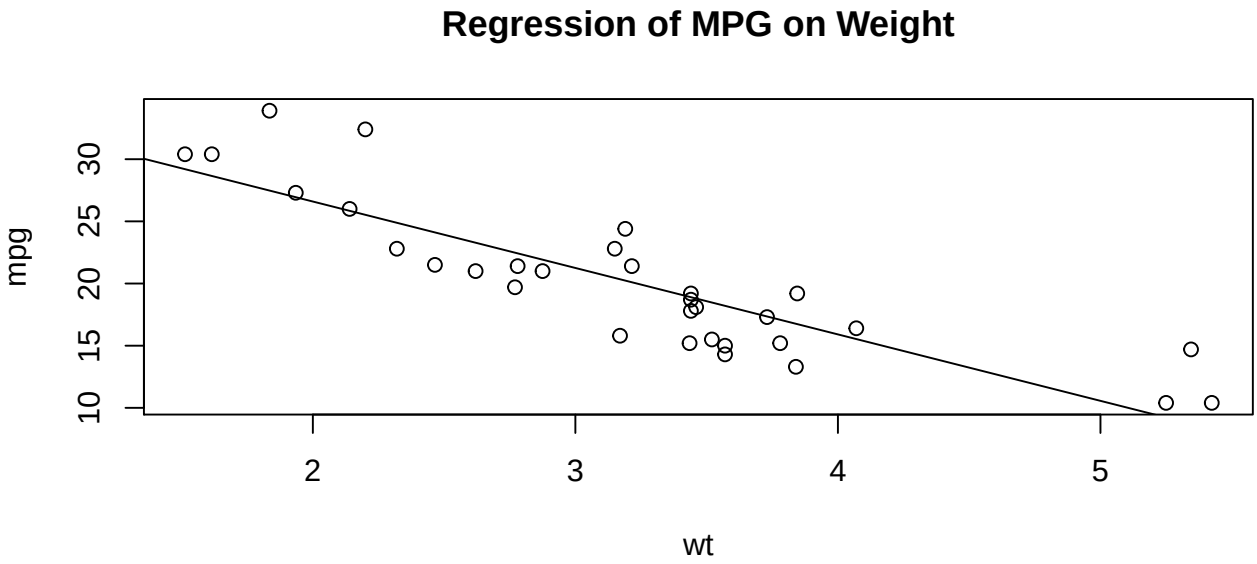
```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
names(mtcars)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

Scatterplot ó gráfico X-Y

```
attach(mtcars)
plot(wt, mpg);
abline(lm(mpg~wt)); title("Regression of MPG on Weight")
```



Scatterplot matrix (ver?pairs)

```
pairs(~mpg+disp+drat+wt,data=mtcars,
      main="Simple Scatterplot Matrix")
```

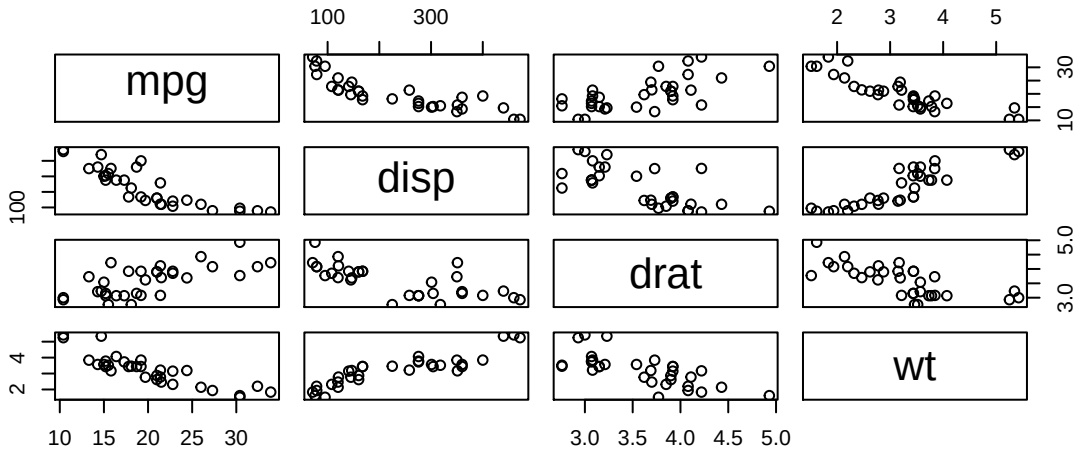
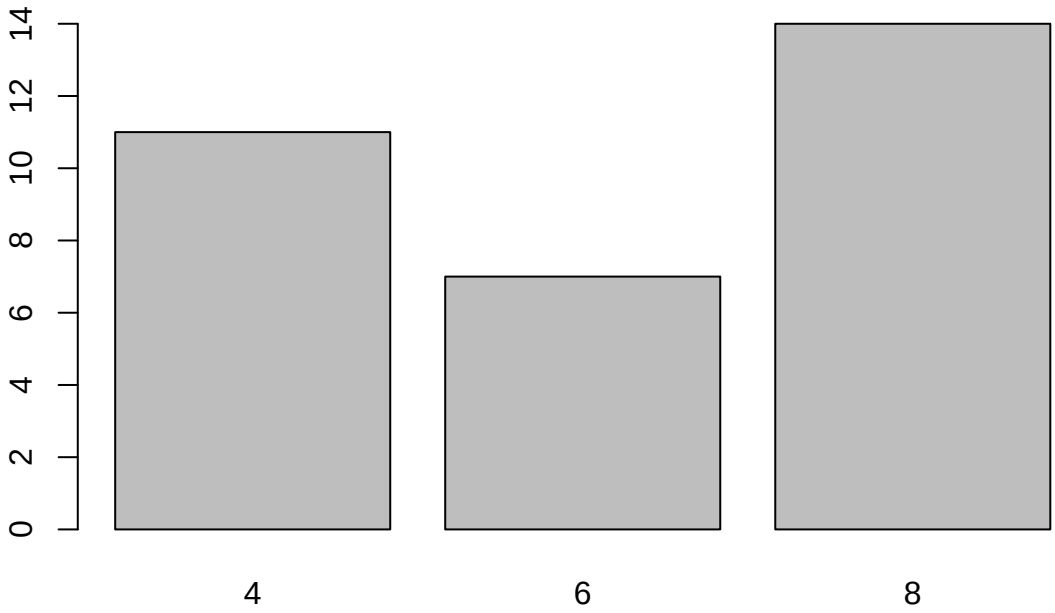


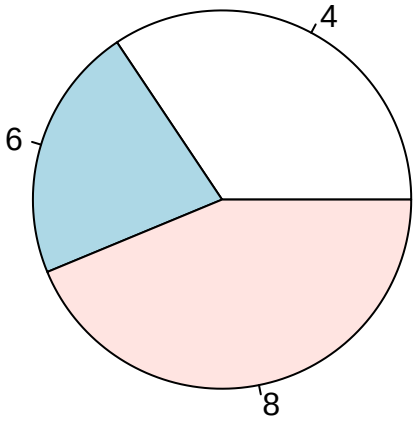
Diagrama de barras (ver ?barplot)

```
tab <- table(mtcars[,c("cyl")]) # convertir a tabla
barplot(tab)
```



Piechart o diagrama de tarta (ver `?pie`)

```
pie(tab)
```



Datos `VADeaths`

- El `data.frame` `VADeaths` contiene las tasas de mortalidad por cada 1000 habitantes en Virginia (EEUU) en 1940
- Las tasas de mortalidad se miden cada 1000 habitantes por año. Se encuentran clasificadas por grupo de edad (filas) y grupo de población (columnas).
- Los grupos de edad son: 50-54, 55-59, 60-64, 65-69, 70-74 y los grupos de población: `Rural/Male`, `Rural/Female`, `Urban/Male` and `Urban/Female`.

```
data(VADeaths)
VADeaths
```

```
##       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
```

- Calcula la media por grupo de edad y la media por grupo de población (**Pista:** puedes usar la función `apply`)

Función `apply`

- **Resultado**

```
apply(VADeaths,1,mean)
```

```
##  50-54  55-59  60-64  65-69  70-74
## 11.050 16.925 25.875 40.400 60.350
```

```
apply(VADeaths,2,mean)
```

```
##   Rural Male Rural Female   Urban Male Urban Female
##        32.74        25.18        40.48        25.28
```

Data `rainforest`

```
library(DAAG)
head(rainforest)
```

```
##    dbh wood bark root rootsk branch         species
## 27   6   NA   NA    6    0.3     NA Acacia mabellae
## 61  23  353   NA  135   13.0     35 Acacia mabellae
## 62  20  208   NA   NA     NA     41 Acacia mabellae
## 63  23  445   NA   NA     NA     50 Acacia mabellae
## 65  24  590   NA   NA     NA     NA Acacia mabellae
## 80   5   14   NA    2    2.4     NA Acacia mabellae
```

- Crear una tabla de conteos para cada `species` y realiza un gráfico descriptivo.
- **Resultado:**

```
table(rainforest$species)
```

```
##
## Acacia mabellae      C. fraseri  Acmena smithii   B. myrtifolia
##              16              12              26              11
```

Diagrama de barras

```
barplot(table(rainforest$species))
```


```
?subset
```

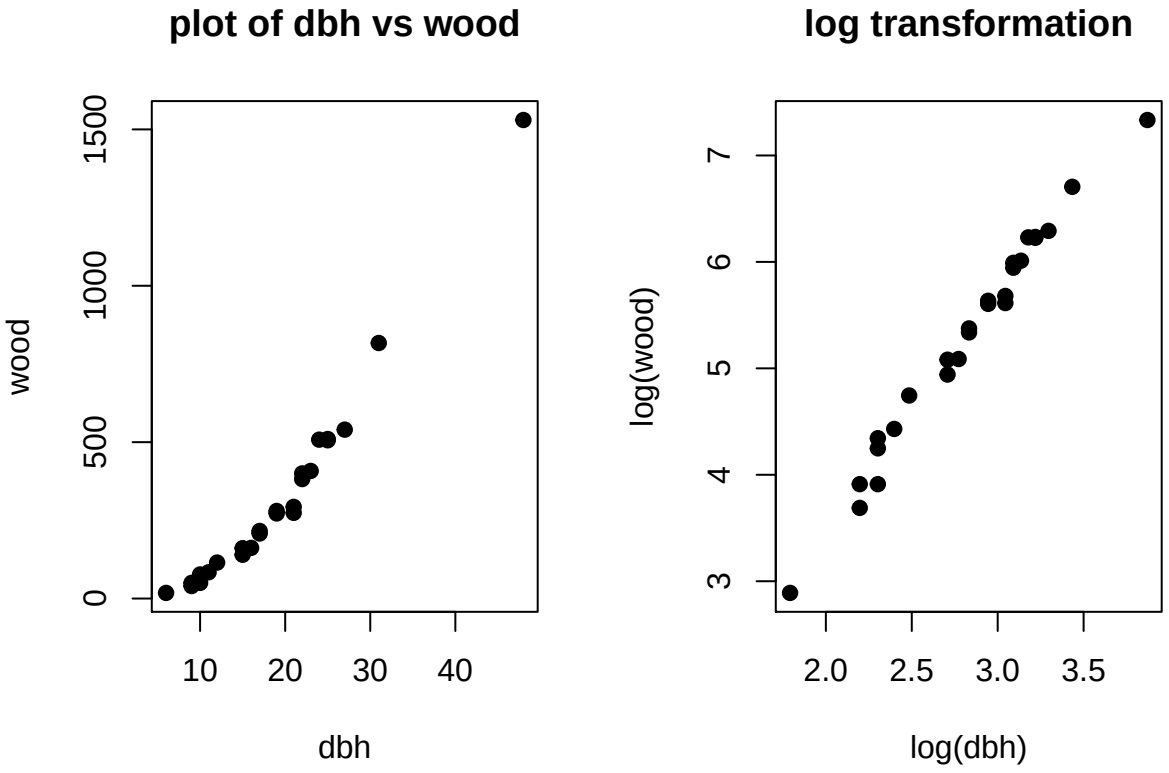
- El `data.frame` `Acmena` está creado a partir de `rainforest` mediante la función `subset`.

```
Acmena <- subset(rainforest, species == "Acmena smithii")
```

- Vamos a realizar un gráfico que relacione la biomasa de la madera (`wood`) y el diámetro a la altura del pecho (`dbh`).
- Utiliza también la escala logarítmica.

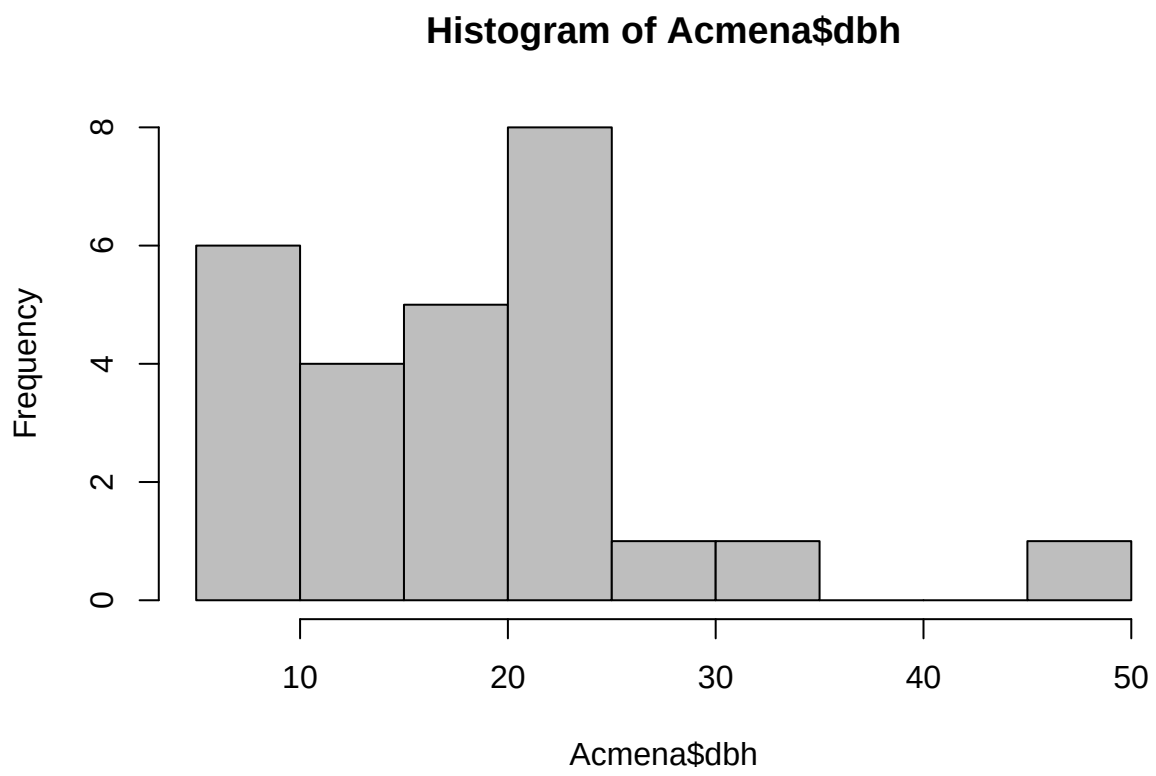
`par` y `mfrow`

```
par(mfrow=c(1,2))
plot(wood~dbh,data=Acmena,pch=19, main="plot of dbh vs wood")
plot(log(wood)~log(dbh),data=Acmena,pch=19,main="log transformation")
```



Histograma

```
hist(Acmena$dbh,col="grey")
```



Más sobre gráficos XY

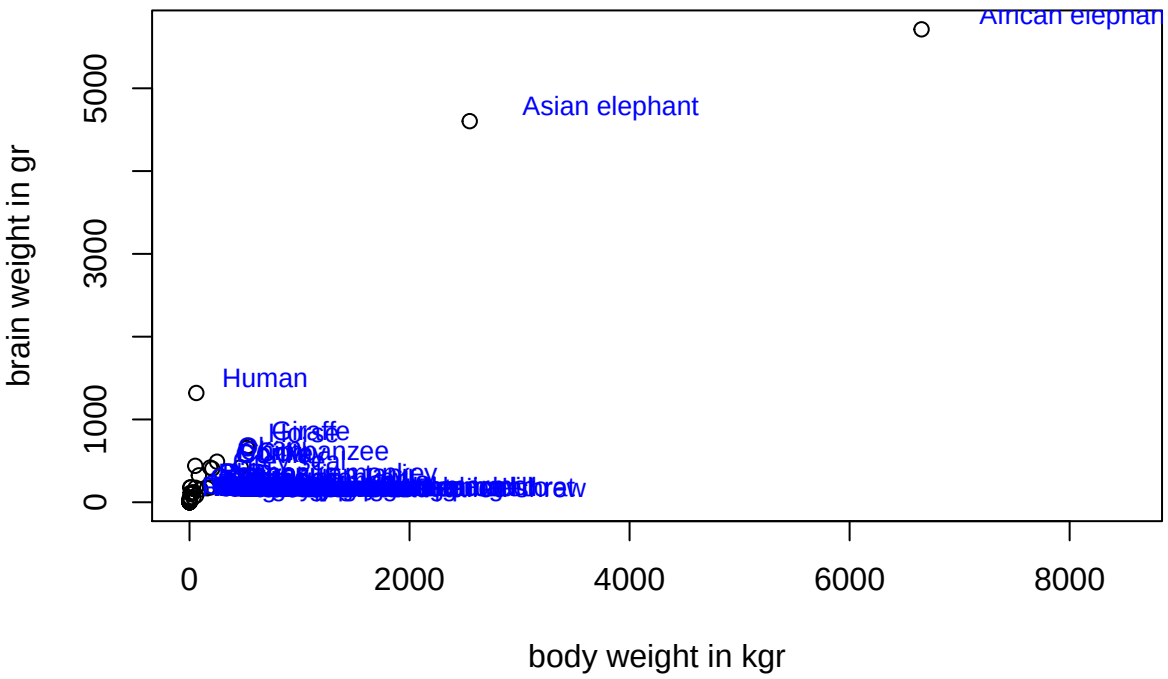
- Datos `mammals`

```
library(MASS)
data("mammals")
?mammals
head(mammals)
```

```
##                    body brain
## Arctic fox        3.385  44.5
## Owl monkey        0.480  15.5
## Mountain beaver   1.350   8.1
## Cow             465.000 423.0
## Grey wolf        36.330 119.5
## Goat             27.660 115.0
```

```
attach(mammals)
species <- row.names(mammals)
x <- body
y <- brain
```

```
library(calibrate)
# scatterplot
plot(x,y, xlab = "body weight in kgr", ylab = "brain weight in gr",
     main="Body vs Brain weight \n for 62 Species of Land Mammals",xlim=c(0,8500))
textxy(x,y,labs=species,col = "blue",cex=0.85)
```



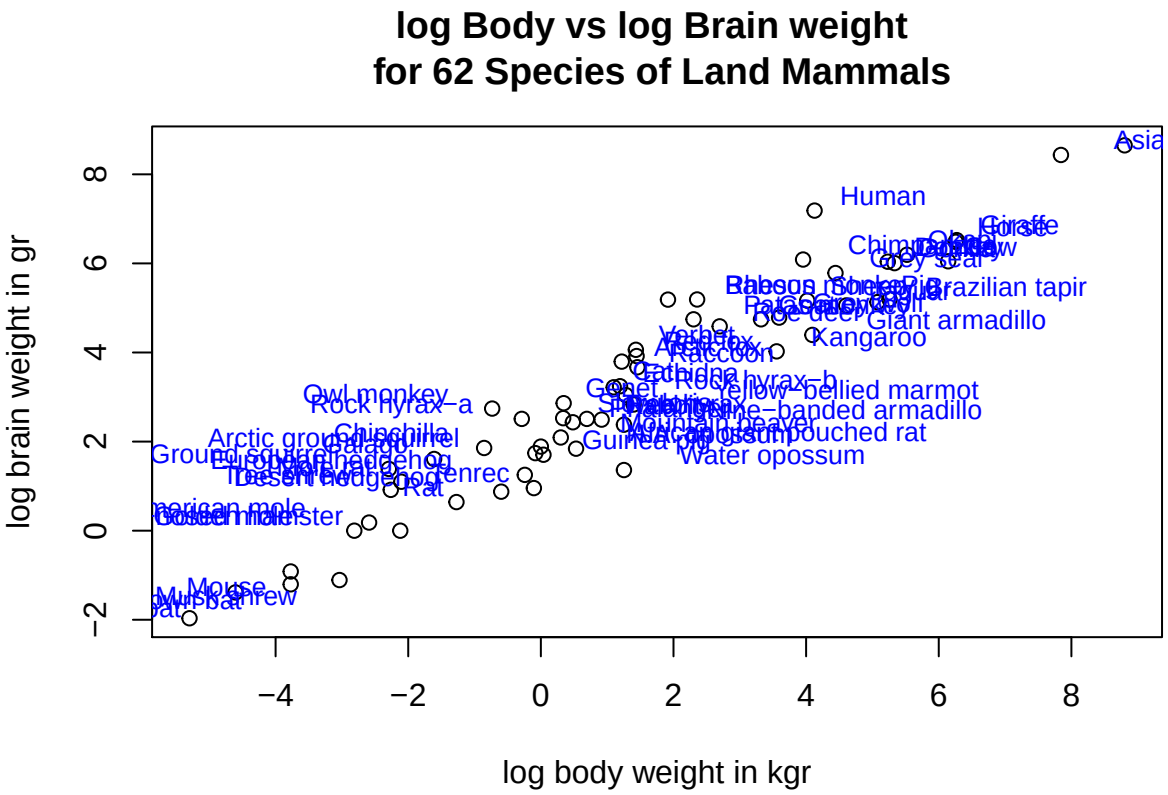
`identify`

Identificar un punto en el scatterplot

```
identify(x,y,species)
```

En escala logarítmica

```
plot(log(x),log(y), xlab = "log body weight in kgr", ylab = "log brain weight in gr",
     main="log Body vs log Brain weight \n for 62 Species of Land Mammals")
textxy(log(x),log(y),labs=species,col = "blue",cex=0.85)
```



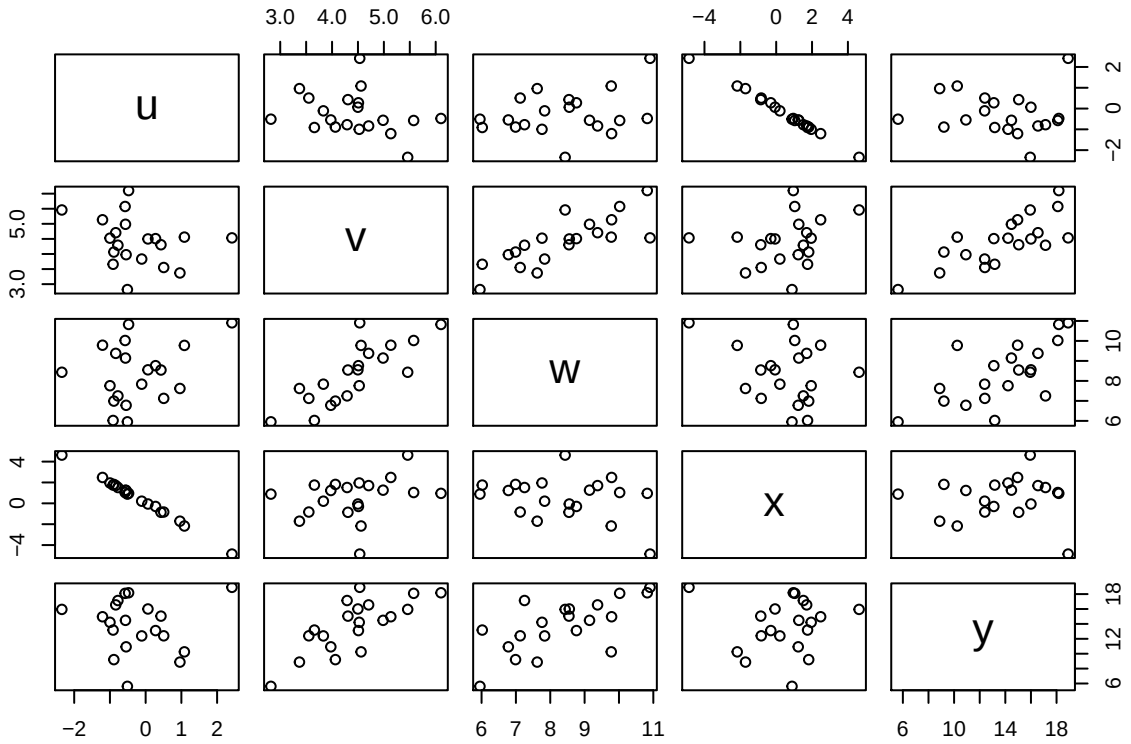
Matrices de correlación

- La función `corrplot` de la librería `corrplot` permite visualizar una matriz de correlaciones calculada mediante la función `cor`
- Vamos a generar unos datos de manera aleatoria.
- Mediante `set.seed(1234)` generaremos números aleatorios a partir de la misma semilla.

```
set.seed(1234)
uData <- rnorm(20)
vData <- rnorm(20,mean=5)
wData <- uData + 2*vData + rnorm(20,sd=0.5)
xData <- -2*uData+rnorm(20,sd=0.1)
yData <-  3*vData+rnorm(20,sd=2.5)
d <- data.frame(u=uData,v=vData,w=wData,x=xData,y=yData)
```

```
pairs
```

```
pairs(d)
```

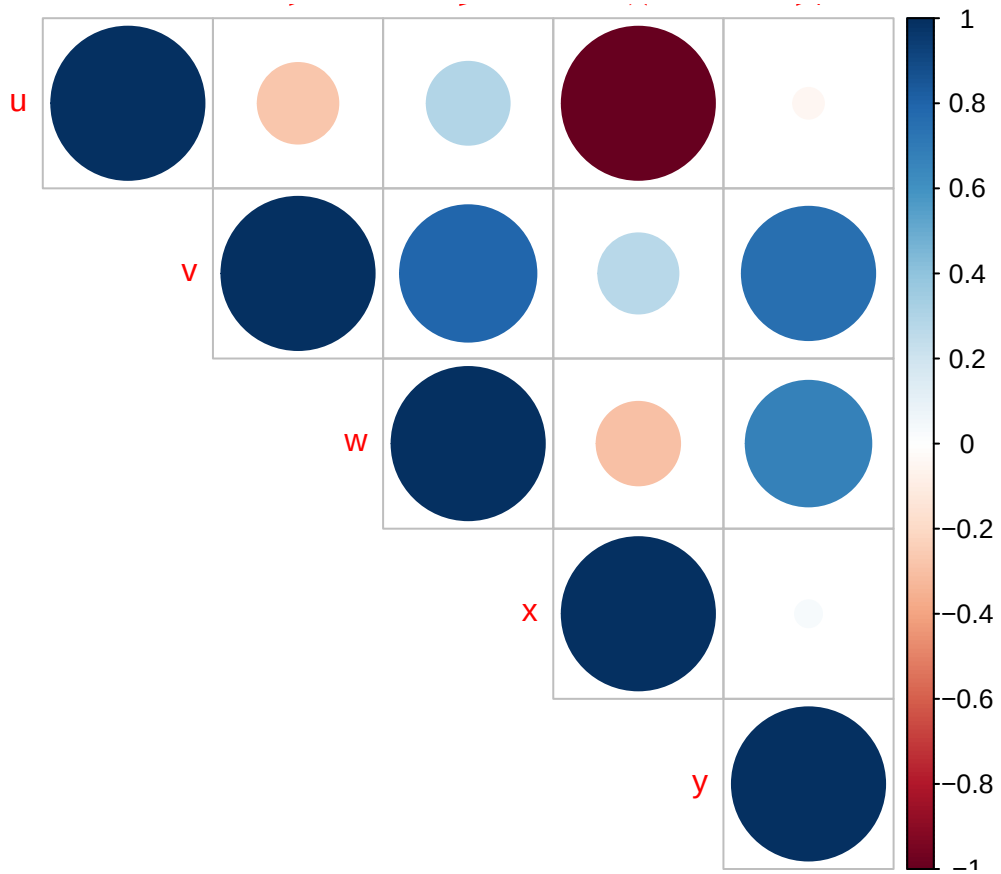


corrplot

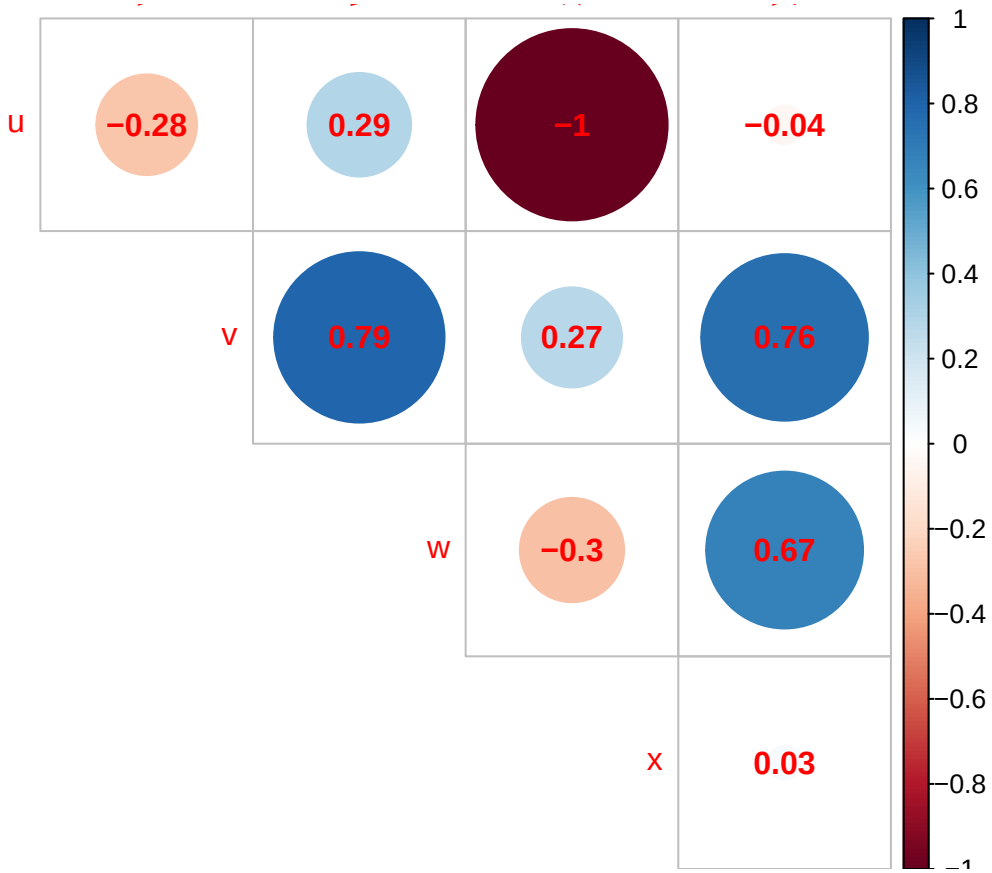
```
library(corrplot)
M <- cor(d)
M
```

```
##             u          v          w           x           y
## u  1.00000000 -0.2765719  0.2927511 -0.99868977 -0.04024035
## v -0.27657193  1.0000000  0.7924159  0.27353287  0.75719956
## w  0.29275113  0.7924159  1.0000000 -0.29680269  0.67192135
## x -0.99868977  0.2735329 -0.2968027  1.00000000  0.03090972
## y -0.04024035  0.7571996  0.6719213  0.03090972  1.00000000
```

```
corrplot(M, method="circle",type="upper")
```




```
corrplot(M, method="circle",type="upper", diag = FALSE,
         addCoef.col = "red")
```



UCBAdmissions

- El conjunto de datos de R, `UCBAdmissions`contiene los datos agregadps de los solicitantes a universidad de Berkeley a los seis departamentos más grandes en 1973 clasificados por sexo y admisión.

```
data("UCBAdmissions")
?UCBAdmissions
apply(UCBAdmissions, c(2,1), sum)
```

```
##         Admit
## Gender   Admitted Rejected
##   Male       1198     1493
##   Female      557     1278
```

```
prop.table(apply(UCBAdmissions, c(2,1), sum))
```

```
##         Admit
## Gender    Admitted  Rejected
##   Male   0.2646929 0.3298719
##   Female 0.1230667 0.2823685
```

```
ftable(UCBAdmissions)
```

```
##                 Dept   A   B   C   D   E   F
## Admit    Gender
## Admitted Male        512 353 120 138  53  22
##          Female       89  17 202 131  94  24
## Rejected Male        313 207 205 279 138 351
##          Female       19   8 391 244 299 317
```

Con `ftable` podemos presentar la información con mayor claridad

```
ftable(round(prop.table(UCBAdmissions), 3),
       row.vars="Dept", col.vars = c("Gender", "Admit"))
```

```
##      Gender     Male            Female
##      Admit  Admitted Rejected Admitted Rejected
## Dept
## A              0.113    0.069    0.020    0.004
## B              0.078    0.046    0.004    0.002
## C              0.027    0.045    0.045    0.086
## D              0.030    0.062    0.029    0.054
## E              0.012    0.030    0.021    0.066
## F              0.005    0.078    0.005    0.070
```

Resulta más intereseante mostrar la información por género `Gender` y `Dept` combinados (dimensiones 2 y 3 del array). Nótese que las tasas de admisión por `male` y `female` son más o menos similares en todos los departamentos, excepto en "A", donde las tasas de las mujeres es mayor.

```
ftable(round(prop.table(UCBAdmissions, c(2,3)), 2),
       row.vars="Dept", col.vars = c("Gender", "Admit"))
```

```
##      Gender     Male            Female
##      Admit  Admitted Rejected Admitted Rejected
## Dept
## A              0.62     0.38     0.82     0.18
## B              0.63     0.37     0.68     0.32
## C              0.37     0.63     0.34     0.66
## D              0.33     0.67     0.35     0.65
## E              0.28     0.72     0.24     0.76
## F              0.06     0.94     0.07     0.93
```

Datos de admisiones agregados por Sexo/Departamento

```
apply(UCBAdmissions, c(1, 2), sum)
```

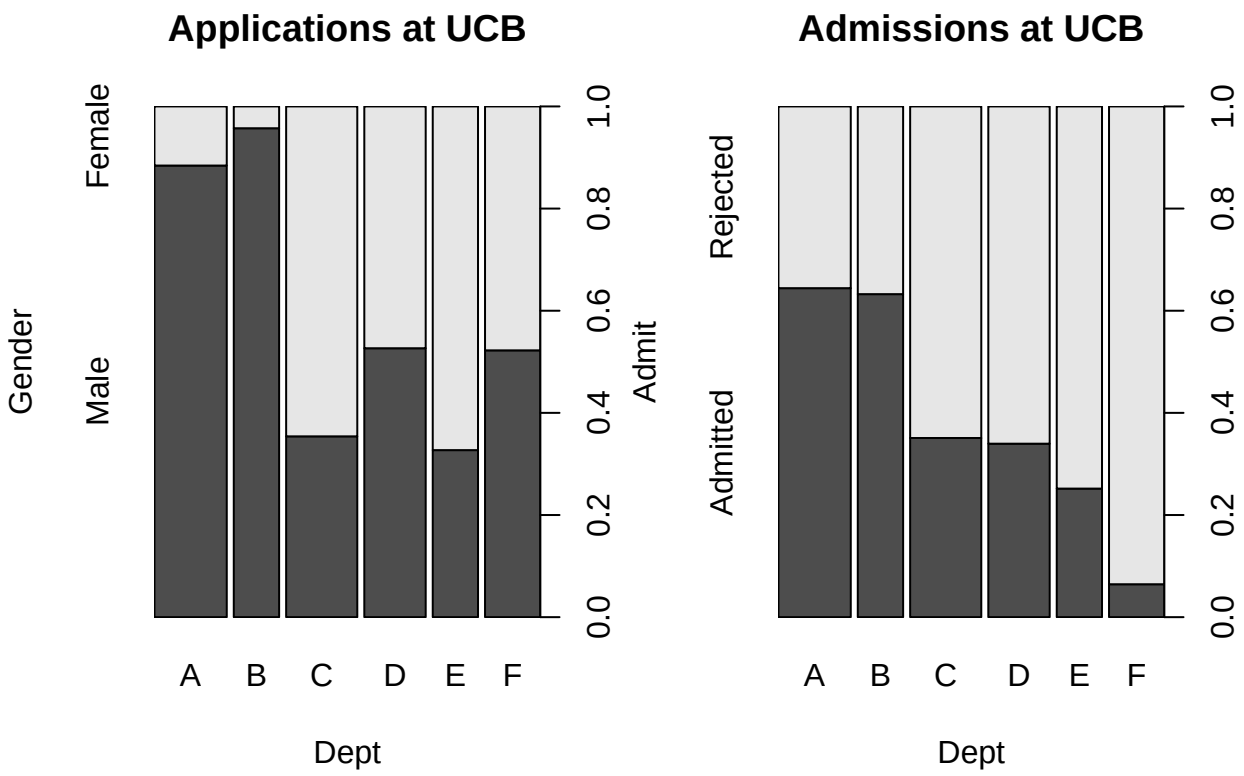
```
##           Gender
## Admit      Male Female
##   Admitted 1198    557
##   Rejected 1493   1278
```

```
apply(UCBAdmissions, c(1, 2), sum)
```

```
##           Gender
## Admit      Male Female
##   Admitted 1198    557
##   Rejected 1493   1278
```

Representación gráfica datos categóricos (`spineplot`)

```
par(mfrow=c(1,2))
spineplot(margin.table(UCBAdmissions, c(3, 2)),
          main = "Applications at UCB")
spineplot(margin.table(UCBAdmissions, c(3, 1)),
          main = "Admissions at UCB")
```



Paradoja de Simpson

- Estos datos ilustran la denominada *paradoja de Simpson*.
- Este hecho ha sido analizado como un posible caso de discriminación por sexo en las tasas de admisión en Berkeley.
- De los 2691 hombres que solicitaron se admitidos, 1198 (44.5%) fueron admitidos, comparado con las 1835 mujeres de las cuales tan sólo 557 (30.4%) fueron admitidas.
- Se podría por tanto concluir que los hombres tienes tasas de admisi?n mayores que las mujeres.
- **Wikipedia:** *Gender Bias UC Berkeley*.
- Ver animación en *link*

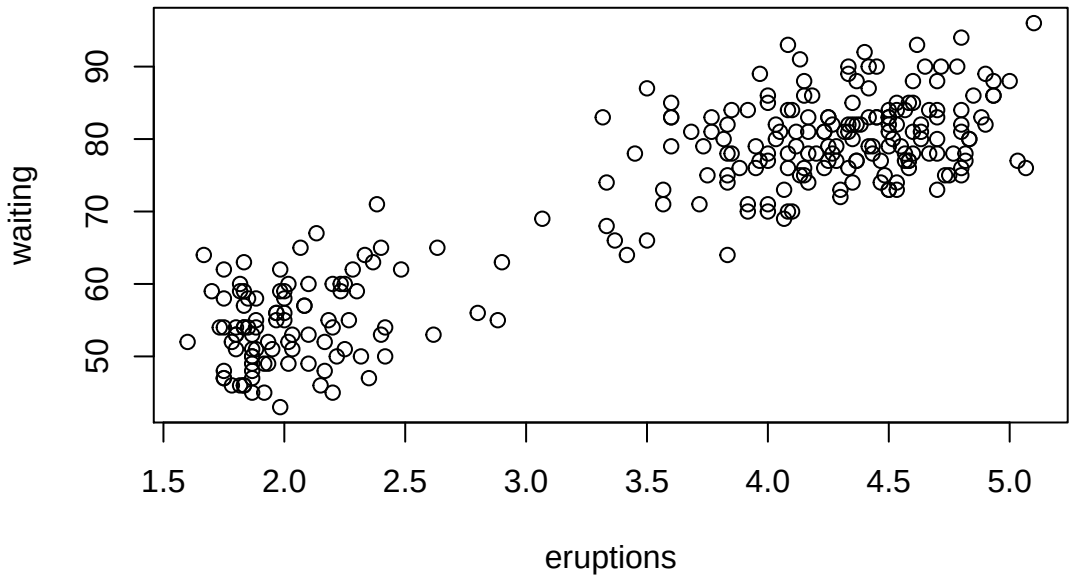
Datos `faithful`

- Consideremos los datos del geyse Old Faithful en el parque nacional de Yellowstone, EEUU.

```
head(faithful)
```

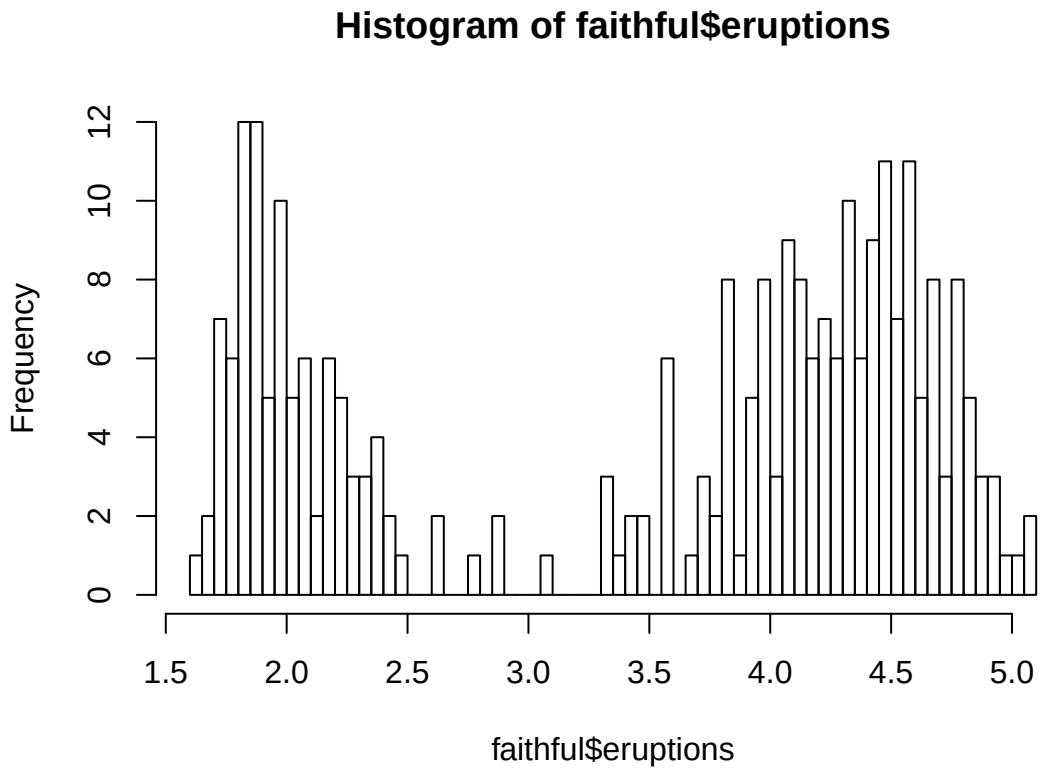
```
##   eruptions waiting
## 1     3.600      79
## 2     1.800      54
## 3     3.333      74
## 4     2.283      62
## 5     4.533      85
## 6     2.883      55
```

```
plot(faithful)
```



Histograma (`hist`)

```
hist(faithful$eruptions,50)
```



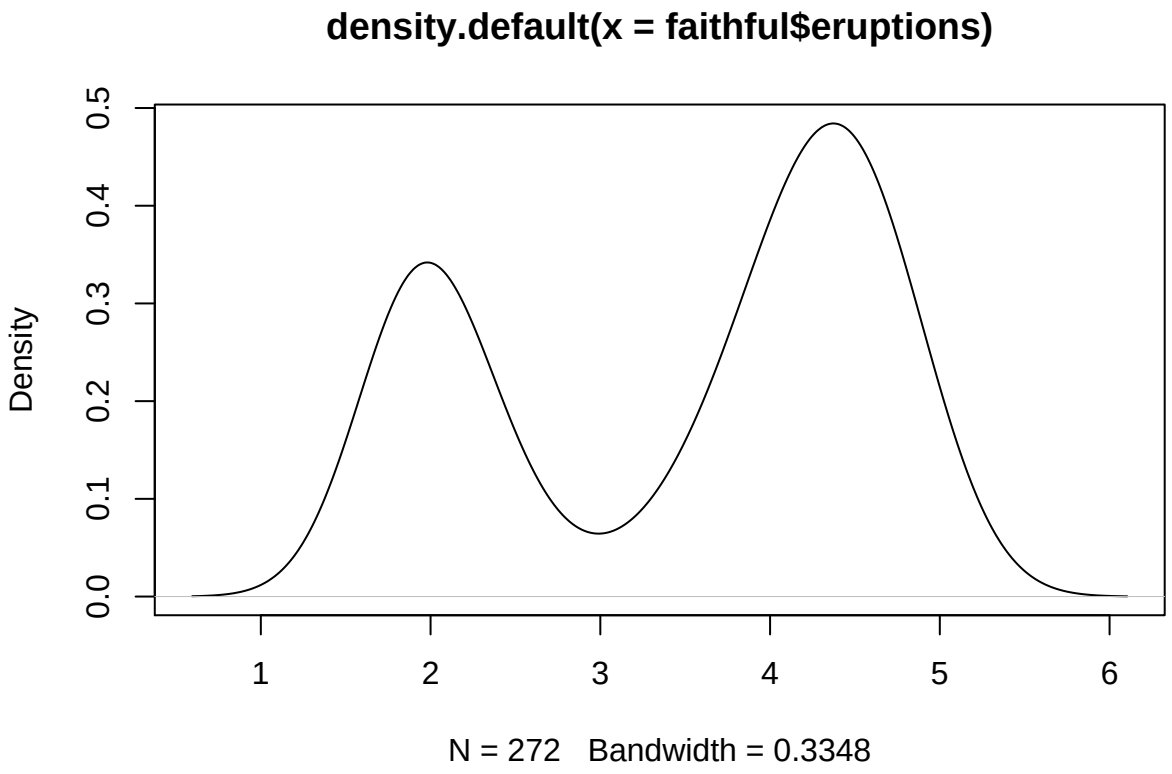
Estimación de densidades

- **Estimación de densidad** construye una estimación dada una distribucion de probabilidad para una muestra dada.

```
library(graphics)
d <- density(faithful$eruptions)
d
```

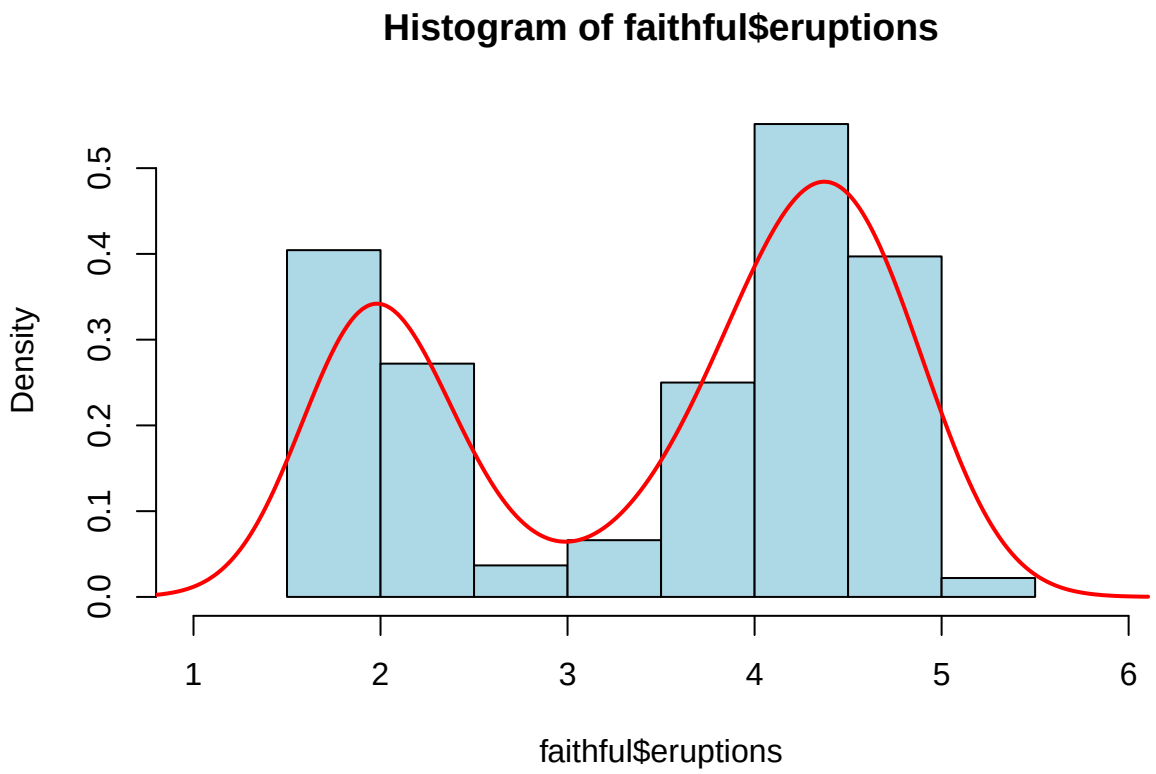
```
##
## Call:
##  density.default(x = faithful$eruptions)
##
## Data: faithful$eruptions (272 obs.); Bandwidth 'bw' = 0.3348
##
##        x                y
##  Min.   :0.5957   Min.   :0.0002262
##  1st Qu.:1.9728   1st Qu.:0.0514171
##  Median :3.3500   Median :0.1447010
##  Mean   :3.3500   Mean   :0.1813462
##  3rd Qu.:4.7272   3rd Qu.:0.3086071
##  Max.   :6.1043   Max.   :0.4842095
```

```
plot(d)
```



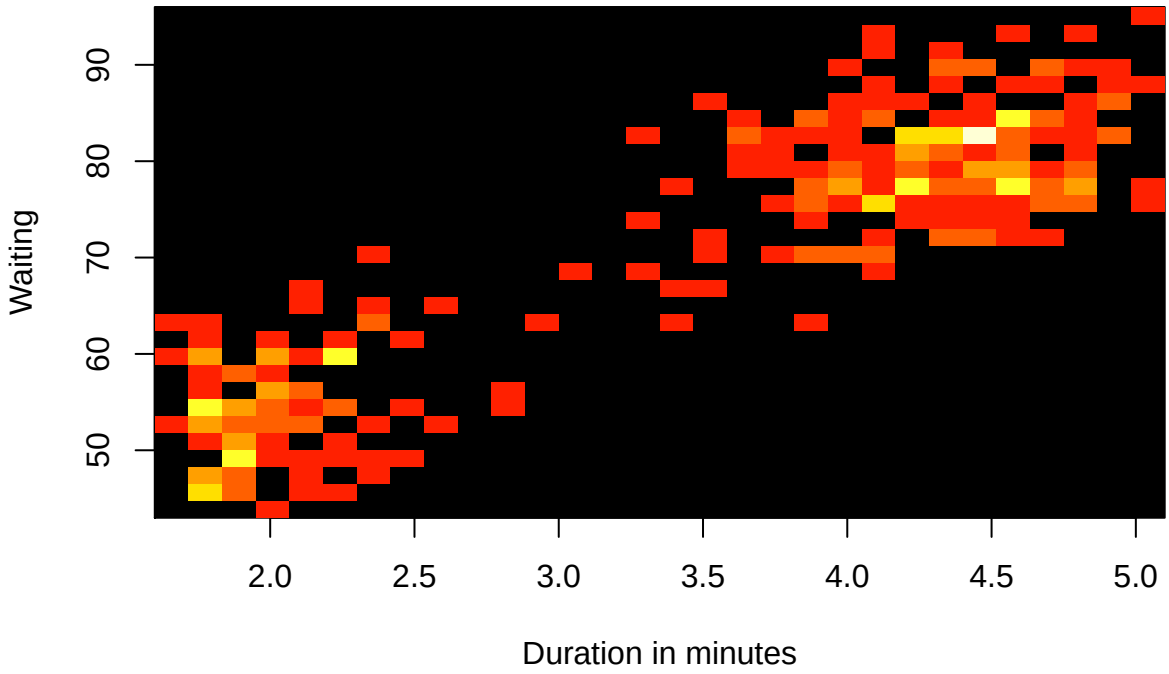
Histograma y Densidad

```
hist(faithful$eruptions,freq=FALSE, col = "lightblue", xlim = c(1,6))
lines(d, col = "red", lwd = 2)
```



Histograma bivariante

```
library(gplots)
h2 <- hist2d(faithful, nbins=30,xlab="Duration in minutes",ylab="Waiting")
```




```
h2
```

```
##
## ----------------------------
```

```
## 2-D Histogram Object
## ----------------------------
##
## Call: hist2d(x = faithful, nbins = 30, xlab = "Duration in minutes",
##     ylab = "Waiting")
##
## Number of data points:  272
## Number of grid bins:  30 x 30
## X range: ( 1.6 , 5.1 )
## Y range: ( 43 , 96 )
```

```
class(h2)
```

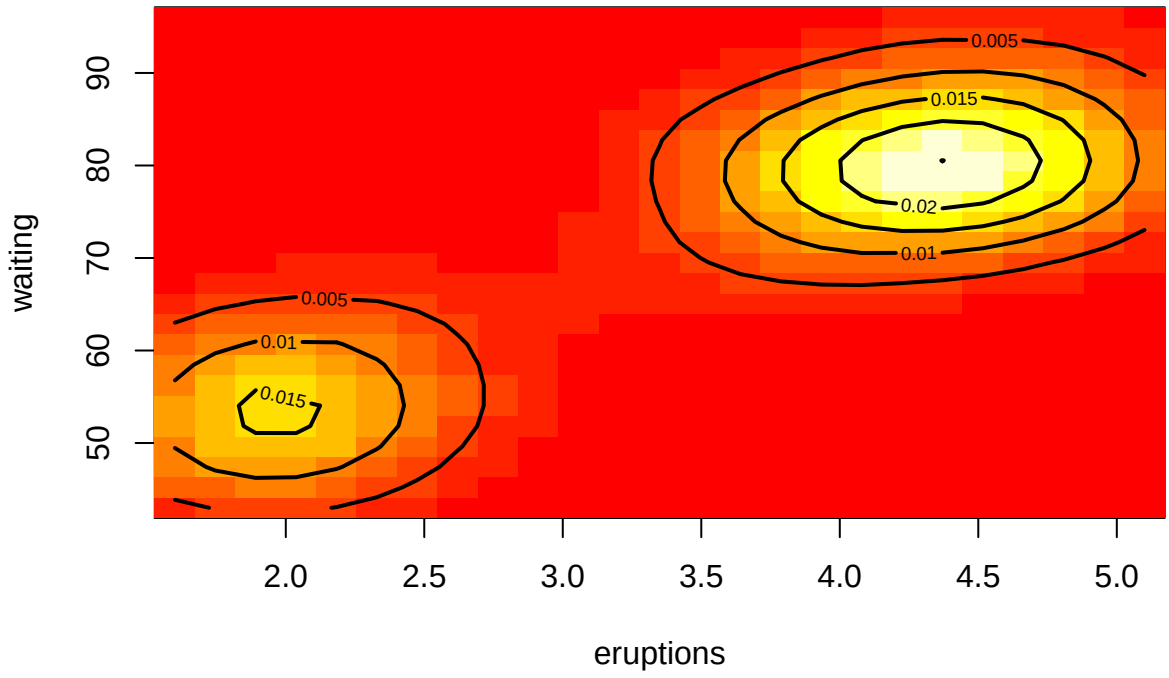
```
## [1] "hist2d"
```

```
names(h2)
```

```
## [1] "counts"   "x.breaks" "y.breaks" "x"        "y"        "nobs"
## [7] "call"
```

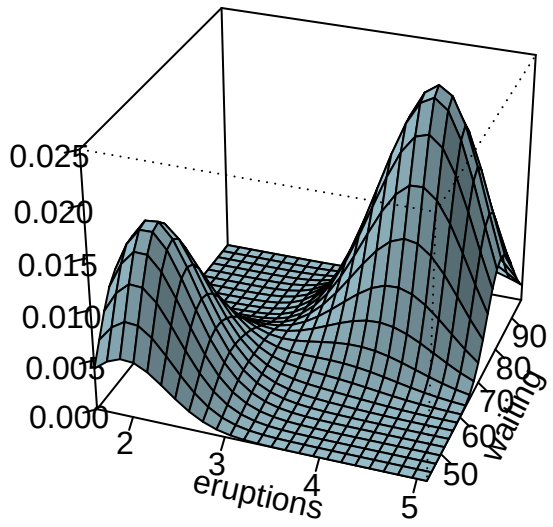
Estimación de densidades bivariantes (kde2d)

```
Dens2d<-kde2d(faithful$eruptions,faithful$waiting)
image(Dens2d,xlab="eruptions",ylab="waiting")
contour(Dens2d,add=TRUE,col="black",lwd=2,nlevels=5)
```



persp

```
persp(Dens2d,phi=30,theta=20,d=5,xlab="eruptions",ylab="waiting",zlab="",shade=.2,col="lightblue",expand=.85,ticktype = "detaile
```



Ejemplo: Forbes 2000 (ranking de las empresas líderes en 2004)

- La lista Forbes 2000 para el año 2004 recogida por la revista Forbes. Esta lista está disponible originalmente en `www.forbes.com`

```
library("HSAUR2")
data("Forbes2000")
dim(Forbes2000)
```

```
## [1] 2000    8
```

```
names(Forbes2000)
```

```
## [1] "rank"        "name"        "country"     "category"    "sales"
## [6] "profits"     "assets"      "marketvalue"
```

```
library(knitr)
kable(head(Forbes2000))
```

rank	name	country	category	sales	profits	assets	marketvalue
1	Citigroup	United States	Banking	94.71	17.85	1264.03	255.30
2	General Electric	United States	Conglomerates	134.19	15.59	626.93	328.54
3	American Intl Group	United States	Insurance	76.66	6.46	647.66	194.87
4	ExxonMobil	United States	Oil & gas operations	222.88	20.96	166.99	277.02
5	BP	United Kingdom	Oil & gas operations	232.57	10.27	177.57	173.54
6	Bank of America	United States	Banking	49.01	10.81	736.45	117.55

Los datos consisten en 2000 observaciones sobre las 8 variables siguientes.

- `rank`: el ranking de la empresa.
- `name`: el nombre de la empresa.
- `country`: un factor que determina el país en el que está situada la empresa.
- `category`: un factor que describe los productos que produce la empresa.
- `sales`: el importe de las ventas de la empresa en miles de millones de dólares.
- `profits`: los beneficios de la empresa en miles de millones de dólares.
- `assets`: los activos de la empresa en miles de millones de dólares.
- `marketvalue`: el valor de mercado de la empresa en miles de millones de dólares.

```
str(Forbes2000)
```

```
## 'data.frame':    2000 obs. of  8 variables:
##  $ rank       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ name       : chr  "Citigroup" "General Electric" "American Intl Group" "ExxonMobil" ...
##  $ country    : Factor w/ 61 levels "Africa","Australia",..: 60 60 60 60 56 60 56 28 60 60 ...
```

```
##  $ category   : Factor w/ 27 levels "Aerospace & defense",..: 2 6 16 19 19 2 2 8 9 20 ...
##  $ sales      : num  94.7 134.2 76.7 222.9 232.6 ...
##  $ profits    : num  17.85 15.59 6.46 20.96 10.27 ...
##  $ assets     : num  1264 627 648 167 178 ...
##  $ marketvalue: num  255 329 195 277 174 ...
```

- ¿Cuántos países diferentes están en el ranking del año 2000?

```
nlevels(Forbes2000[,"country"])
```

```
## [1] 61
```

- Cuáles son éstos países?

```
levels(Forbes2000[,"country"])
```

```
##  [1] "Africa"                       "Australia"
##  [3] "Australia/ United Kingdom"    "Austria"
##  [5] "Bahamas"                      "Belgium"
##  [7] "Bermuda"                      "Brazil"
##  [9] "Canada"                       "Cayman Islands"
## [11] "Chile"                        "China"
## [13] "Czech Republic"               "Denmark"
## [15] "Finland"                      "France"
## [17] "France/ United Kingdom"       "Germany"
## [19] "Greece"                       "Hong Kong/China"
## [21] "Hungary"                      "India"
## [23] "Indonesia"                    "Ireland"
## [25] "Islands"                      "Israel"
## [27] "Italy"                        "Japan"
## [29] "Jordan"                       "Kong/China"
## [31] "Korea"                        "Liberia"
## [33] "Luxembourg"                   "Malaysia"
## [35] "Mexico"                       "Netherlands"
## [37] "Netherlands/ United Kingdom"  "New Zealand"
## [39] "Norway"                       "Pakistan"
## [41] "Panama/ United Kingdom"       "Peru"
## [43] "Philippines"                  "Poland"
## [45] "Portugal"                     "Russia"
## [47] "Singapore"                    "South Africa"
## [49] "South Korea"                  "Spain"
## [51] "Sweden"                       "Switzerland"
## [53] "Taiwan"                       "Thailand"
## [55] "Turkey"                       "United Kingdom"
## [57] "United Kingdom/ Australia"    "United Kingdom/ Netherlands"
## [59] "United Kingdom/ South Africa" "United States"
## [61] "Venezuela"
```

- Cuáles en el top 20?

```
top20 <- droplevels(subset(Forbes2000,rank<=20))
levels(top20[,"country"])
```

```
## [1] "France"                      "Japan"
## [3] "Netherlands"                 "Netherlands/ United Kingdom"
## [5] "Switzerland"                 "United Kingdom"
## [7] "United States"
```

- As a simple summary statistic, the frequencies of the levels of such a factor variable can be found from

```
table(top20[,"country"])
```

```
##
##                      France                       Japan
##                           2                           1
##                 Netherlands Netherlands/ United Kingdom
##                           1                           1
##                 Switzerland              United Kingdom
##                           1                           3
##               United States
```

```
##                          11
```

- Which type of companies?

```
levels(Forbes2000[,"category"])
```

```
##  [1] "Aerospace & defense"              "Banking"
##  [3] "Business services & supplies"     "Capital goods"
##  [5] "Chemicals"                        "Conglomerates"
##  [7] "Construction"                     "Consumer durables"
##  [9] "Diversified financials"           "Drugs & biotechnology"
## [11] "Food drink & tobacco"             "Food markets"
## [13] "Health care equipment & services" "Hotels restaurants & leisure"
## [15] "Household & personal products"    "Insurance"
## [17] "Materials"                        "Media"
## [19] "Oil & gas operations"             "Retailing"
## [21] "Semiconductors"                   "Software & services"
## [23] "Technology hardware & equipment"  "Telecommunications services"
## [25] "Trading companies"                "Transportation"
## [27] "Utilities"
```

- How many of each category?

```
table(Forbes2000[,"category"])
```

```
##
##              Aerospace & defense                          Banking
##                               19                              313
##     Business services & supplies                  Capital goods
##                               70                               53
##                         Chemicals                  Conglomerates
##                               50                               31
##                     Construction                Consumer durables
##                               79                               74
##           Diversified financials            Drugs & biotechnology
##                              158                               45
##             Food drink & tobacco                     Food markets
##                               83                               33
## Health care equipment & services     Hotels restaurants & leisure
##                               65                               37
##    Household & personal products                        Insurance
##                               44                              112
##                         Materials                           Media
##                               97                               61
##             Oil & gas operations                        Retailing
##                               90                               88
##                   Semiconductors              Software & services
##                               26                               31
##  Technology hardware & equipment      Telecommunications services
##                               59                               67
##                Trading companies                   Transportation
##                               25                               80
##                        Utilities
##                              110
```

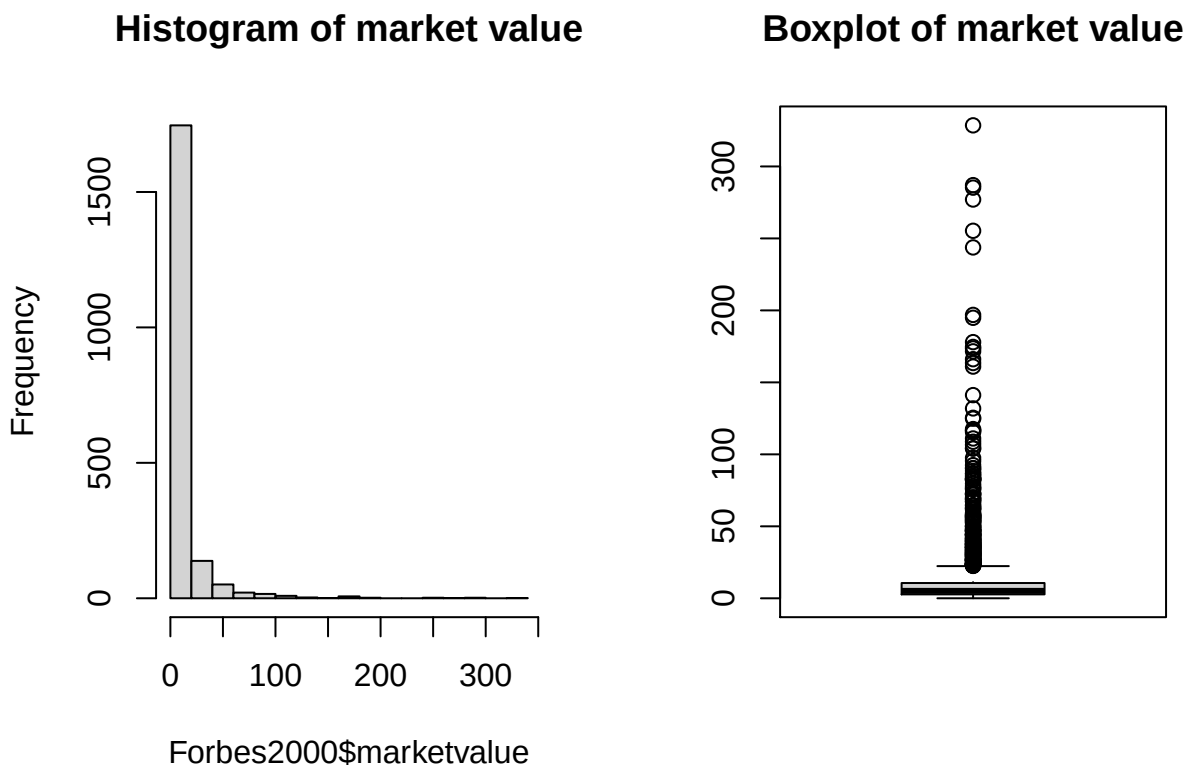
- A simple summary statistics such as the mean, median, quantiles and range can be found from continuous variables such as `sales`

```
summary(Forbes2000[,"sales"])
```

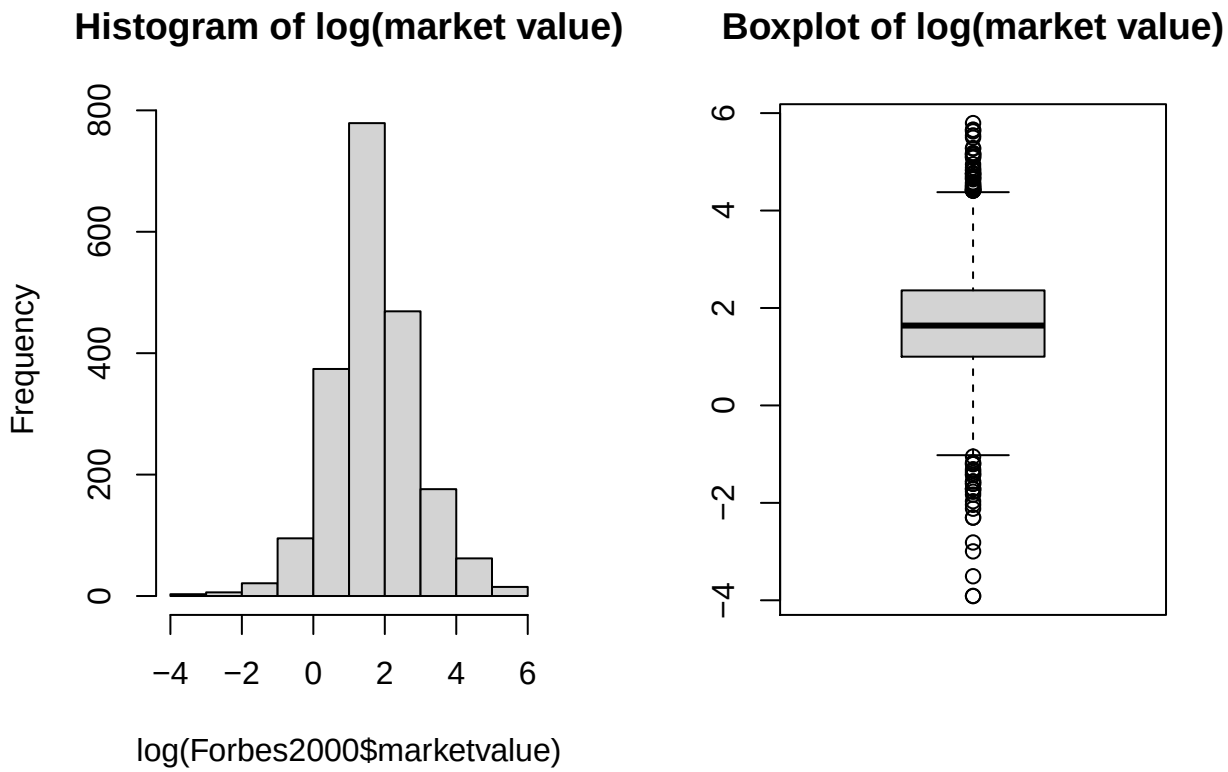
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.010   2.018   4.365   9.697   9.548 256.300
```

- Histogramas y boxplots

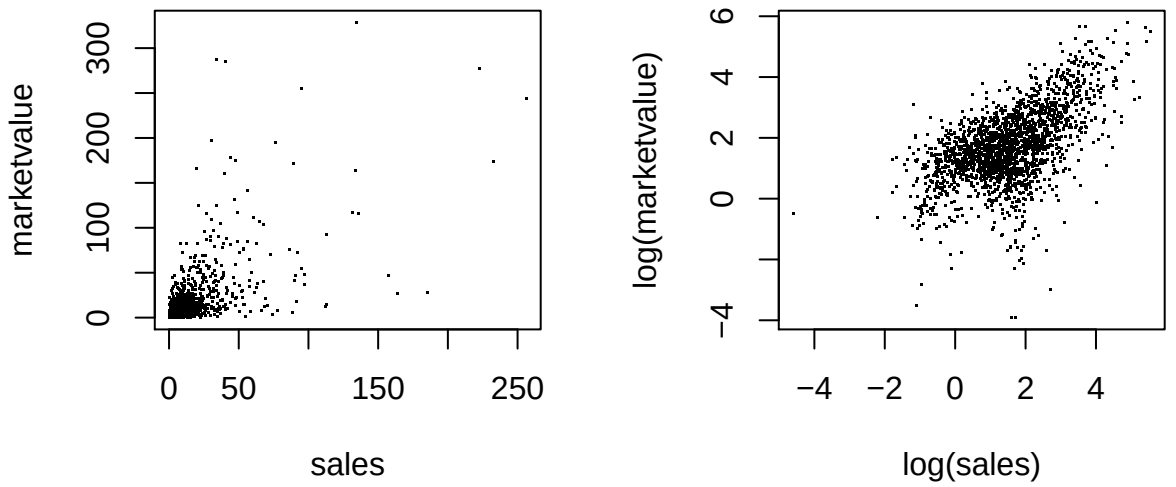
```
par(mfrow=c(1,2))
hist(Forbes2000$marketvalue, col="lightgrey",main="Histogram of market value")
boxplot(Forbes2000$marketvalue, col="lightgrey",main="Boxplot of market value")
```



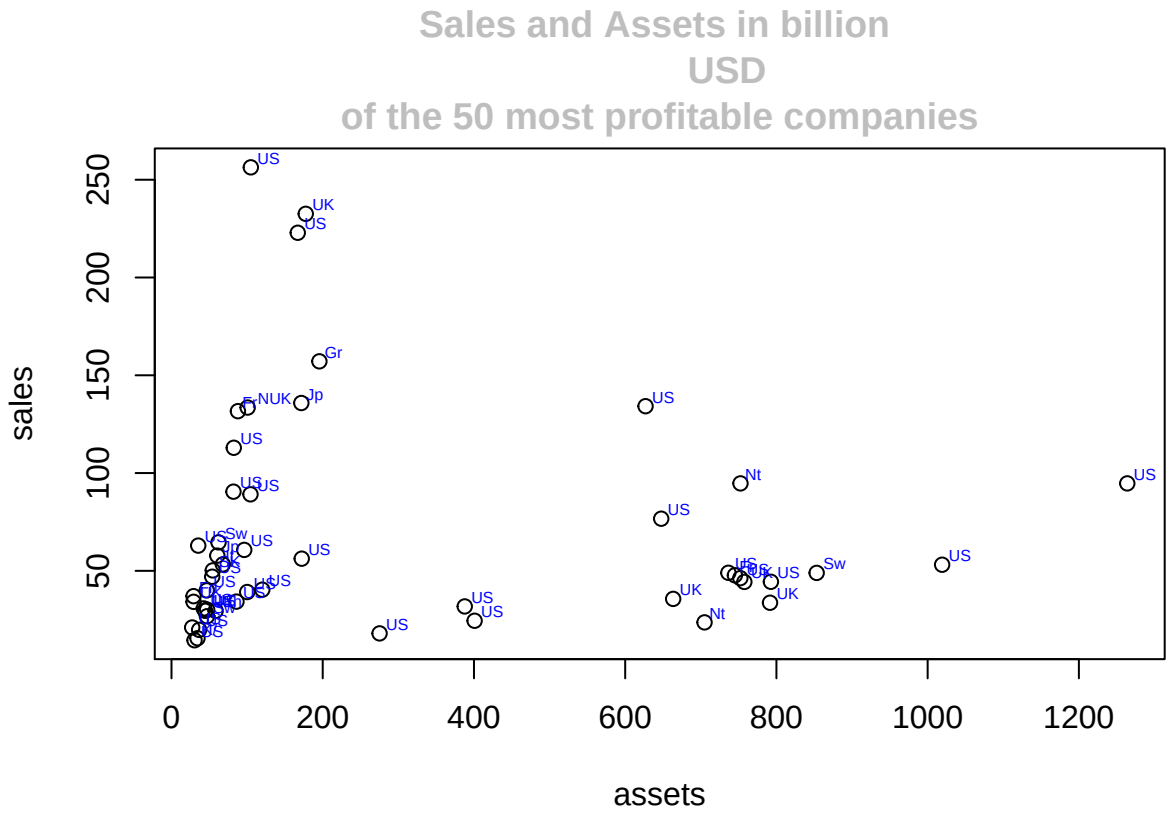

```
par(mfrow=c(1,2))
hist(log(Forbes2000$marketvalue),col="lightgrey",
     main="Histogram of log(market value)")
boxplot(log(Forbes2000$marketvalue),col="lightgrey",
        main="Boxplot of log(market value)")
```



```r
par(mfrow=c(1,2))
plot(marketvalue ~ sales, data = Forbes2000, pch = ".")
plot(log(marketvalue) ~ log(sales), data = Forbes2000, pch = ".")
```



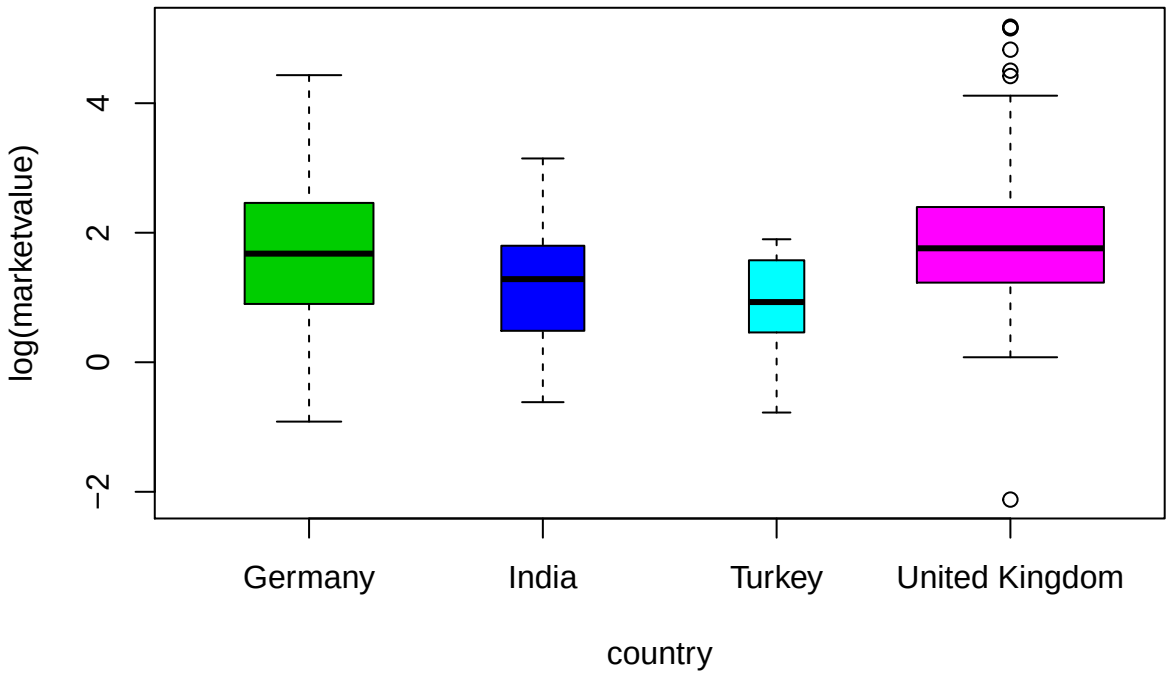

```r
library(calibrate)
profits_all = na.omit(Forbes2000$profits)  # all_profts without No data
order_profits = order(profits_all)     # index of the profitable companies
                                       #    in decreasing order
top_50 = rev(order_profits)[1:50]      # top 50 profitable companies

sales = Forbes2000$sales[top_50]       # sales of the 50 top profitable companies
assets = Forbes2000$assets[top_50]     # assets of the 50 top profitable companies
countries = Forbes2000$country[top_50] # countries where the 50 top profitable
                                       #    companies are found

plot(assets,sales,pch =1)
textxy(assets,sales, abbreviate(countries,2),col = "blue",cex=0.5)  # used to put the
                                                                    # countries where the companies are found
title(main = "Sales and Assets in billion
             USD \n of the 50 most profitable companies ", col.main = "gray")
```



Gráficos por factor

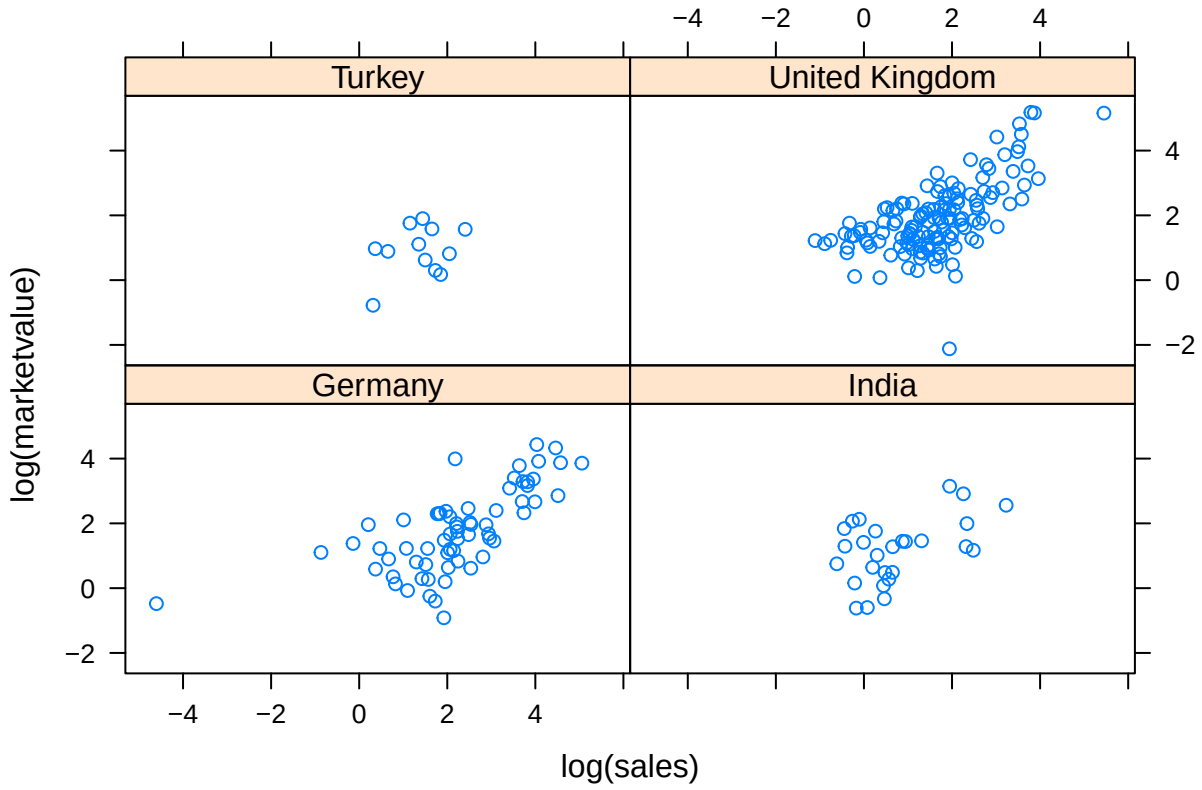
Boxplots de los logaritmos del valor de mercado para cuatro países seleccionados, el ancho de las cajas es proporcional a las raíces cuadradas del número de empresas.

```
tmp <- subset(Forbes2000,
        country %in% c("United Kingdom", "Germany",
                       "India", "Turkey"))
tmp$country <- tmp$country[,drop = TRUE]
plot(log(marketvalue) ~ country, data = tmp, col = 3:6,
       ylab = "log(marketvalue)", varwidth = TRUE)
```



Scatterplots by country

```
library(lattice)
xyplot(log(marketvalue)~log(sales)|country,data=tmp)
```



Preguntas

1. Calcular el beneficio medio de las empresas en EE.UU. y el beneficio medio de las empresas en el Reino Unido, Francia y Alemania.
2. Encuentre todas las empresas alemanas con beneficios negativos.

3. ¿A qué categoría de negocios pertenecen la mayoría de las compañías de las islas Bermuda?
4. Encuentre el valor promedio de las ventas de las compañías en cada país en el conjunto de datos de Forbes, y encuentre el número de compañías en cada país con ganancias superiores a 5 mil millones de dólares estadounidenses.

Melanoma maligno en los Estados Unidos

Fisher y Belle (1993) reportan tasas de mortalidad por melanoma maligno de la piel en hombres blancos durante el período 1950-1969, en cada estado del territorio continental de los Estados Unidos.

```
data("USmelanoma",package="HSAUR2")
```

Los datos consisten en 48 observaciones sobre las siguientes 5 variables.

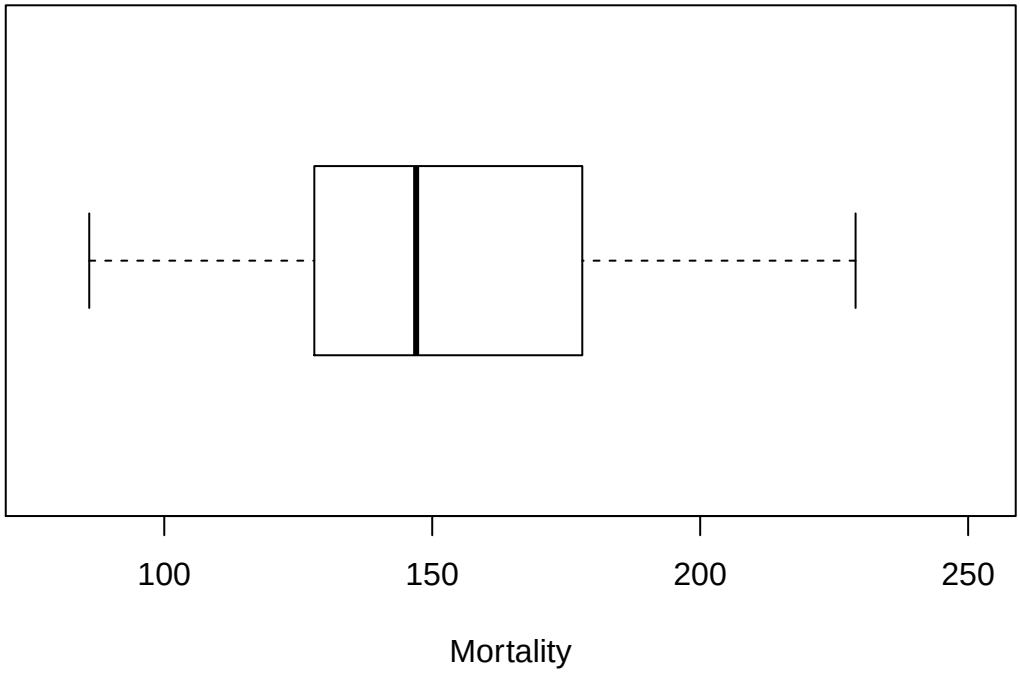
- `mortality` número de varones blancos muertos por melanoma maligno entre 1950 y 1969 por cada millón de habitantes.
- `latitude`: latitud del centro geográfico del estado.
- `longitude`: longitud del centro geográfico de cada estado.
- `ocean`: variable binaria que indica la contigüidad a un océano a niveles 'no' o 'sí'.

Gráficos de las tasas de mortalidad

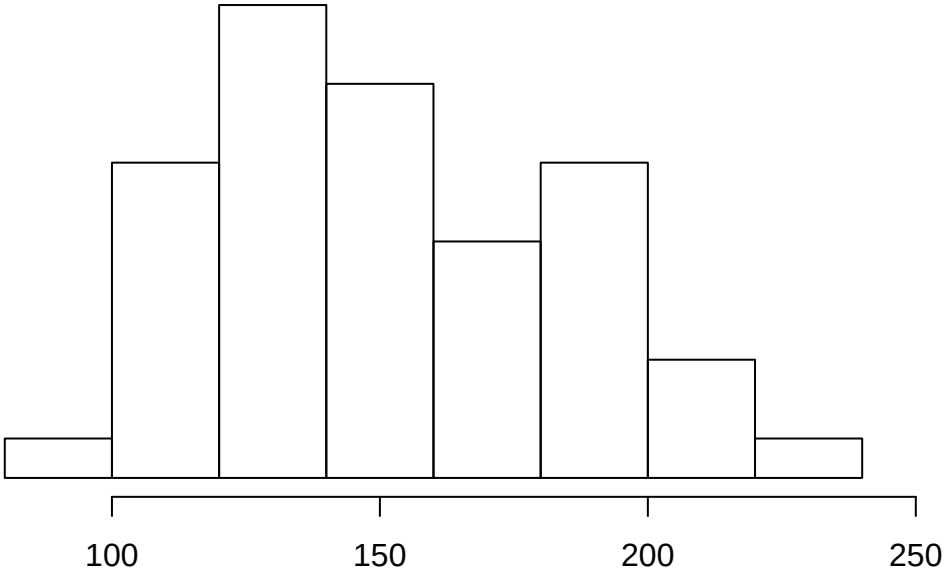
```
xr <- range(USmelanoma$mortality) * c(0.9, 1.1)
```

Boxplot

```
#layout(matrix(1:2, nrow = 2))
boxplot(USmelanoma$mortality, ylim = xr, horizontal = TRUE,xlab = "Mortality")
```

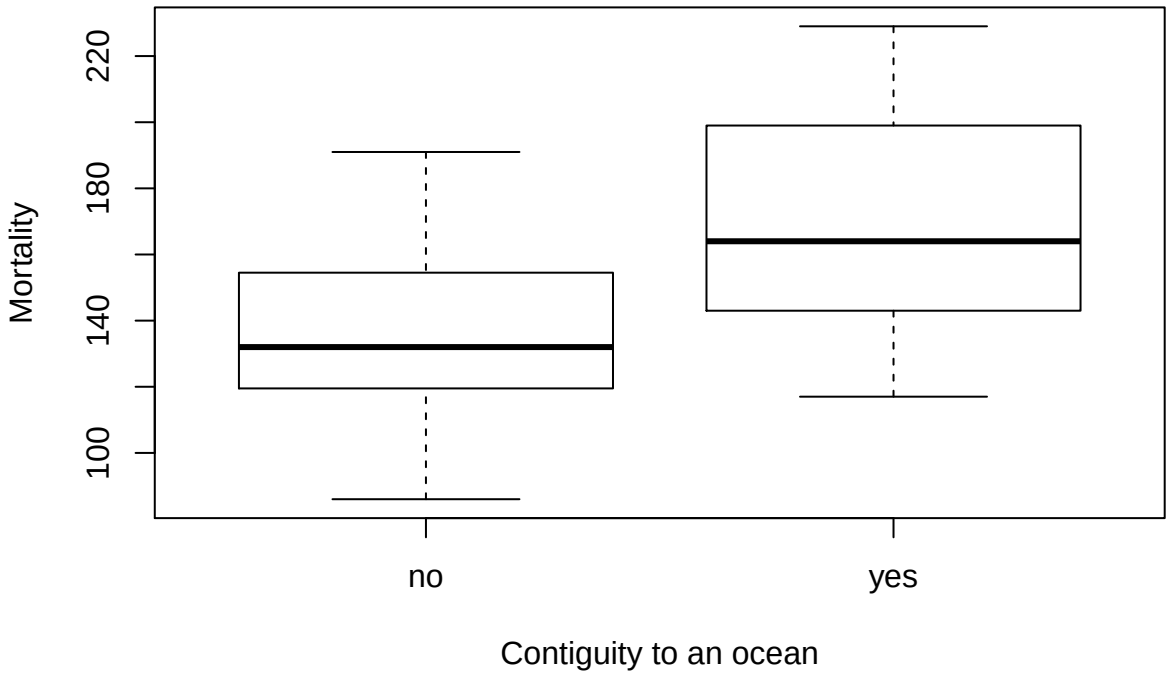



```
hist(USmelanoma$mortality, xlim = xr, xlab = "", main = "",axes = FALSE, ylab = "")
axis(1)
```



Tasas de mortalidad por melanoma maligno por contigüidad a un océano

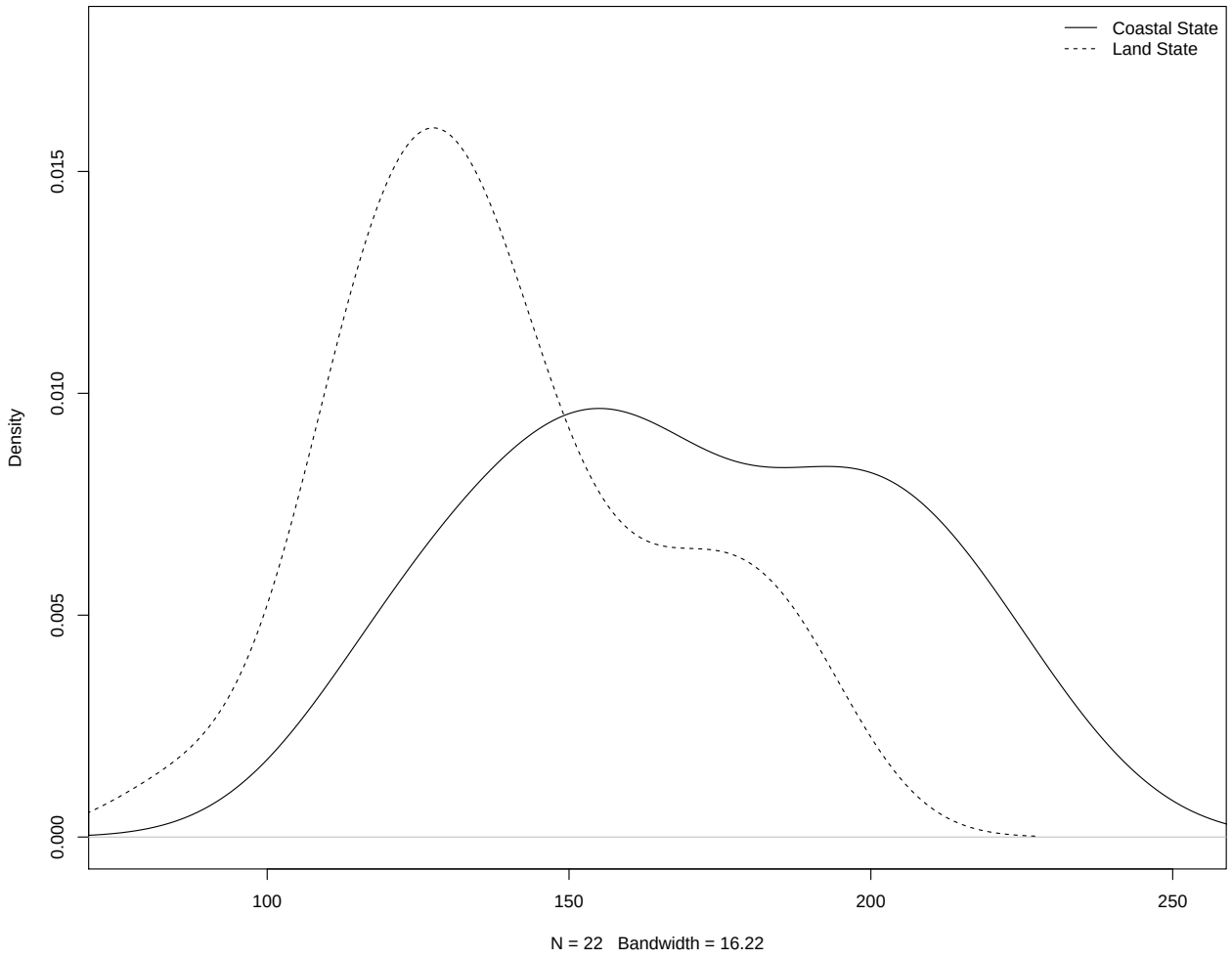
```
plot(mortality ~ ocean, data = USmelanoma, xlab = "Contiguity to an ocean", ylab = "Mortality")
```



Los histogramas a menudo pueden ser engañosos a la hora de mostrar distribuciones debido a su dependencia del número de clases elegidas. Una alternativa es estimar formalmente la función de densidad de una variable y luego graficar la estimación resultante.

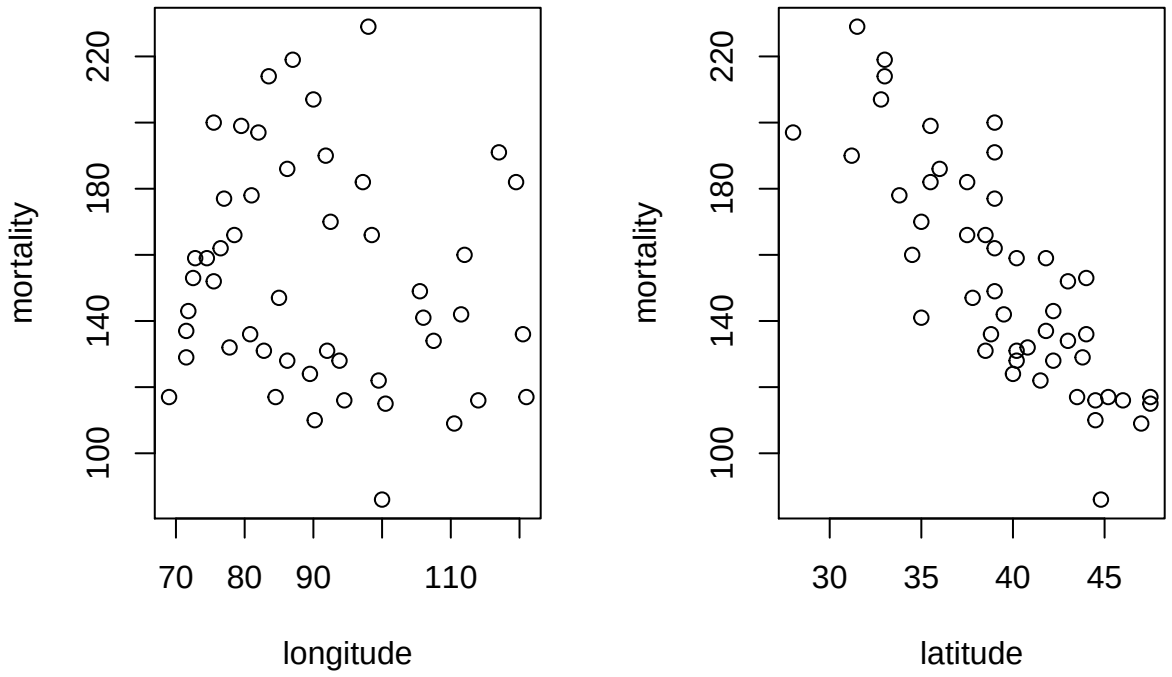
Las densidades estimadas de las tasas de mortalidad por melanoma maligno por contigüidad a un océano se ven así:

```
dyes <- with(USmelanoma, density(mortality[ocean == "yes"]))
dno <- with(USmelanoma, density(mortality[ocean == "no"]))
plot(dyes, lty = 1, xlim = xr, main = "", ylim = c(0, 0.018))
lines(dno, lty = 2)
legend("topright", lty = 1:2, legend = c("Coastal State","Land State"), bty = "n")
```



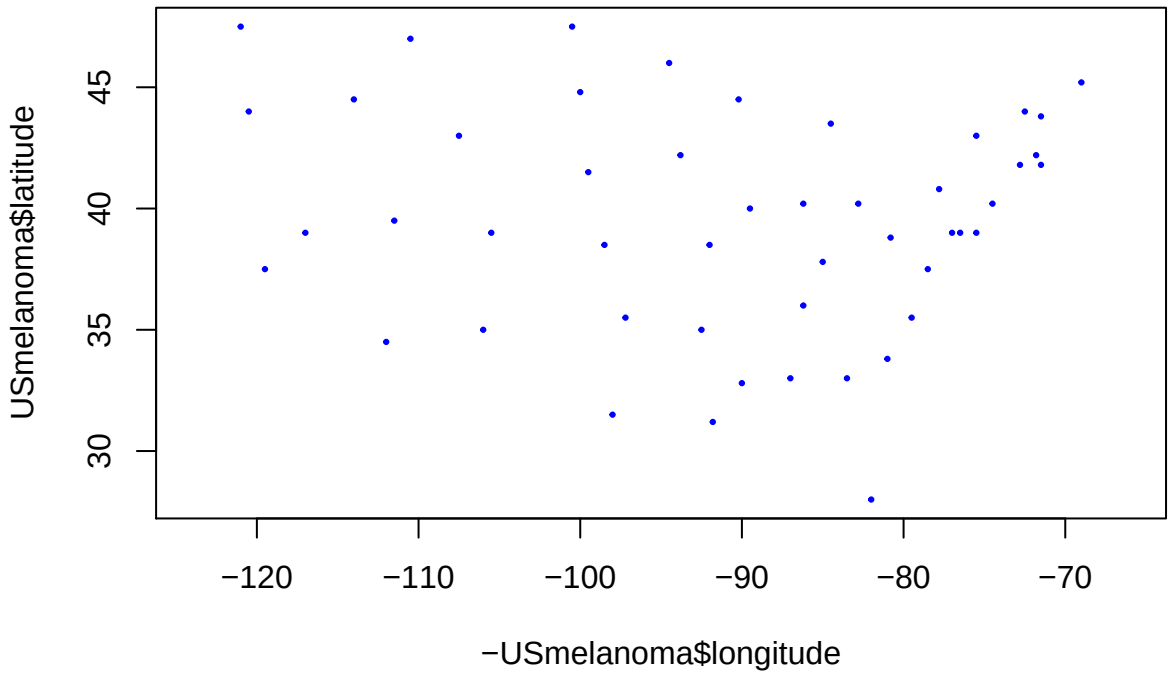
Ahora podríamos pasar a ver cómo se relacionan las tasas de mortalidad con la ubicación geográfica de un estado representada por la latitud y longitud del centro del estado.

```
layout(matrix(1:2, ncol = 2))
plot(mortality ~ -longitude, data = USmelanoma)
plot(mortality ~ latitude, data = USmelanoma)
```

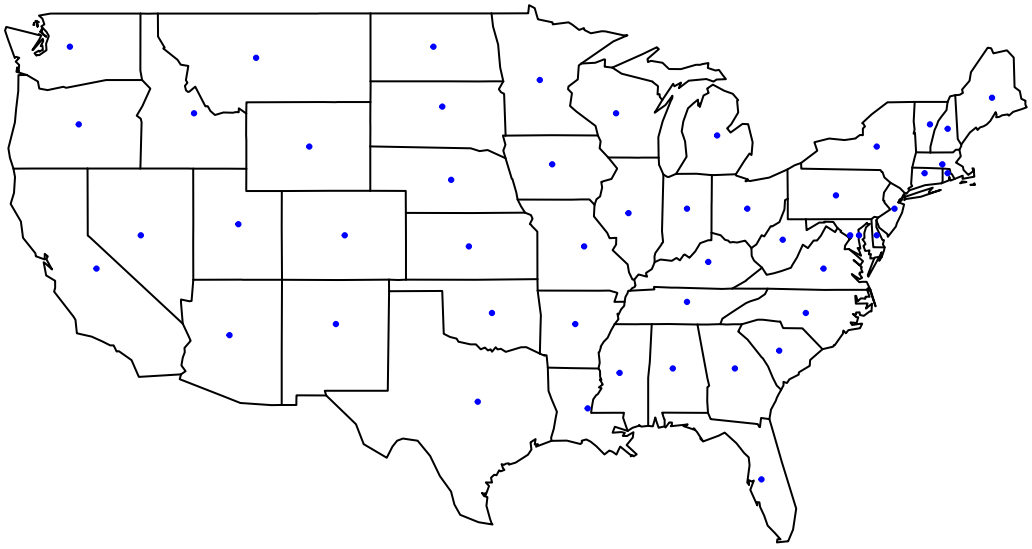


Los datos contienen la longitud y latitud de los centroides.

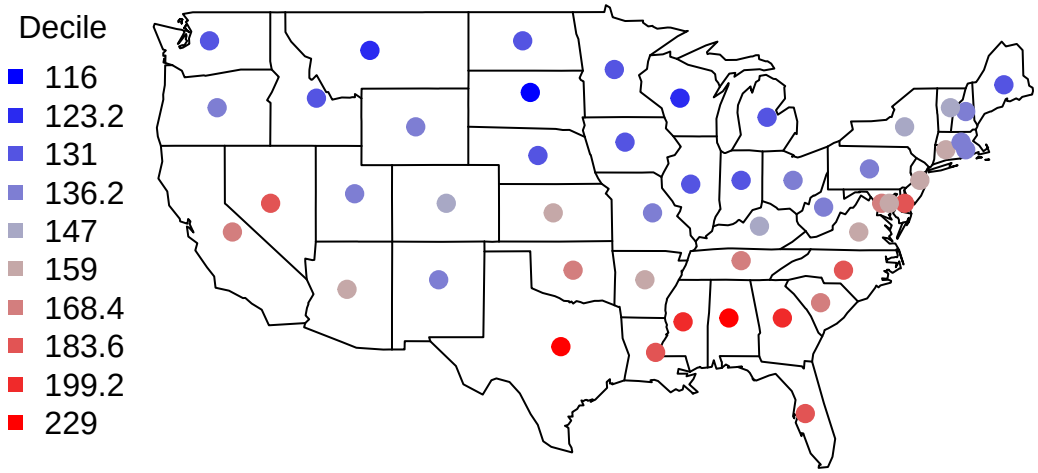
```
plot(-USmelanoma$longitude,USmelanoma$latitude,asp=1.5,cex=.3,pch=19,col="blue")
```



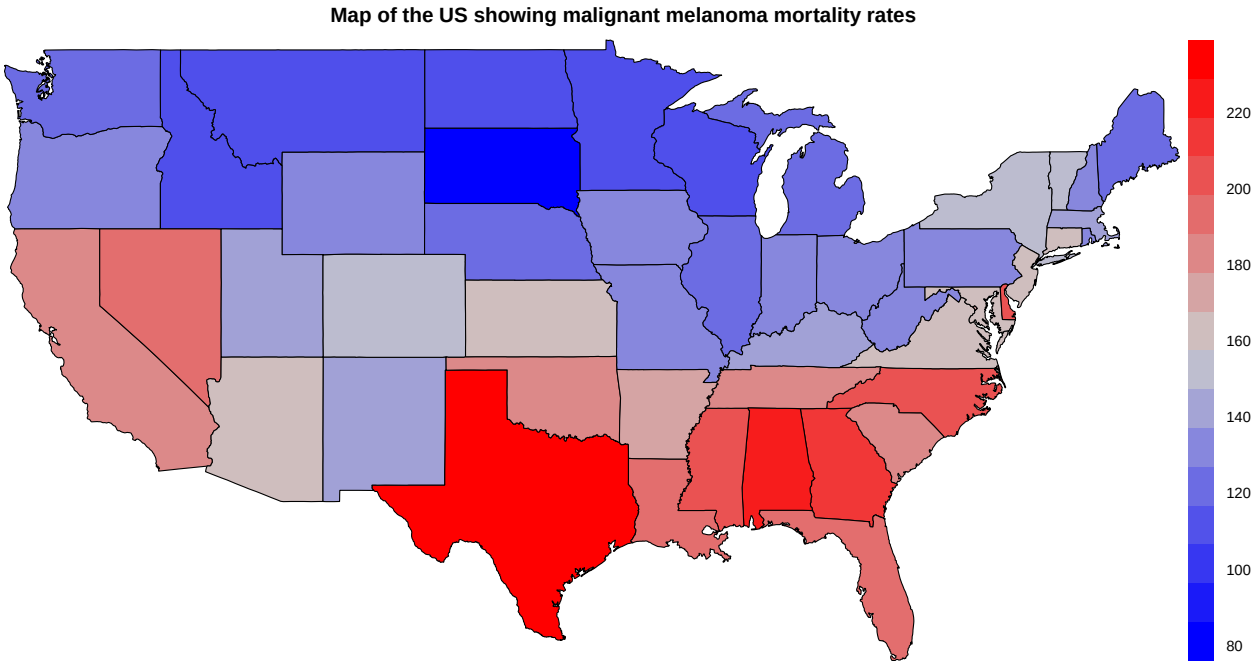

```
library("sp")
library("maps")
library("maptools")
library("RColorBrewer")
map("state")
points(-USmelanoma$longitude,USmelanoma$latitude,asp=1.5,cex=.3,pch=19,col="blue")
```



```r
# Crear una función para generar una paleta de colores continua
rbPal <- colorRampPalette(c('blue','grey','red'))
# Esto añade una columna de valores de color
# basado en los valores de y
USmelanoma$Col <- (rbPal(10)[as.numeric(cut(USmelanoma$mortality,breaks = 10))])
map("state",xlim=c(-135,-65))
points(-USmelanoma$longitude,USmelanoma$latitude,col=USmelanoma$Col,asp=1.5,pch=19,cex=1.2)
legend("topleft",title="Decile",legend=quantile(USmelanoma$mortality,seq(0.1,1,l=10)),col =rbPal(10),pch=15,cex=1.,box.col = NA)
```




```r
states <- map("state", plot = FALSE, fill = TRUE)
IDs <- sapply(strsplit(states$names, ":"), function(x) x[1])
rownames(USmelanoma) <- tolower(rownames(USmelanoma))

us1 <- map2SpatialPolygons(states, IDs=IDs,proj4string = CRS("+proj=longlat +datum=WGS84"))
us2 <- SpatialPolygonsDataFrame(us1, USmelanoma)

col <- colorRampPalette(c('blue', 'gray80','red'))

spplot(us2, "mortality", col.regions = col(200),par.settings = list(axis.line = list(col =  'transparent')),main="Map of the US
```

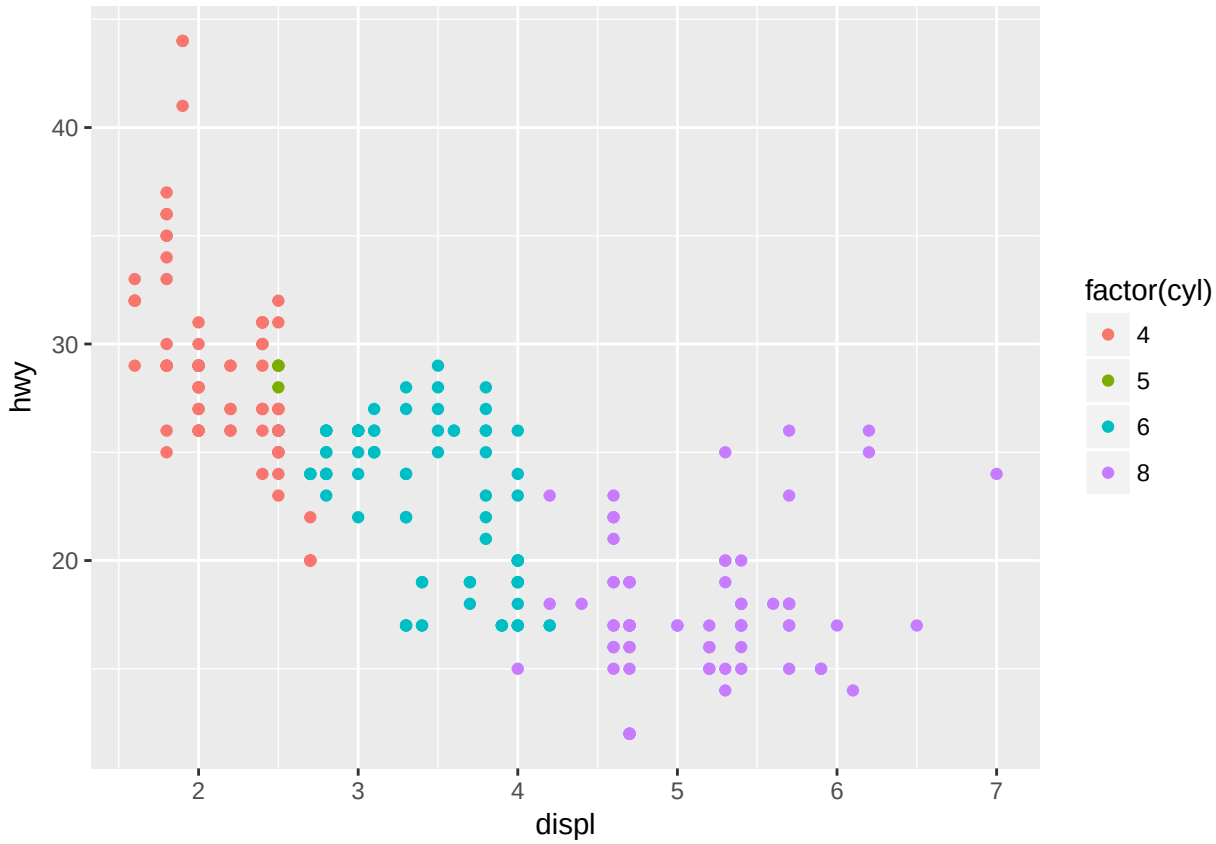


Gráficos avanzados con la librería `ggplot2`

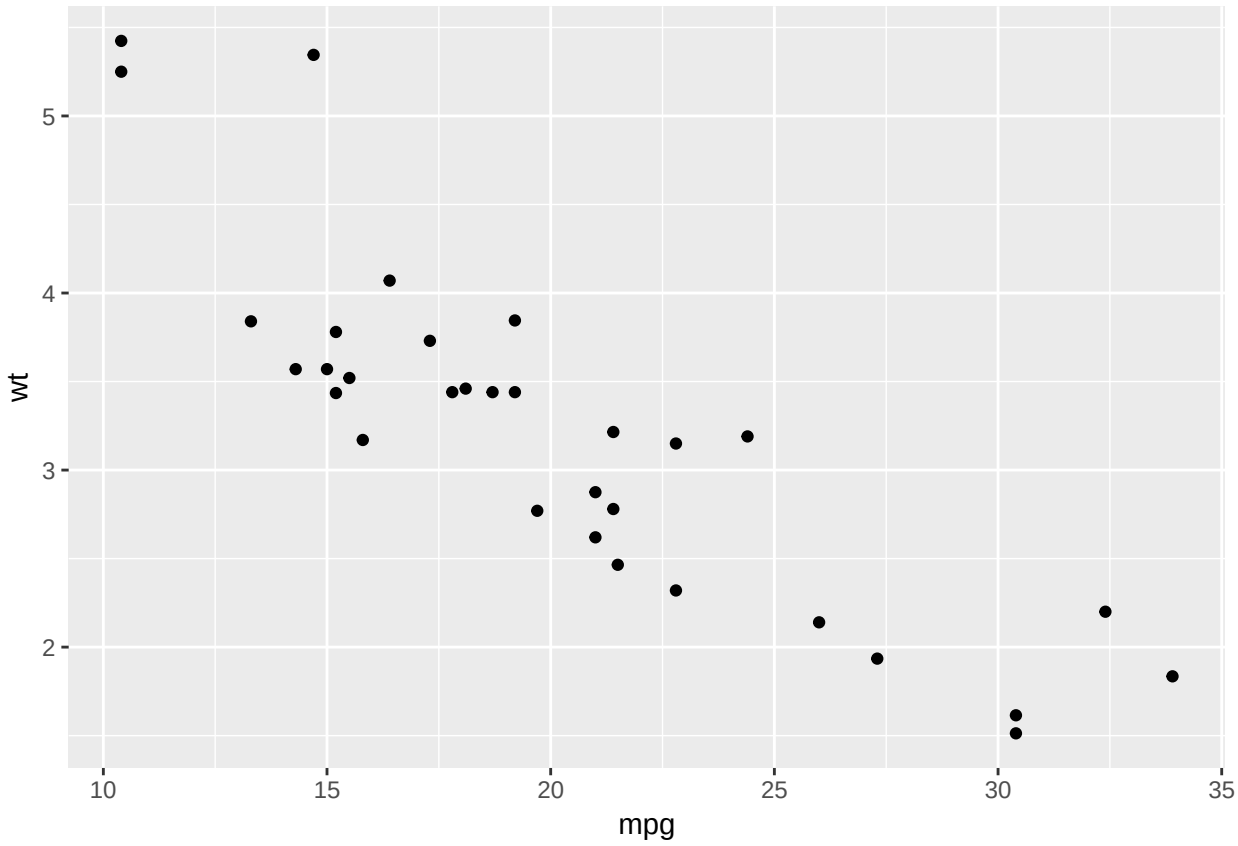
- Toma como referencia una metodología de visualización de datos llamada The Grammar of Graphics, (Wilkinson, 2005).
- La idea es describir los mapeos visuales para poder armar visualizaciones complejas sin preocuparnos por la parte dificil.
- Gramática consistente basada en grammar of graphics (Wilkinson, 2005)
- Librería muy flexible
- Mantenimiento muy activo de la librería
- Gran lista de distribución y con mucha participación
- Es posible crear gráficos visualmente atractivos y elegantes
- Simple gestión de leyendas

Más información

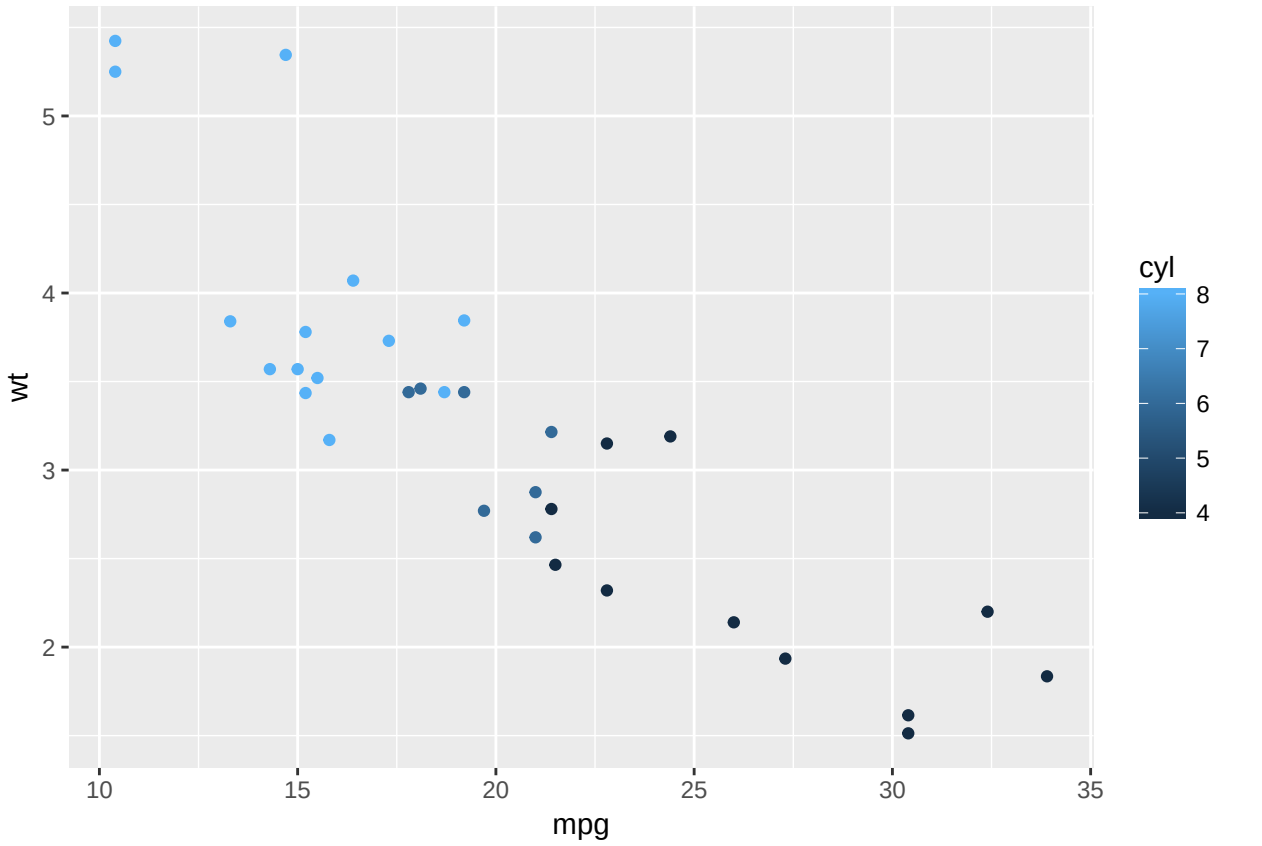
```
library(ggplot2)
?qplot
qplot(displ, hwy, data = mpg, colour = factor(cyl))
```



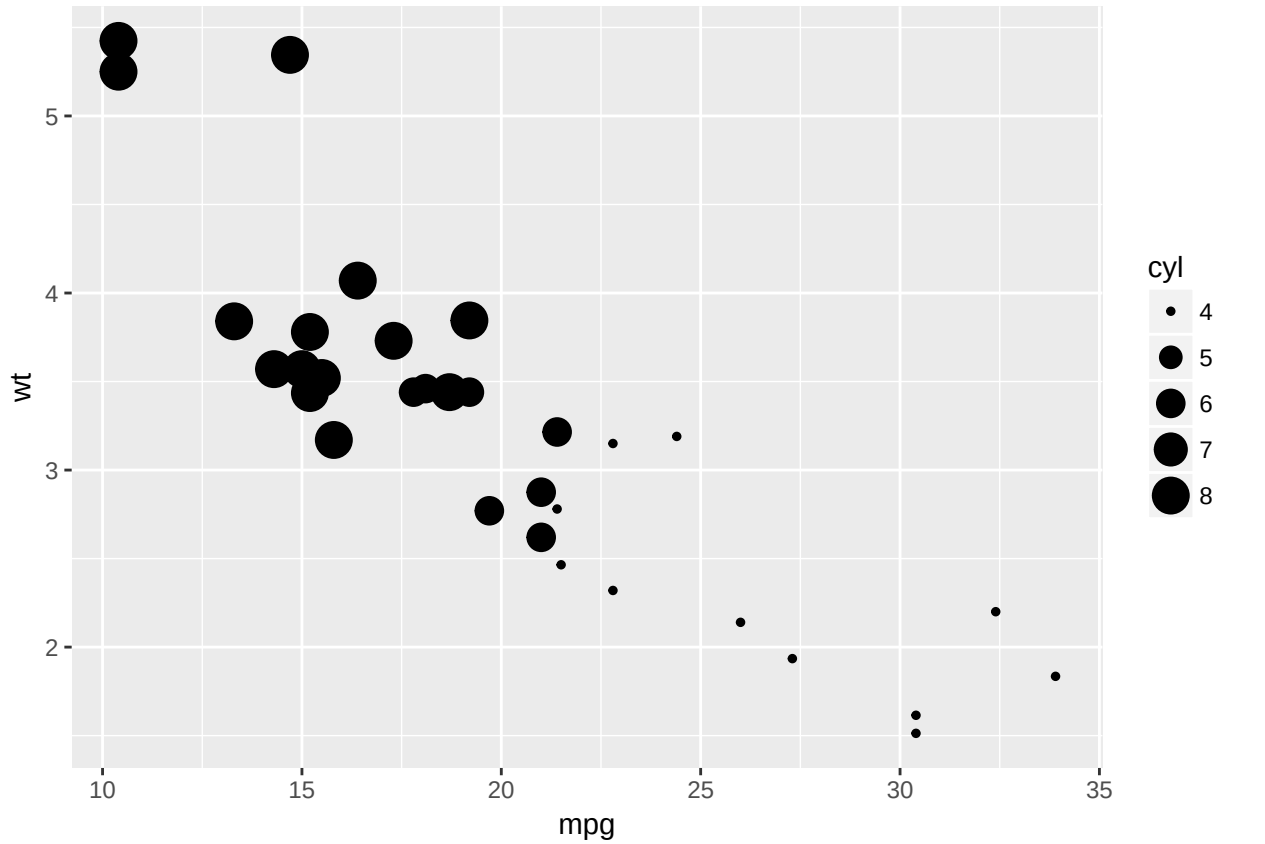

```
qplot(mpg, wt, data = mtcars)
```



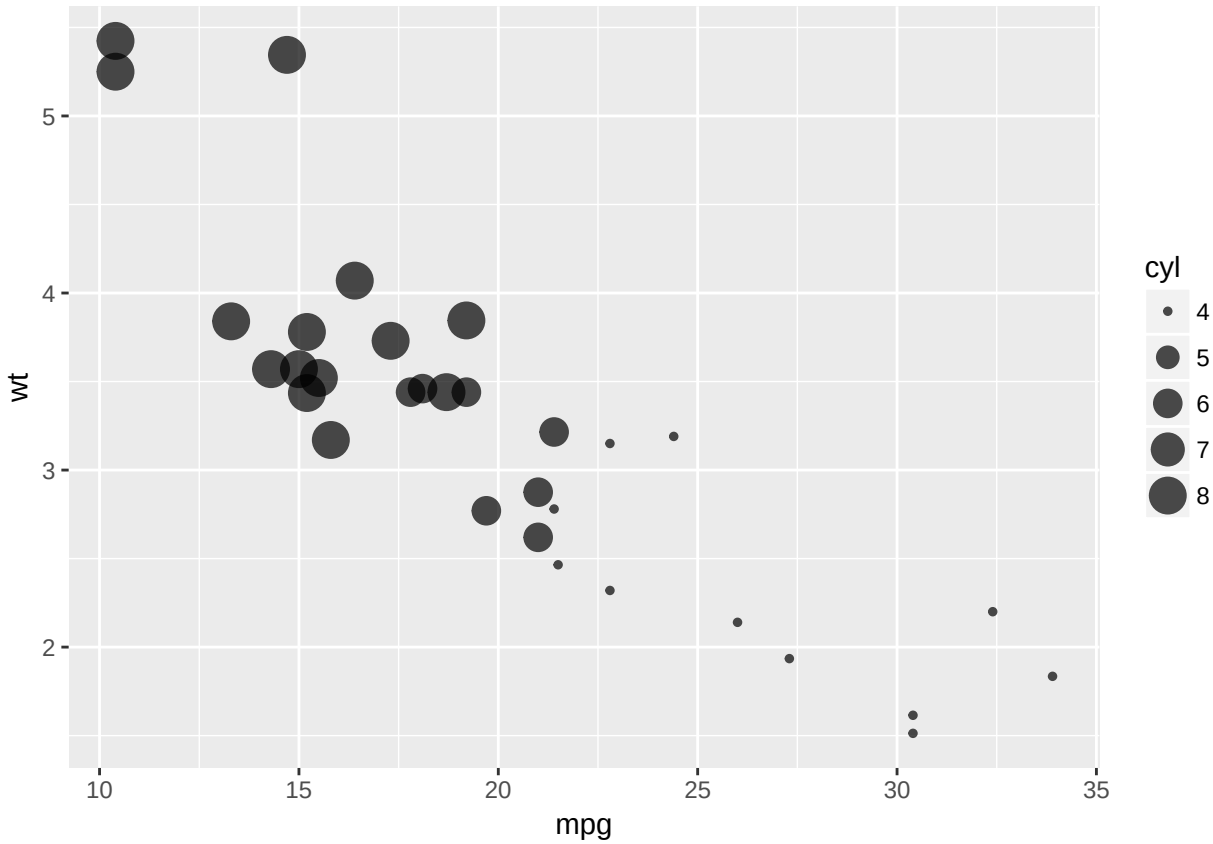

```
qplot(mpg, wt, data = mtcars, colour = cyl)
```



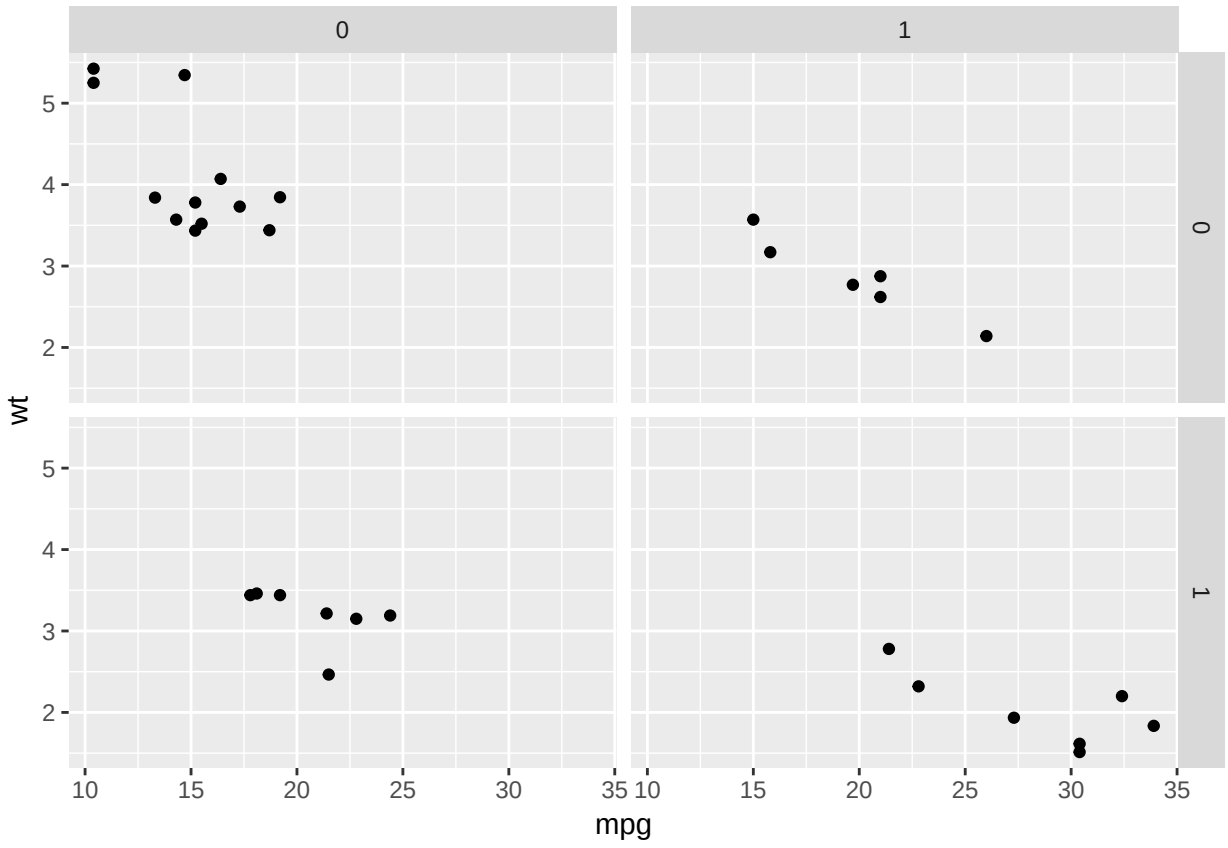

```
qplot(mpg, wt, data = mtcars, size = cyl)
```



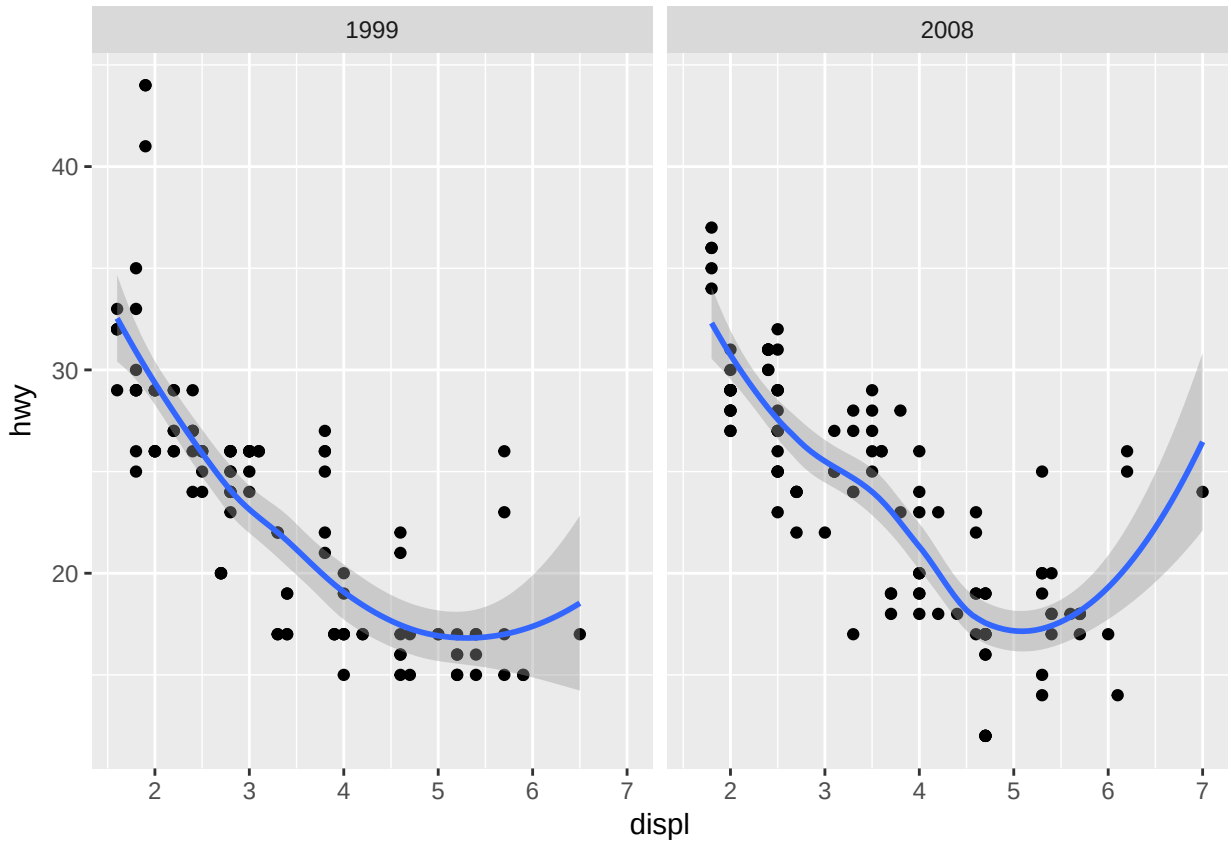

```
qplot(mpg, wt, data = mtcars, size = cyl, alpha = I(0.7))
```



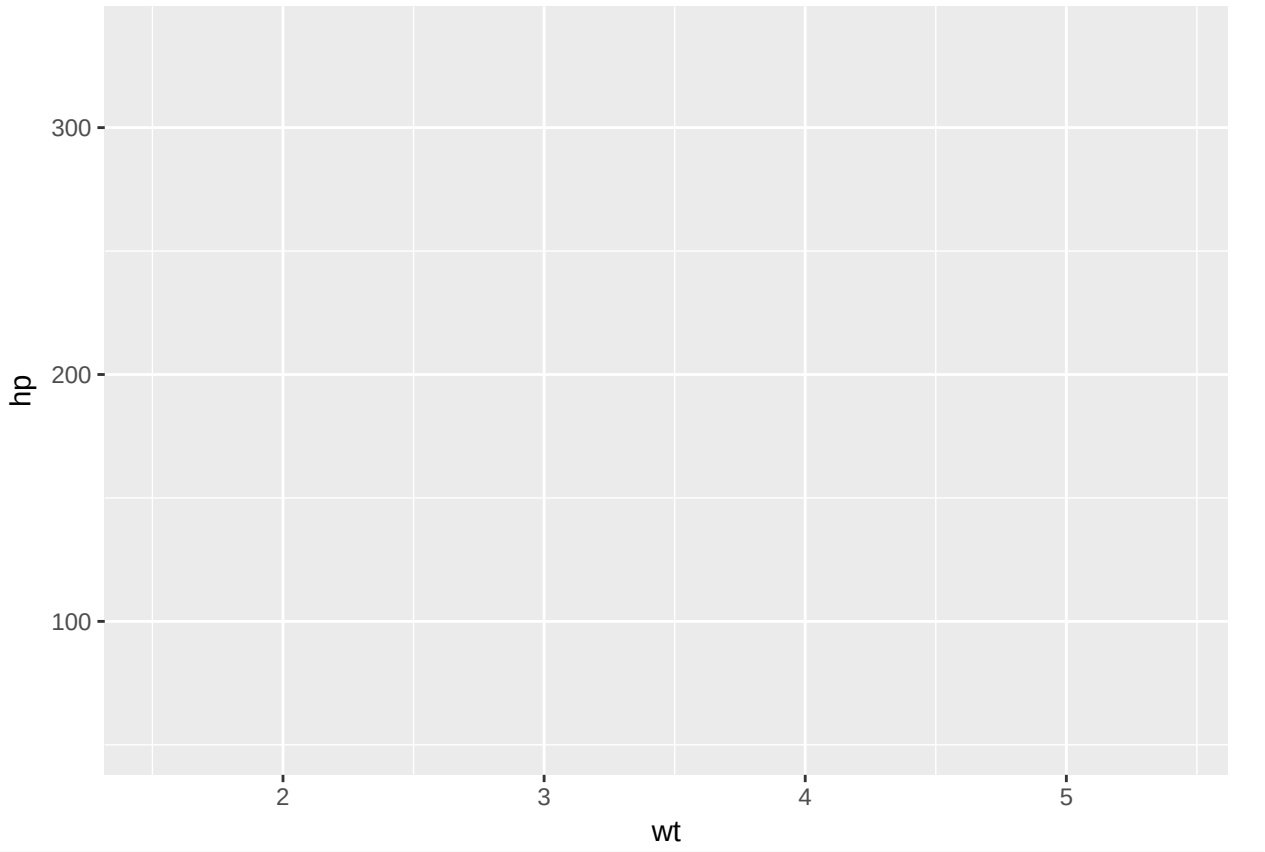

```
qplot(mpg, wt, data = mtcars, facets = vs ~ am)
```



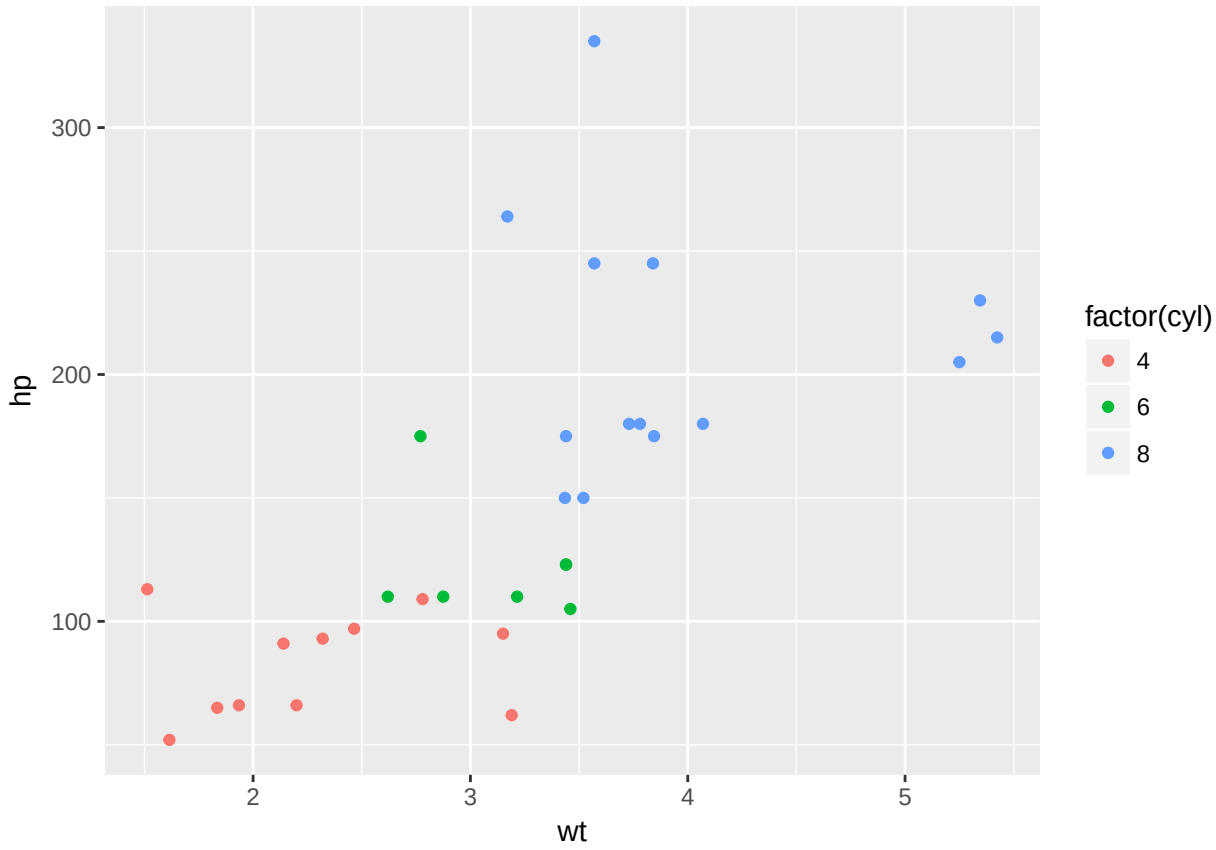

```
qplot(displ, hwy, data=mpg, facets = . ~ year) + geom_smooth()
```



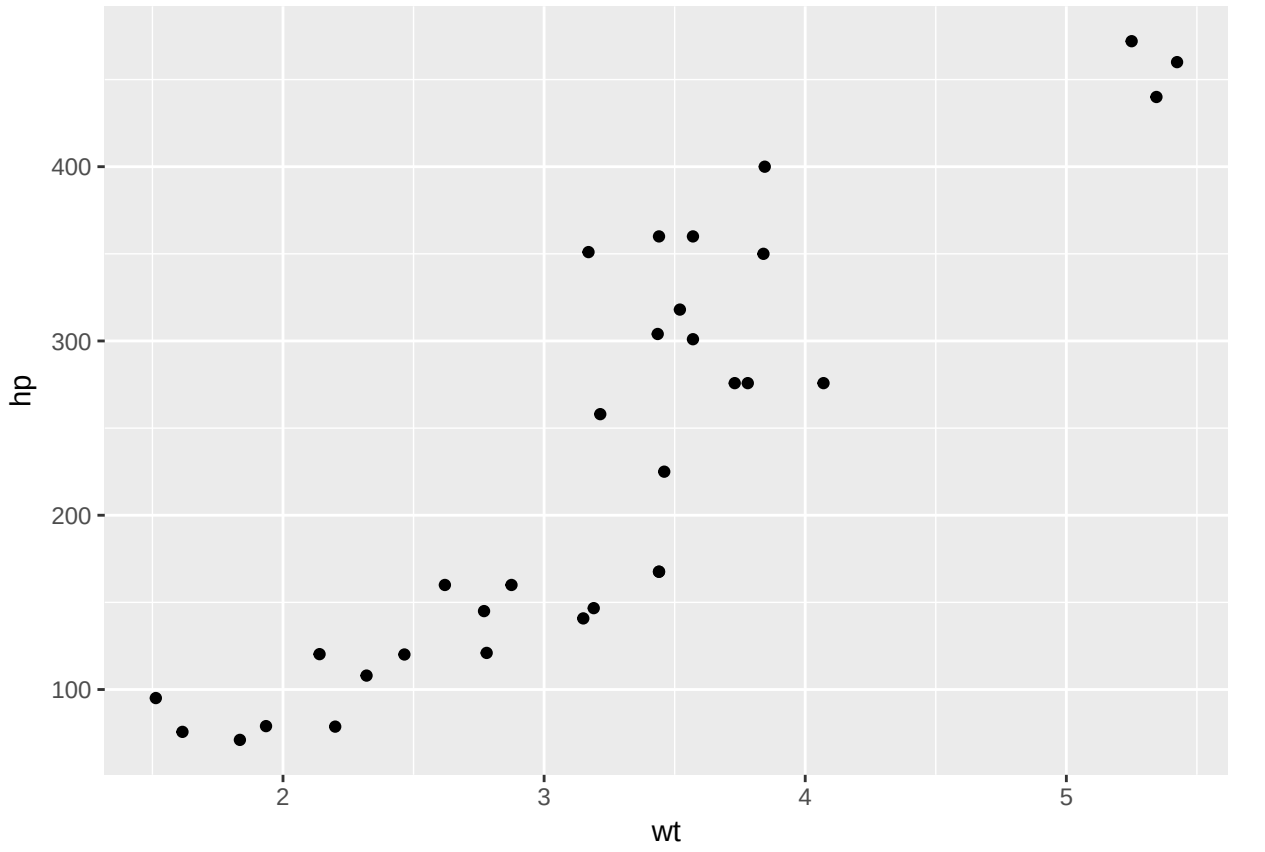

```
p <- ggplot(mtcars)
p <- p + aes(wt, hp)
p
```



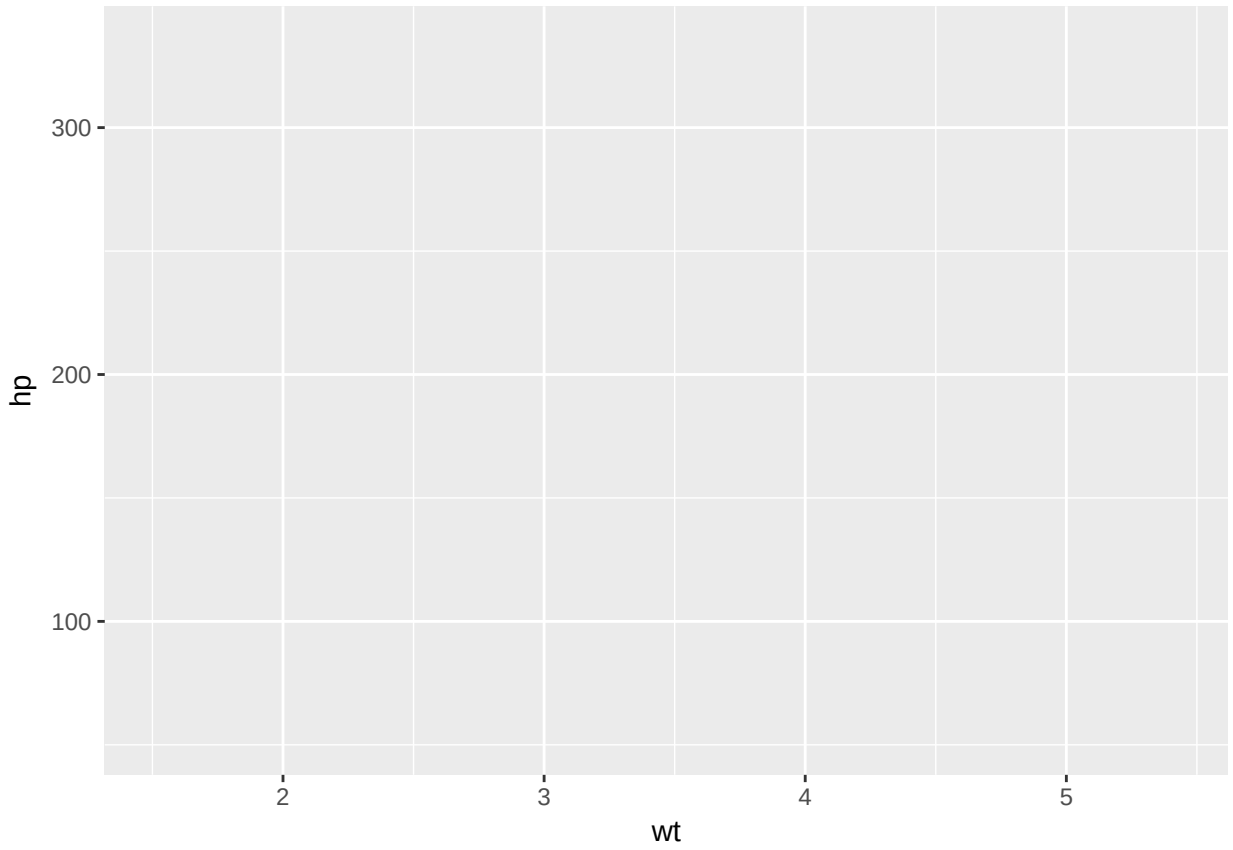

```
p + geom_point(aes(colour = factor(cyl)))
```



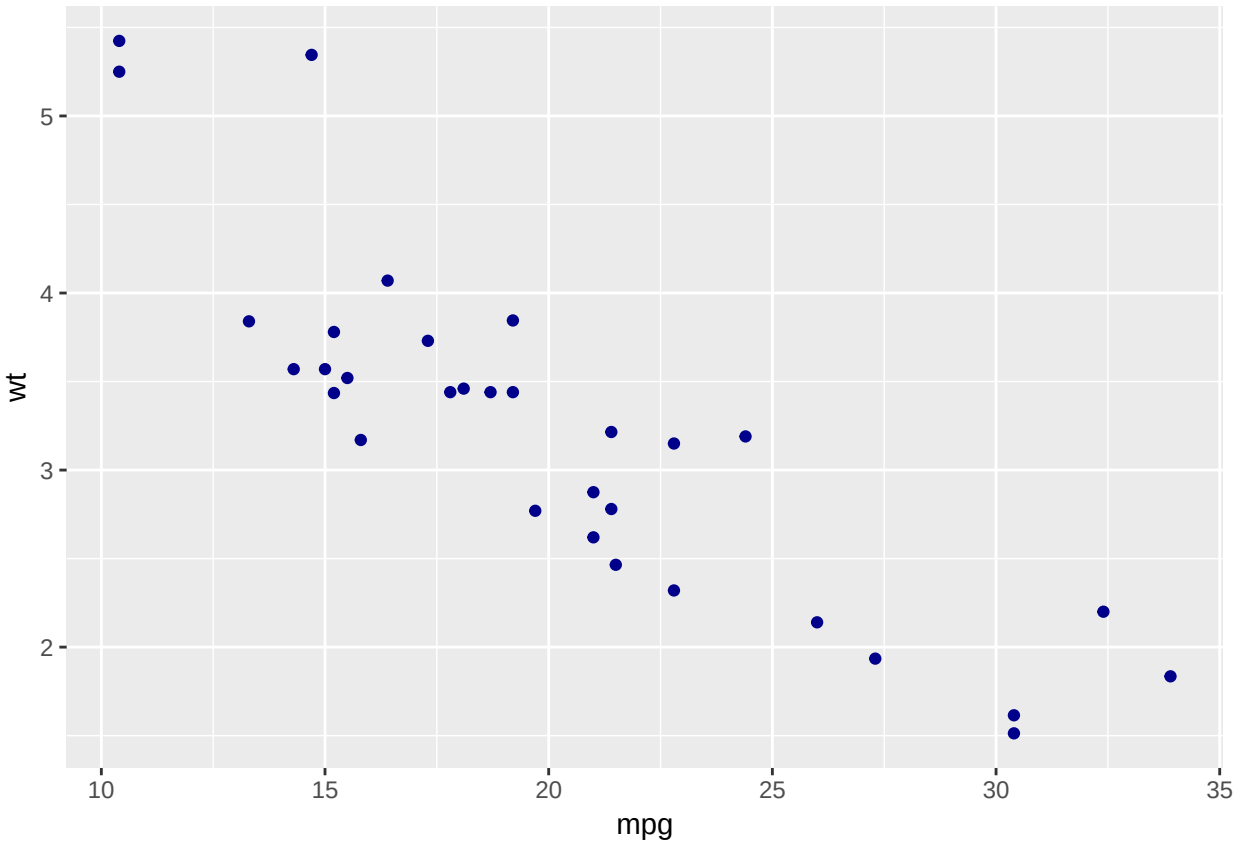

```
p + geom_point(aes(y = disp))
```




```
p
```




```
p <- ggplot(mtcars, aes(mpg, wt))
p + geom_point(colour = "darkblue")
```

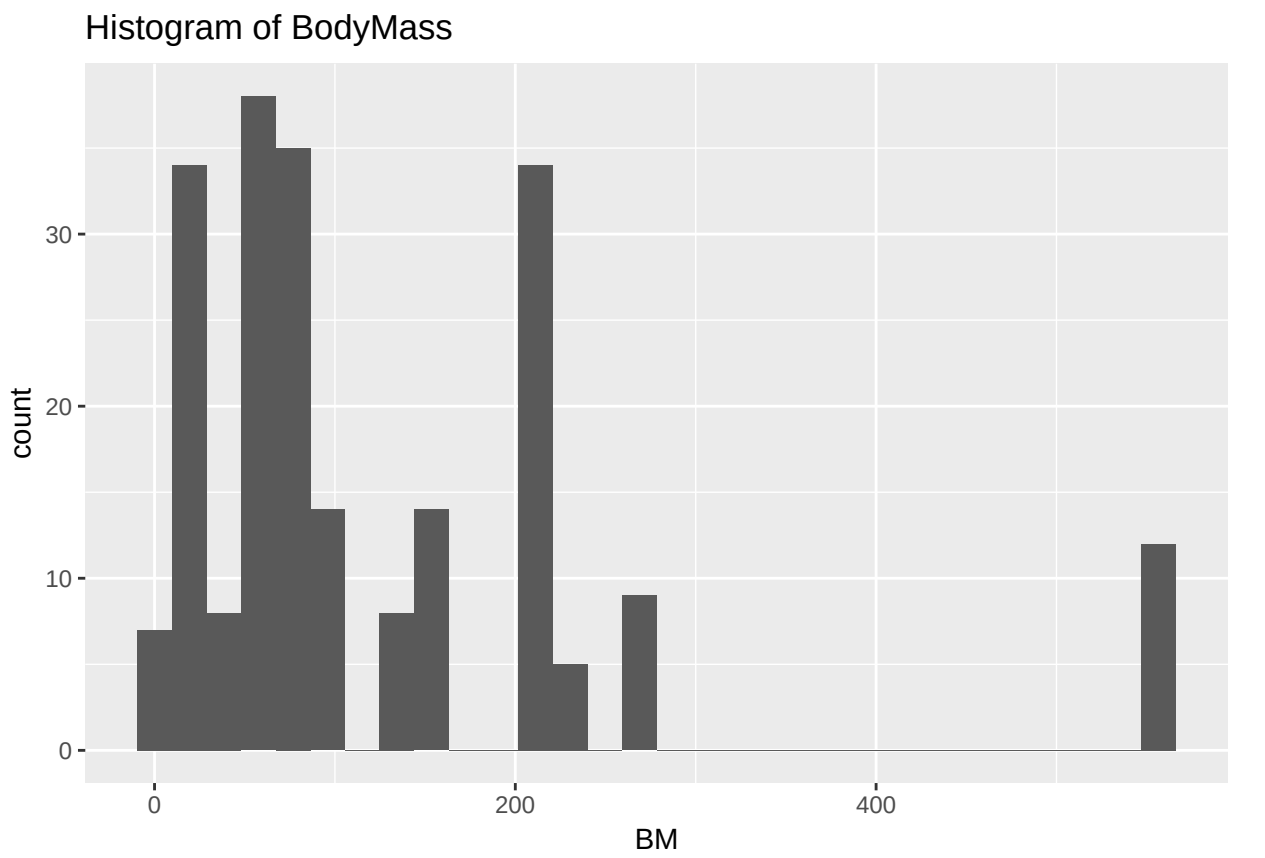



```
filepath <- "http://idaejin.github.io/bcam-courses/azti-2016/introR/data/ggplot2_data.txt"

myData<-read.table(file=url(filepath),header=TRUE,sep="\t")

str(myData)
```

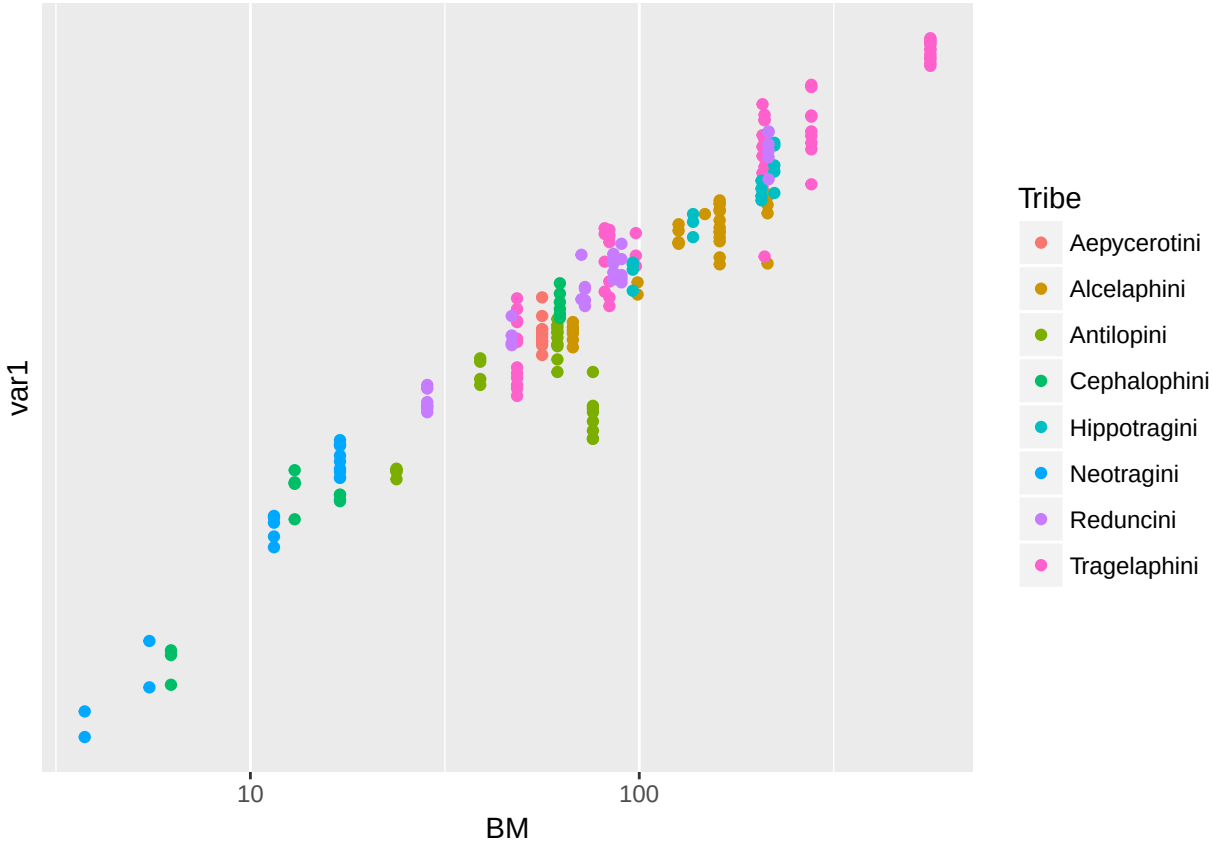
```
## 'data.frame':    218 obs. of  4 variables:
##  $ Tribe: Factor w/ 8 levels "Aepycerotini",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Hab  : Factor w/ 4 levels "F","H","L","O": 3 3 3 3 3 3 3 3 3 3 ...
##  $ BM   : num  56.2 56.2 56.2 56.2 56.2 ...
##  $ var1 : num  36.5 40.9 37 36.2 36.6 37.7 37.3 39 37.7 35.3 ...
```

```
qplot(data=myData,x=BM,main="Histogram of BodyMass")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qplot(data=myData,x=BM,y=var1,log="xy",color=Tribe)
```



Maps

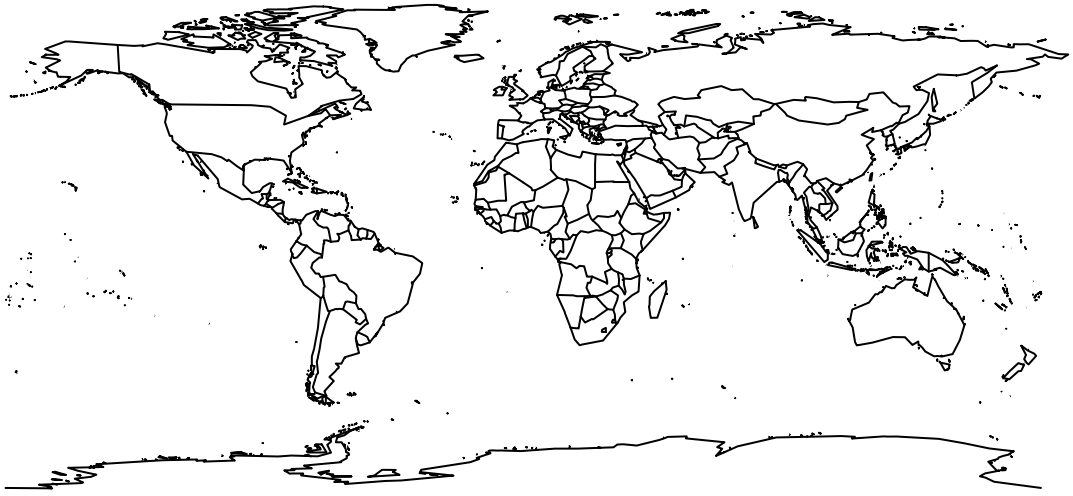
Paquetes para Regresión Espacial / Geoestadística / Métodos de Patrones de Puntos Espaciales

- `sp`, `maptools`, `spatstat`
- `maps`

```
library(maps)
```

Sintaxis básica

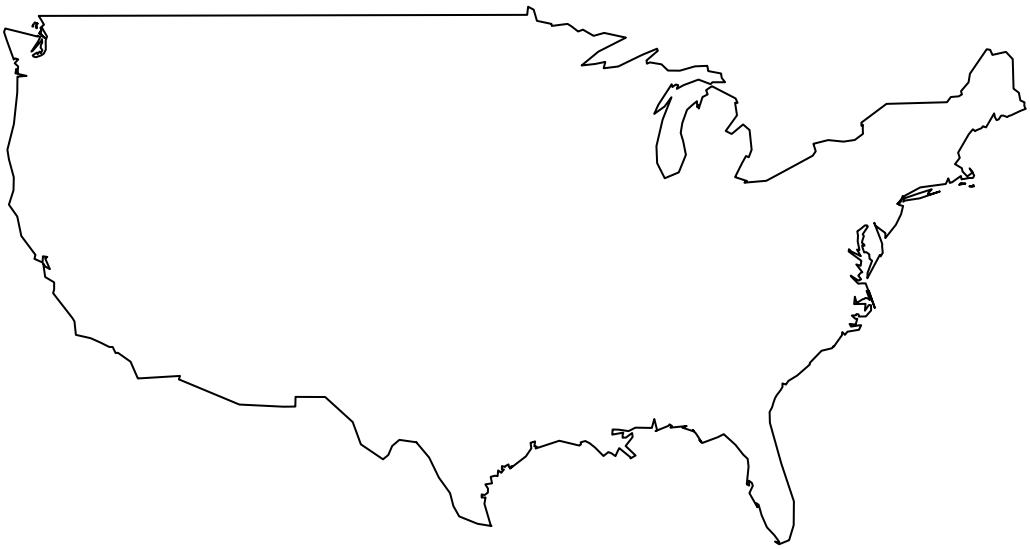
```
map(database = "world",regions=".")
```



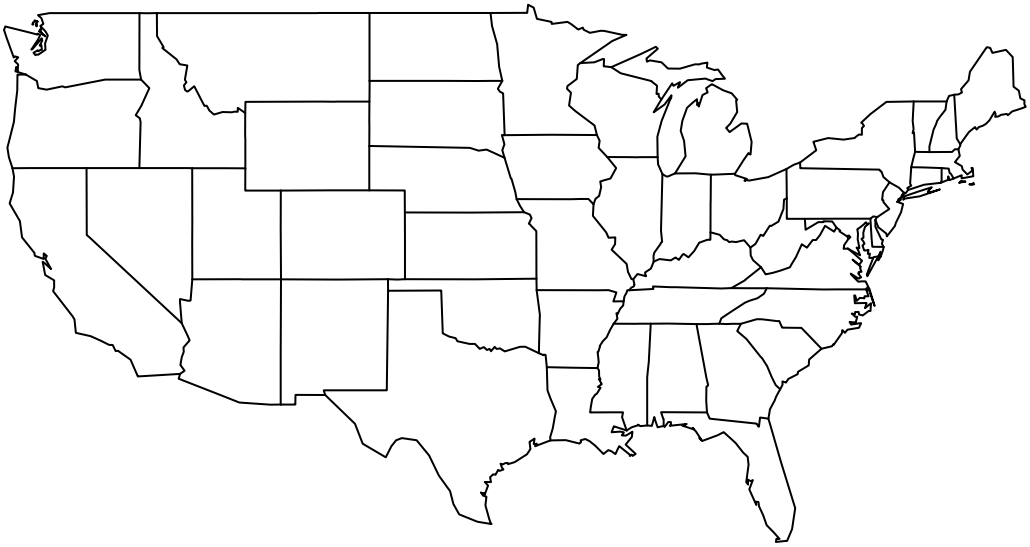
Hay bases de datos disponibles para EE.UU., Francia, Italia y Nueva Zelanda. Para otros países, es necesario importar una base de datos con

el mapa correspondiente.

```
map(database = "usa")
```

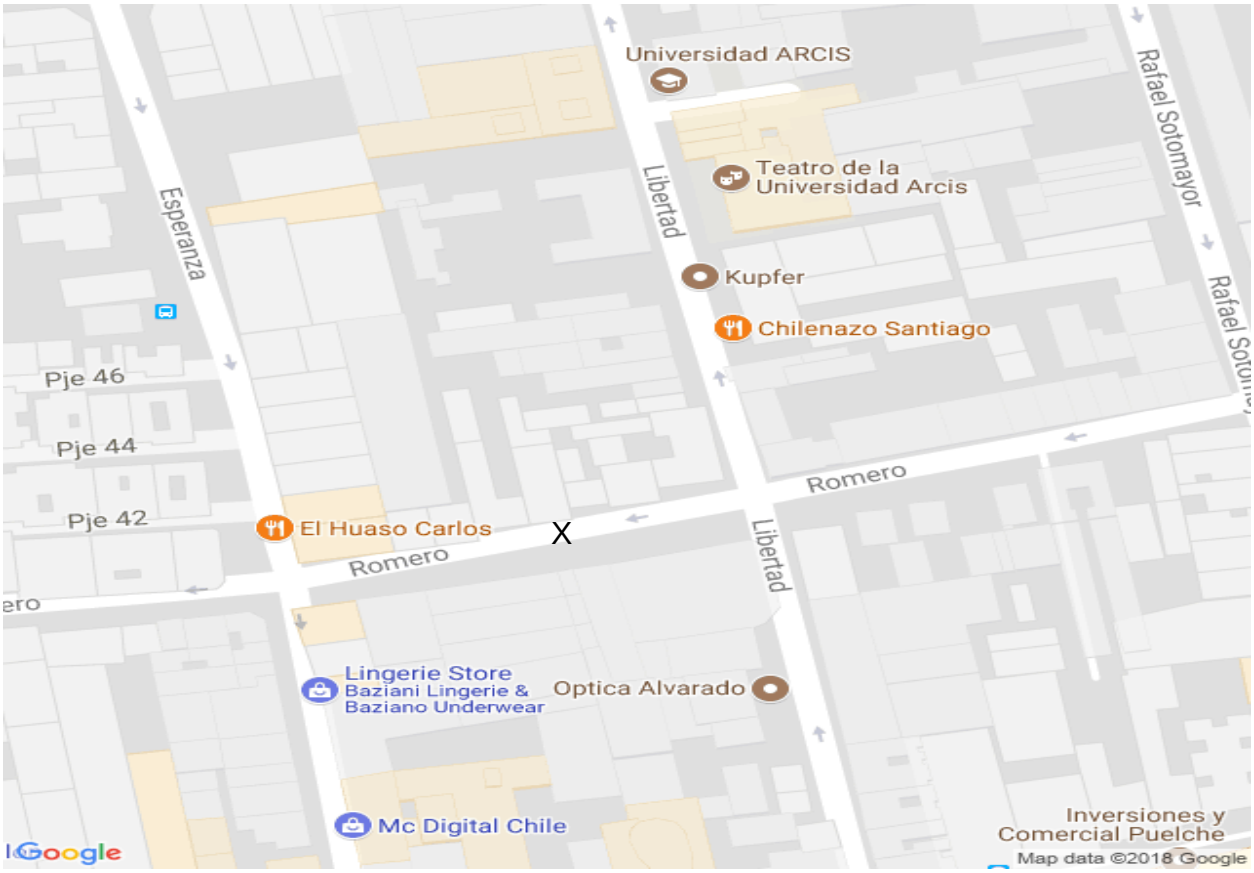


```
map("state")
```



Con el paquete `RgoogleMaps`, puedes dibujar un fondo desde Google Maps!

```
library(RgoogleMaps)
lat <- -33.447487
lon <- -70.673676
center <- c(lat, lon)
zoom <- 18
MyMap <- GetMap(center=center, zoom=zoom)
PlotOnStaticMap(MyMap)
text(lat,lon, "X")
```



ggmap ofrece capacidades gráficas como 'ggplot2".

```
library(ggmap)
geocode("Alameda, Santiago de Chile, Chile")
qmap("Santiago, Chile", zoom = 14)
mapdist("Valparaíso", "Santiago")
route("Valparaíso, Chile", "Santiago, Chile", alternatives = FALSE)
```

Ejemplo de uso de qmap y ggplot2

```
desde <- 'Valparaíso, Chile'
hasta <- 'Santiago, Chile'

rutas <- route(desde, hasta, alternatives = FALSE)
head(rutas)

ggplot() +
  geom_segment(aes(x = startLon, y = startLat, xend = endLon, yend = endLat, colour = route), size = 1.5, data = rutas)
```