8
Phân tích số liệu bằng biểu đồ

Yếu tố thị giác rất quan trọng. Người Trung Quốc có câu "một biểu đồ có giá trị bằng cả vạn chữ viết". Quả thật, biểu đồ tốt có khả năng gây ấn tượng cho người đọc báo khoa học rất lớn, và thường có giá trị đại diện cho cả công trình nghiên cứu. Vì thế biểu đồ là một phương tiện hữu hiệu nhất để nhấn mạnh thông điệp của bài báo. Biểu đồ thường được sử dụng để thể hiện xu hướng và kết quả cho từng nhóm, nhưng cũng có thể dùng để trình bày dữ kiện một cách gọn gàng. Các biểu đồ dễ hiểu, nội dung phong phú là những phương tiện vô giá. Do đó, nhà nghiên cứu cần phải suy nghĩ một cách sáng tạo cách thể hiện số liệu quan trọng bằng biểu đồ. Vì thế, phân tích biểu đồ đóng một vai trò cực kì quan trọng trong phân tích thống kê. Có thể nói, không có đồ thị là phân tích thống kê không có nghĩa.

Trong ngôn ngữ R có rất nhiều cách để thiết kế một biểu đồ gọn và đẹp. Phần lớn những hàm để thiết kế biểu đồ có sẵn trong R, nhưng một số loại biểu đồ tinh vi và phức tạp khác có thể thiết kế bằng các package chuyên dụng như `lattice` hay `trellis` có thể tải từ website của R. Trong chương này tôi sẽ chỉ cách vẽ các biểu đồ thông dụng bằng cách sử dụng các hàm phổ biến trong R.

8.1 Môi trường và thiết kế biểu đồ

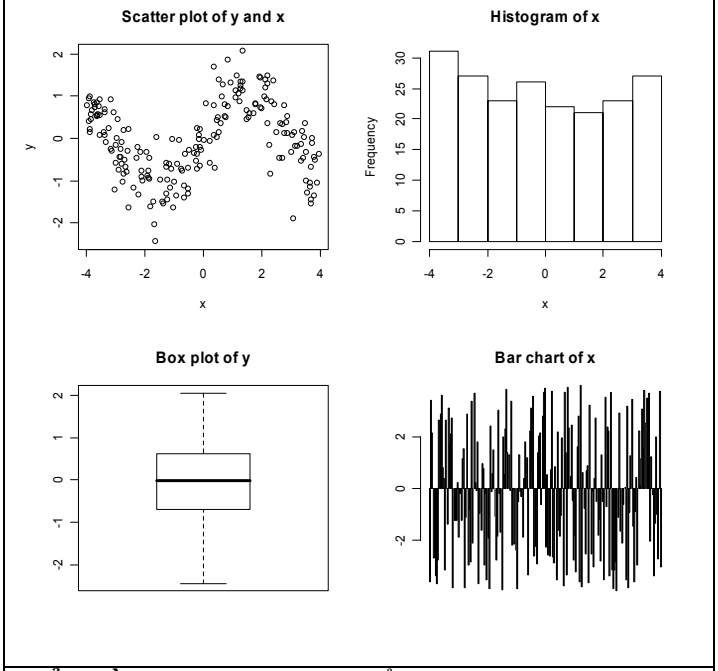
8.1.1 Nhiều biểu đồ cho một cửa sổ (windows)

Thông thường, R vẽ một biểu đồ cho một cửa sổ. Nhưng chúng ta có thể vẽ nhiều biểu đồ trong một cửa sổ bằng cách sử dụng hàm `par`. Chẳng hạn như `par(mfrow=c(1,2))` có hiệu năng chia cửa sổ ra thành 1 dòng và hai cột, tức là chúng ta có thể trình bày hai biểu đồ kề cạnh bên nhau. Còn `par(mfrow=c(2,3))` chia cửa sổ ra thành 2 dòng và 3 cột, tức chúng ta có thể trình bày 6 biểu đồ trong mộ cửa sổ. Sau khi đã vẽ xong, chúng ta có thể quay về với "chế độ" 1 cửa sổ bằng lệnh `par(mfrow=c(1,1)`.

Ví dụ sau đây tạo ra một dữ liệu gồm hai biến *x* và *y* bằng phương pháp mô phỏng (tức số liệu hoàn toàn được tạo ra bằng R). Sau đó, chúng ta chia cửa sổ thành 2 dòng và 2 cột, và trình bày bốn loại biểu đồ từ dữ liệu được mô phỏng:

```
> par(mfrow=c(2,2))
> N <- 200
> x <- runif(N, -4, 4)
> y <- sin(x) + 0.5*rnorm(N)
> plot(x,y, main="Scatter plot of y and x")
> hist(x, main="Histogram of x")
> boxplot(y, main="Box plot of y")
```

```
> barplot(x, main="Bar chart of x")
> par(mfrow=c(1,1))
```



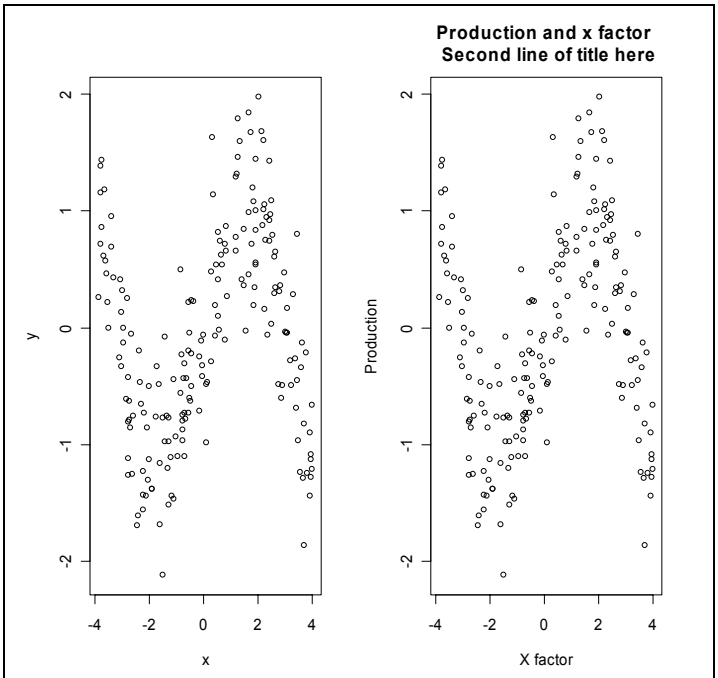
Biểu đồ 1. Cách chia cửa sổ thành 2 dòng và 2 cột và trình bày 4 biểu đồ trong cùng một cửa số.

8.1.2 Đặt tên cho trục tung và trục hoành

Biểu đồ thường có trục tung (y-axis) và trục hoành. Vì dữ liệu thường được gọi bằng các chữ viết tắt, cho nên biểu đồ cần phải có tên cho từng biến để dễ theo dõi. Trong ví dụ sau đây, biểu đồ bên trái không có tên mà chỉ dùng tên của biến gốc (tức *x* và *y*), còn bên phải có tên dễ hiểu hơn.

```
> par(mfrow=c(1,2))
> N <- 200
> x <- runif(N, -4, 4)
> y <- sin(x) + 0.5*rnorm(N)
> plot(x,y)
> plot(x, y, xlab="X factor",
             ylab="Production",
             main="Production and x factor \n Second line of title here")
> par(mfrow=c(1,1))
```

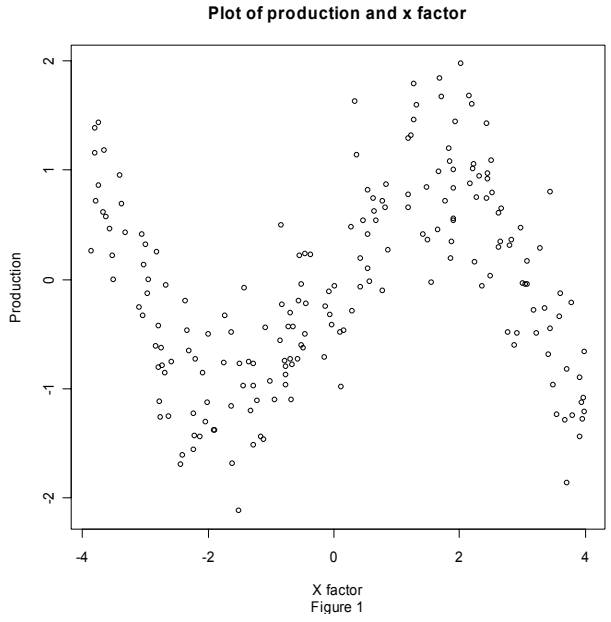
Trong các lệnh trên, `xlab` (viết tắt từ x label)và `ylab` (viết tắt từ y label) dùng để đặt tên cho trục hoành và trục tung. Còn `main` được dùng để đặt tên cho biểu đồ. Chú ý rằng trong `main` có kí hiệu `\n` dùng để viết dòng thứ hai (nếu tên gọi biểu đồ quá dài).



Biểu đồ 2. Biểu đồ bên trái không có tên gọi, biểu đồ bên phải có tên gọi cho trục tung, trục hoành và tên của biểu đồ.

Ngoài ra, chúng ta còn có thể sử dụng hàm `title` và `sub` để đặt tên:

```
> plot(x, y, xlab="Time",
             ylab="Production")
> title(main="Plot of production and x factor",
        sub="Figure 1")
```



8.1.3 Cho giới hạn của trục tung và trục hoành

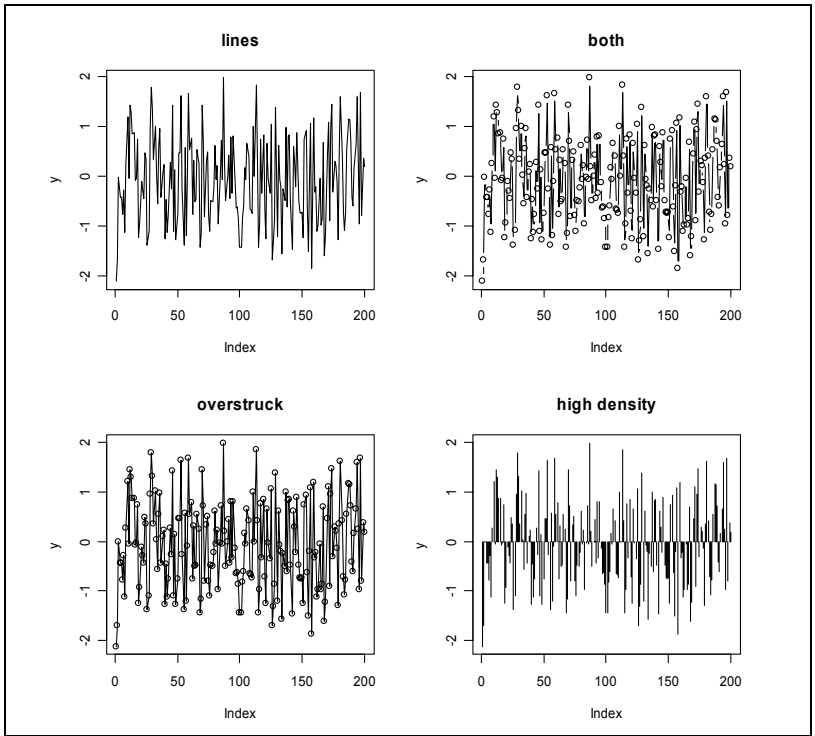
Nếu không cung cấp giới hạn của trục tung và trục hoành, R sẽ tự động tìm điều chỉnh và cho các số liệu này. Tuy nhiên, chúng ta cũng có thể kiểm soát biểu đồ bằng cách sử dụng `xlim` và `ylim` để cho R biết cụ thể giới hạn của hai trục này:

```
> plot(x, y, xlab=”X factor”,
             ylab=”Production”,
             main=”Plot of production and x factor”,
             xlim=c(-5, 5),
             ylim=c(-3, 3))
```

8.1.4 Thể loại và đường biểu diễn

Trong một dãy biểu đồ, chúng ta có thể yêu cầu R vẽ nhiều kiểu và đường biểu diễn khác nhau.

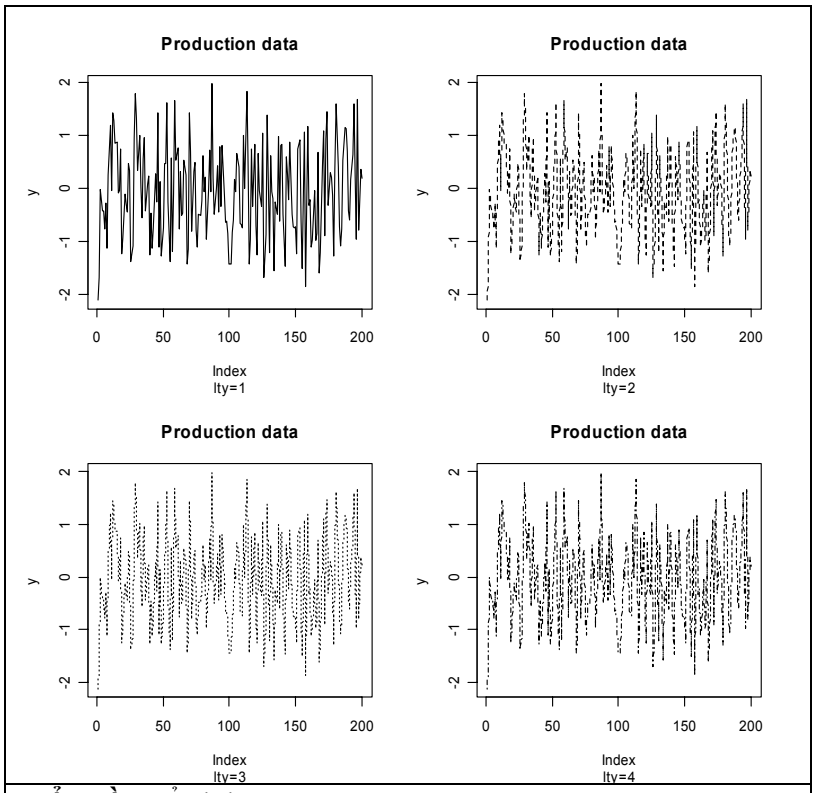
```
> par(mfrow=c(2,2))
> plot(y, type="l"); title("lines")
> plot(y, type="b"); title("both")
> plot(y, type="o"); title("overstruck")
> plot(y, type="h"); title("high density")
```



Biểu đồ 3. Kiểu biểu đồ và đường biểu diễn.

Ngoài ra, chúng ta cũng có thể nhiều đường biểu diễn bằng `lty` như sau:

```
> par(mfrow=c(2,2))
> plot(y, type="l", lty=1); title(main="Production data", sub="lty=1")
> plot(y, type="l", lty=2); title(main="Production data", sub="lty=2")
> plot(y, type="l", lty=3); title(main="Production data", sub="lty=3")
> plot(y, type="l", lty=4); title(main="Production data", sub="lty=4")
```



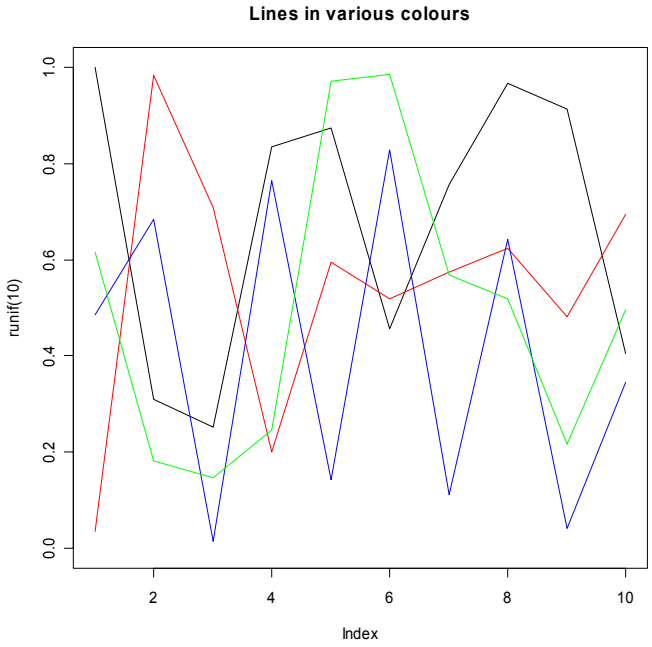
Biểu đồ 4. Ảnh hưởng của `lty`.

8.1.5 Màu sắc, khung, và kí hiệu

Chúng ta có thể kiểm soát màu sắc của một biểu đồ bằng lệnh `col`. Giá trị mặc định của `col` là 1. Tuy nhiên, chúng ta có thể thay đổi các màu theo ý muốn hặoc bằng cách cho số hoặc bằng cách viết ra tên màu như "`red`", "`blue`", "`green`", "`orange`", "`yellow`", "`cyan`", v.v...

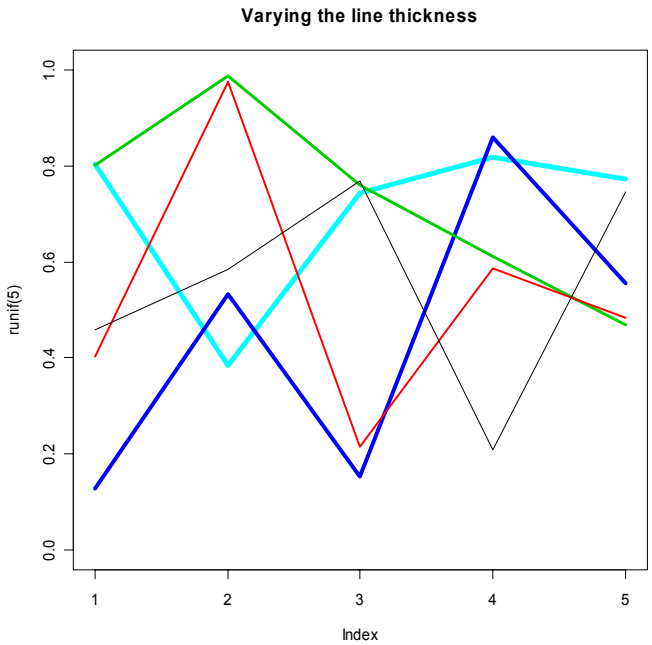
Ví dụ sau đây dùng một hàm để vẽ ba đường biểu diễn với ba màu đỏ, xanh nước biển, và xanh lá cây:

```
> plot(runif (10), ylim=c(0,1), type='l')
> for (i in c('red', 'blue', 'green'))
   {
   lines(runif (10), col=i )
  }
> title(main="Lines in various colours")
```



Ngoài ra, chúng ta còn có thể vẽ đường biểu diễn bằng cách tăng bề dày của mỗi đường:

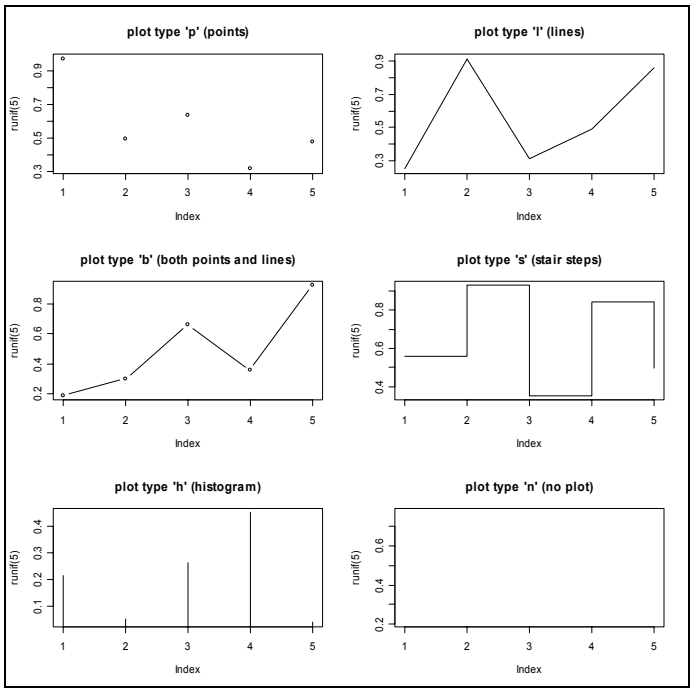
```
> plot(runif(5), ylim=c(0,1), type='n')
> for (i in 5:1)
  {
  lines( runif(5), col=i, lwd=i )
  }
> title(main="Varying the line thickness")
```



Hình dạng của biểu đồ cũng có thể thay đổi bằng `type` như sau:

```
> op <- par(mfrow=c(3,2))
```

```
> plot(runif(5), type = 'p',
       main = "plot type 'p' (points)")
> plot(runif(5), type = 'l',
       main = "plot type 'l' (lines)")
> plot(runif(5), type = 'b',
       main = "plot type 'b' (both points and lines)")
> plot(runif(5), type = 's',
       main = "plot type 's' (stair steps)")
> plot(runif(5), type = 'h',
       main = "plot type 'h' (histogram)")
> plot(runif(5), type = 'n',
       main = "plot type 'n' (no plot)")
> par(op)
```

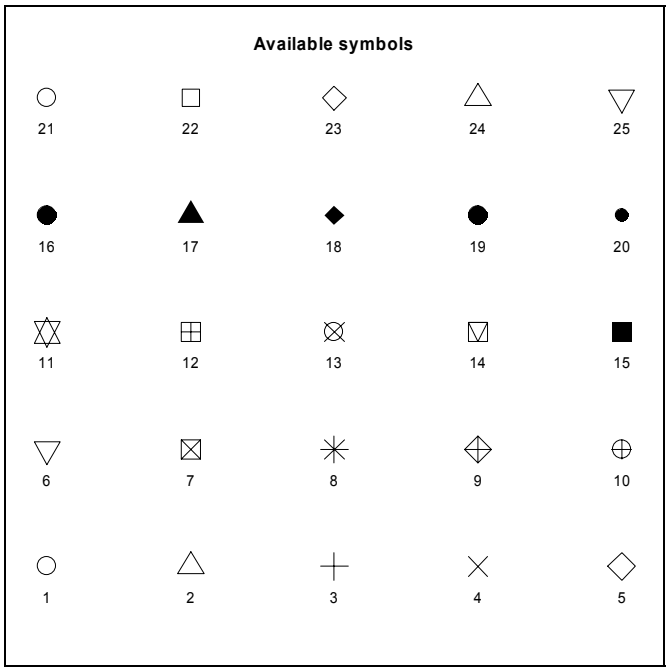


Khung biểu đồ có thể kiểm soát bằng lệnh `bty` với các thông số như sau:

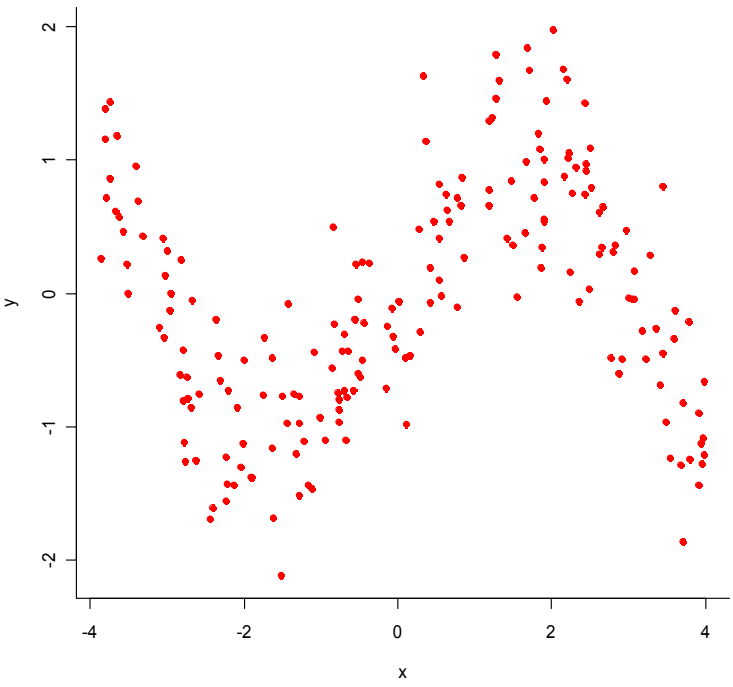
`bty="n"`	Không có vòng khung chung quanh biểu đồ
`bty="o"`	Có 4 khung chung quanh biểu đồ
`bty="c"`	Vẽ một hộp gồm 3 cạnh chung quanh biểu đồ theo hình chữ C
`bty="l"`	Vẽ hộp 2 cạnh chung quanh biểu đồ theo hình chữ L
`bty="7"`	Vẽ hộp 2 cạnh chung quanh biểu đồ theo hình số 7

Cách hay nhất để bạn đọc làm quen với các cách vẽ biểu đồ này là bằng cách thử trên R để biết rõ hơn.

Kí hiệu của một biểu đồ cũng có thể thay thế bằng cách cung cấp số cho `pch` (plotting character) trong R. Các kí hiệu thông dụng là:




```
> plot(x, y, col="red", pch=16, bty="l")
```



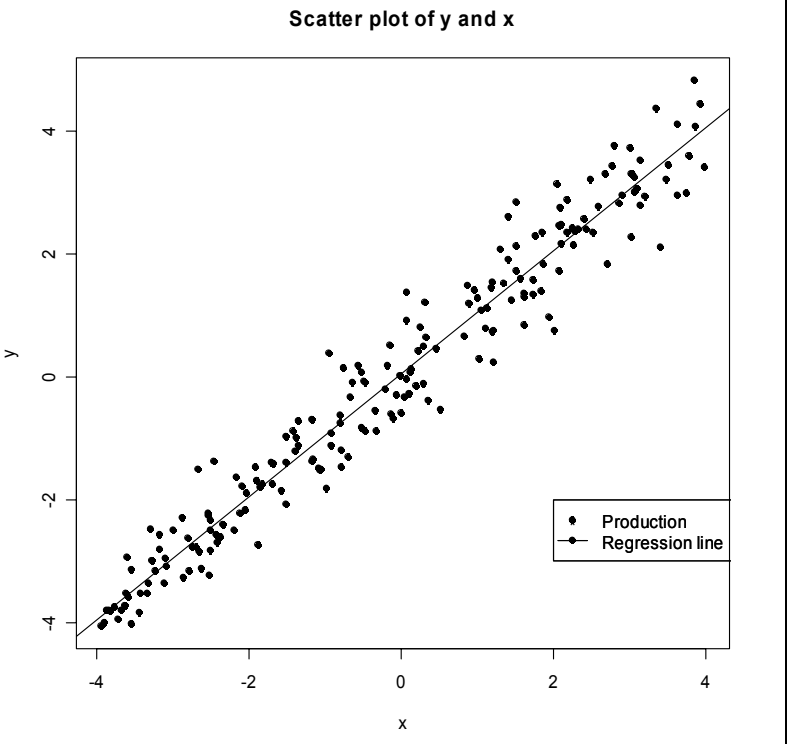
Biểu đồ 4. Ảnh hưởng của `pch=16` và `col="red", bty="l"`.

8.1.6 Ghi chú (legend)

Hàm `legend` rất có ích cho việc ghi chú một biểu đồ và giúp người đọc hiểu được ý nghĩa của biểu đồ tốt hơn. Cách sử dụng `legend` có thể minh joạ bằng ví dụ sau đây:

```
> N <- 200
> x <- runif(N, -4, 4)
> y <- x + 0.5*rnorm(N)
> plot(x,y, pch=16, main="Scatter plot of y and x")
> reg <- lm(y~x)
> abline(reg)
> legend(2,-2, c("Production","Regression line"), pch=16, lty=c(0,1))
```

Thông số `legend(2, -2)` có nghĩa là đặt phần ghi chú vào trục hoành (x-axis) bằng 2 và trục tung (y-axis) bằng -2.



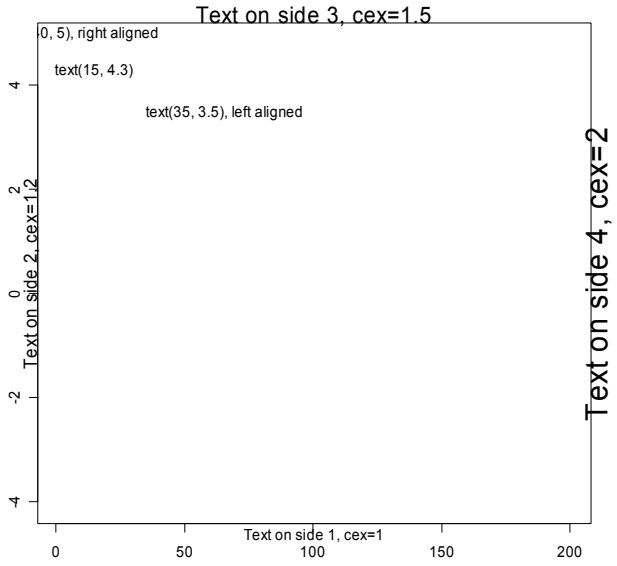
Biểu đồ 5. Ảnh hưởng của `legend`

8.1.7 Viết chữ trong biểu đồ

Phần lớn các biểu đồ không cung cấp phương tiện để viết chữ hay ghi chú trong biểu đồ, hay có cung cấp nhưng rất hạn chế. Trong R có hàn `mtext()` cho phép chúng ta đặt chữ viết hay giải thích bên cạnh hay trong biểu đồ.

Bắt đầu từ phía dưới của biểu đồ (`side=1`), chúng ta chuyển theo hướng kim đồng hố đến cạnh số 4. Lệnh `plot` trong ví dụ sau đây không in tên của trục và tên của biểu đồ, nhưng chỉ cung cấp một cái khung. Trong ví dụ này, chúng ta sử dụng `cex` (character expansion) để kiểm soát kích thước của chữ viết. Theo mặc định thì `cex=1`, nhưng với `cex=2`, chữ viết sẽ có kích thước gấp hai lần kích thước mặc định. Lệnh `text()` cho phép chúng ta đặt chữ viết vào một vị trí cụ thể. Lệnh thứ nhất đặt chữ viết trong ngoặc kép và trung tâm tại `x=15, y=4.3`. Qua sử dụng `adj`, chúng ta còn có thể sắp xếp về phía trái (`adj=0`) sao cho tọa độ là điểm xuất phát của chữ viết.

```
> plot(y, xlab=" ", ylab=" ", type="n")
> mtext("Text on side 1, cex=1", side=1,cex=1)
> mtext("Text on side 2, cex=1.2", side=2,cex=1.2)
> mtext("Text on side 3, cex=1.5", side=3,cex=1.5)
> mtext("Text on side 4, cex=2", side=4,cex=2)
> text(15, 4.3, "text(15, 4.3)")
> text(35, 3.5, adj=0, "text(35, 3.5), left aligned")
> text(40, 5, adj=1, "text(40, 5), right aligned")
```



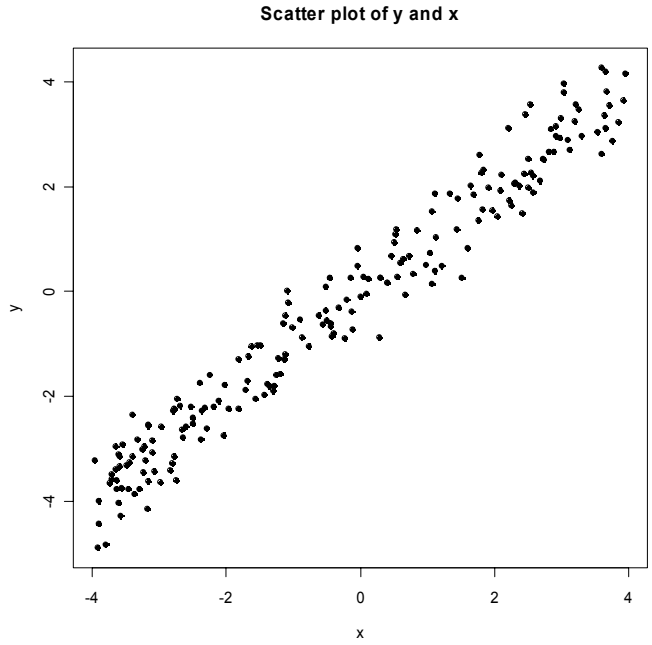
8.1.8 Đặt kí hiệu vào biểu đồ. `abline()` có thể sử dụng để vẽ một đường thẳng, với những thông số như sau:

`abline(a,b)`: đường hồi qui tuyến tính a=intercept và b=slope.
`abline(h=30)` vẽ một đường ngang tại y=30.
`abline(v=12)` vẽ một đường thẳng đứng tại điểm x=12.

Ngoài ra, chúng ta còn có thể cho vào biểu đồ một mũi tên để ghi chú một điểm số liệu nào đó.

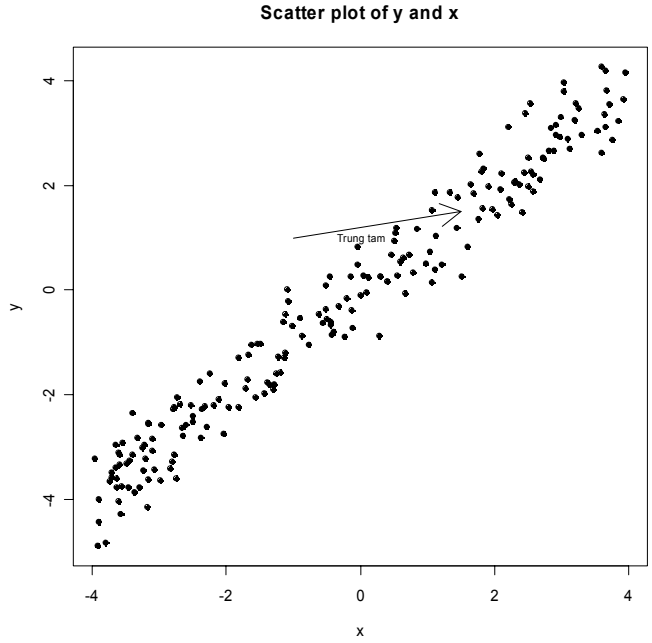
```
> N <- 200
```

```
> x <- runif(N, -4, 4)
> y <- x + 0.5*rnorm(N)
> plot(x,y, pch=16, main="Scatter plot of y and x")
```



Giả sử chúng ta muốn ghi chú ngay tại x=0 và y=0 là điểm trung tâm, chúng ta trước hết dùng arrows để vẽ mũi tên. Trong lệnh sau đây, `arrows(-1, 1, 1.5, 1.5)` có nghĩa như sau tọa độ x=-1, y=1 bắt đầu vẽ mũi tên và chấm dứt tại tọa độ x=1.5, y=1.5. Phần `text(0, 1)` yêu cầu R viết chữ tại tọa độ x=0, y=1.

```
> arrows(-1, 1.0, 1.5, 1.5)
> text(0, 1, "Trung tam", cex=0.7)
```



8.2 Số liệu cho phân tích biểu đồ

Sau khi đã biết qua môi trường và những lựa chọn để thiết kế một biểu đồ, bây giờ chúng ta có thể sử dụng một số hàm thông dụng để vẽ các biểu đồ cho số liệu. Theo tôi, biểu đồ có thể chia thành 2 loại chính: biểu đồ dùng để mô tả một biến số và biểu đồ về mối liên hệ giữa hai hay nhiều biến số. Tất nhiên, biến số có thể là liên tục hay không liên tục, cho nên, trong thực tế, chúng ta có 4 loại biểu đồ. Trong phần sau đây, tôi sẽ điểm qua các loại biểu đồ, từ đơn giản đến phức tạp.

Có lẽ cách tốt nhất để tìm hiểu cách vẽ đồ thị bằng R là bằng một dữ liệu thực tế. Tôi sẽ quay lại **ví dụ 2** trong chương trước. Trong ví dụ đó, chúng ta có dữ liệu gồm 8 cột (hay biến số): `id, sex, age, bmi, hdl, ldl, tc,` và `tg.` (Chú ý, `id` là mã số của 50 đối tượng nghiên cứu; `sex` là giới tính (nam hay nữ); `age` là độ tuổi; `bmi` là tỉ số trọng lương; `hdl` là high density cholesterol; `ldl` là low density cholesterol; `tc` là tổng số - total – cholesterol; và `tg` triglycerides). Dữ liệu được chứa trong directory directory `c:\works\insulin` dưới tên `chol.txt`. Trước khi vẽ đồ thị, chúng ta bắt đầu bằng cách nhập dữ liệu này vào R.

```
> setwd("c:/works/stats")
> cong <- read.table("chol.txt", header=TRUE, na.strings=".")
> attach(cong)
```

Hay để tiện việc theo dõi tôi sẽ nhập các dữ liệu đó bằng các lệnh sau đây:

```
sex <- c("Nam", "Nu", "Nu","Nam","Nam", "Nu","Nam","Nam","Nam", "Nu",
         "Nu","Nam", "Nu","Nam","Nam", "Nu", "Nu", "Nu", "Nu", "Nu",
         "Nu", "Nu", "Nu", "Nu","Nam","Nam", "Nu","Nam", "Nu", "Nu",
         "Nu","Nam","Nam", "Nu", "Nu","Nam", "Nu","Nam", "Nu", "Nu",
```

```
         "Nam", "Nu","Nam","Nam","Nam", "Nu","Nam","Nam", "Nu", "Nu")

age <- c(57, 64, 60, 65, 47, 65, 76, 61, 59, 57,
         63, 51, 60, 42, 64, 49, 44, 45, 80, 48,
         61, 45, 70, 51, 63, 54, 57, 70, 47, 60,
         60, 50, 60, 55, 74, 48, 46, 49, 69, 72,
         51, 58, 60, 45, 63, 52, 64, 45, 64, 62)

bmi <- c( 17, 18, 18, 18, 18, 18, 19, 19, 19, 19, 20, 20, 20, 20, 20,
          20, 21, 21, 21, 21, 21, 21, 21, 21, 22, 22, 22, 22, 22, 22,
          22, 22, 22, 22, 23, 23, 23, 23, 23, 23, 23, 23, 24, 24, 24,
          24, 24, 24, 25, 25)

hdl <- c(5.000,4.380,3.360,5.920,6.250,4.150,0.737,7.170,6.942,5.000,
         4.217,4.823,3.750,1.904,6.900,0.633,5.530,6.625,5.960,3.800,
         5.375,3.360,5.000,2.608,4.130,5.000,6.235,3.600,5.625,5.360,
         6.580,7.545,6.440,6.170,5.270,3.220,5.400,6.300,9.110,7.750,
         6.200,7.050,6.300,5.450,5.000,3.360,7.170,7.880,7.360,7.750)

ldl <- c(2.0,  3.0,  3.0,  4.0,  2.1,  3.0,  3.0,  3.0,  3.0, 2.0,
         5.0,  1.3,  1.2,  0.7,  4.0,  4.1,  4.3,  4.0,  4.3, 4.0,
         3.1,  3.0,  1.7,  2.0,  2.1,  4.0,  4.1,  4.0,  4.2, 4.2,
         4.4,  4.3,  2.3,  6.0,  3.0,  3.0,  2.6,  4.4,  4.3, 4.0,
         3.0,  4.1,  4.4,  2.8,  3.0,  2.0,  1.0,  4.0,  4.6,  4.0)

tc <-c (4.0, 3.5, 4.7, 7.7, 5.0, 4.2, 5.9, 6.1, 5.9, 4.0,
        6.2, 4.1, 3.0, 4.0, 6.9, 5.7, 5.7, 5.3, 7.1, 3.8,
        4.3, 4.8, 4.0, 3.0, 3.1, 5.3, 5.3, 5.4, 4.5, 5.9,
        5.6, 8.3, 5.8, 7.6, 5.8, 3.1, 5.4, 6.3, 8.2, 6.2,
        6.2, 6.7, 6.3, 6.0, 4.0, 3.7, 6.1, 6.7, 8.1, 6.2)

tg <- c(1.1, 2.1, 0.8, 1.1, 2.1, 1.5, 2.6, 1.5, 5.4, 1.9,
        1.7, 1.0, 1.6, 1.1, 1.5, 1.0, 2.7, 3.9, 3.0, 3.1,
        2.2, 2.7, 1.1, 0.7, 1.0, 1.7, 2.9, 2.5, 6.2, 1.3,
        3.3, 3.0, 1.0, 1.4, 2.5, 0.7, 2.4, 2.4, 1.4, 2.7,
        2.4, 3.3, 2.0, 2.6, 1.8, 1.2, 1.9, 3.3, 4.0, 2.5)

cong <- data.frame(sex, age, bmi, hdl, ldl, tc, tg)
```

Sau khi đã có số liệu, chúng ta sẵn sàng tiến hành phân tích số liệu bằng biểu đồ như sau:

8.3 Biểu đồ cho một biến số rời rạc (discrete variable): `barplot`

Biến `sex` trong dữ liệu trên có hai giá trị (`nam` và `nu`), tức là một biến không liên tục. Chúng ta muốn biết tần số của giới tính (bao nhiêu nam và bao nhiêu nữ) và vẽ một biểu đồ đơn giản. Để thực hiện ý định này, trước hết, chúng ta cần dùng hàm table để biết tần số:

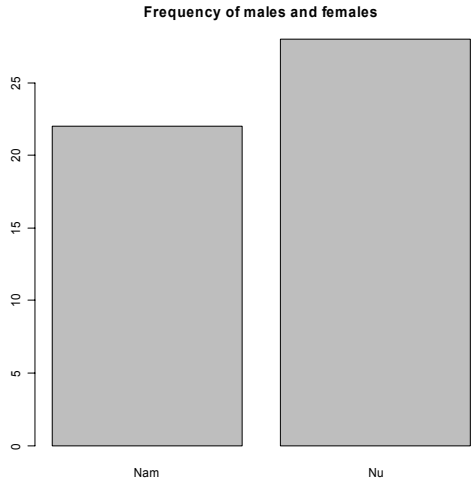
```
> sex.freq <- table(sex)
> sex.freq
sex
Nam  Nu
 22  28
```

Có 22 nam và 28 nữa trong nghiên cứu. Sau đó dùng hàm `barplot` để thể hiện tần số này như sau:

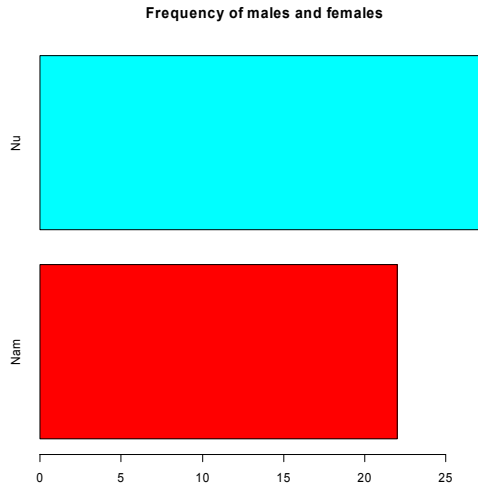
```
> barplot(sex.freq, main="Frequency of males and females")
```

Biểu trên cũng có thể có được bằng một lệnh đơn giản hơn (**Biểu đồ 8a**):

```
> barplot(table(sex), main="Frequency of males and females")
```



Biểu đồ 8a. Tần số giới tính thể hiện bằng cột số.



Biểu đồ 8b. Tần số giới tính thể hiện bằng dòng số.

Thay vì thể hiện tần số nam và nữ bằng 2 cột, chúng ta có thể thể hiện bằng hai dòng bằng thông số `horiz = TRUE`, như sau (xem kết quả trong **Biểu đồ 6b**):

```
> barplot(sex.freq,
        horiz = TRUE,
        col = rainbow(length(sex.freq)),
        main="Frequency of males and females")
```

8.4 Biểu đồ cho hai biến số rời rạc (discrete variable): `barplot`

Age là một biến số liên tục. Chúng ta có thể chia bệnh nhân thành nhiều nhóm dựa vào độ tuổi. Hàm `cut` có chức năng "cắt" một biến liên tục thành nhiều nhóm rời rạc. Chẳng hạn như:

```
> ageg <- cut(age, 3)
> table(ageg)
ageg
  (42,54.7] (54.7,67.3]   (67.3,80]
         19          24           7
```

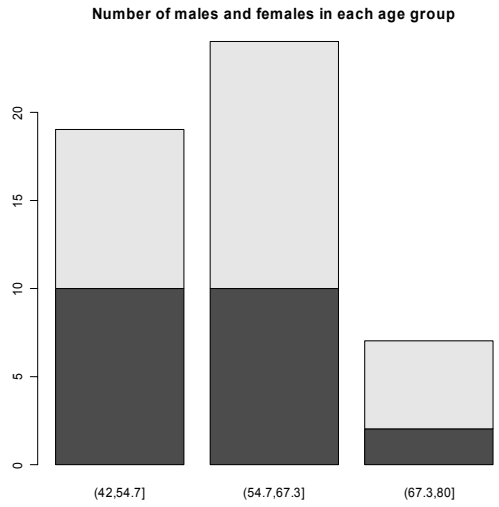
Có hiệu quả chia biến `age` thành 3 nhóm. Tần số của ba nhóm này là: 42 tuổi đến 54.7 tuổi thành nhóm 1, 54.7 đến 67.3 thành nhóm 2, và 67.3 đến 80 tuổi thành nhóm 3. Nhóm 1 có 19 bệnh nhân, nhóm 2 và 3 có 24 và 7 bệnh nhân.

Bây giờ chúng ta muốn biết có bao nhiêu bệnh nhân trong từng độ tuổi và từng giới tính bằng lệnh `table`:

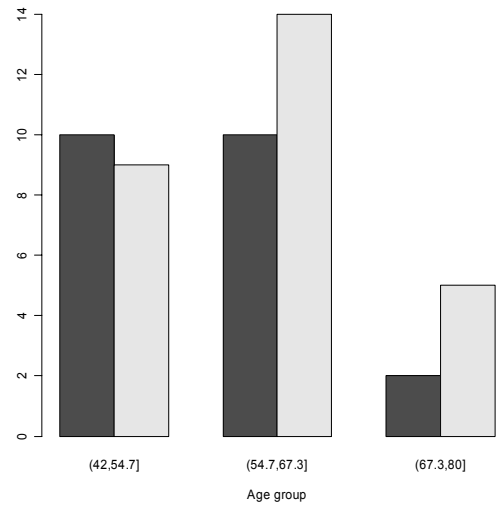
```
> age.sex <- table(sex, ageg)
> age.sex
     ageg
sex   (42,54.7] (54.7,67.3] (67.3,80]
  Nam        10          10         2
  Nu          9          14         5
```

Kết quả trên cho thấy chúng ta có 10 bệnh nhân nam và 9 nữ trong nhóm tuổi thứ nhất, 10 nam và 14 nữa trong nhóm tuổi thứ hai, v.v… Để thể hiện tần số của hai biến này, chúng ta vẫn dùng `barplot:`

```
> barplot(age.sex, main="Number of males and females in each age
   group")
```



Biểu đồ 7a. Tần số giới tính và nhóm tuổi thể hiện bằng cột số.



Biểu đồ 7b. Tần số giới tính và nhóm tuổi thể hiện bằng hai dòng số.

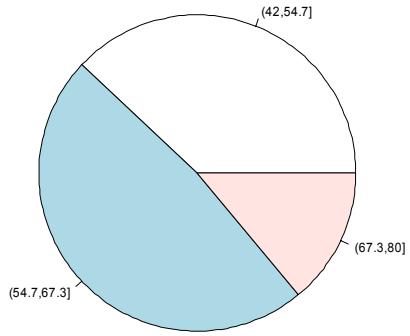
Trong **Biểu đồ 7a**, mỗi cột là cho một độ tuổi, và phần đậm của cột là nữ, và phần màu nhạt là tần số của nam giới. Thay vì thể hiện tần số nam nữ trong một cột, chúng ta cũng có thể thể hiện bằng 2 cột với `beside=T` như sau (**Biểu đồ 7b**):

```
barplot(age.sex, beside=TRUE, xlab="Age group")
```

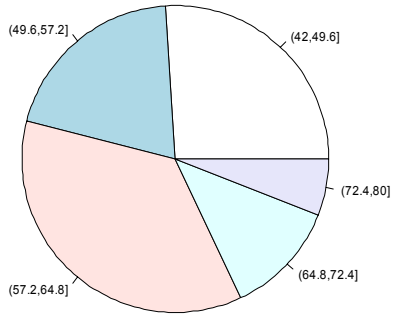
8.5 Biểu đồ hình tròn

Tần số một biến rời rạc cũng có thể thể hiện bằng biểu đồ hình tròn. Ví dụ sau đây vẽ biểu đồ tần số của độ tuổi. **Biểu đồ 8a** là 3 nhóm độ tuổi, và **Biểu đồ 8b** là biểu đồ tần số cho 5 nhóm tuổi:

```
> pie(table(ageg))
pie(table(cut(age,5)))
```



Biểu đồ 8a. Tần số cho 3 nhóm tuổi



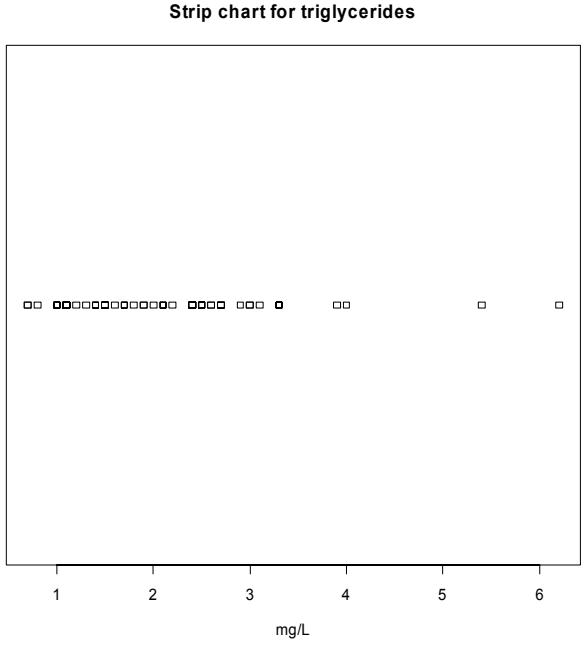
Biểu đồ 8b. Tần số cho 5 nhóm tuổi

8.6 Biểu đồ cho một biến số liên tục: `stripchart` và `hist`

8.6.1 Stripchart

Biểu đồ strip cho chúng ta thấy tính liên tục của một biến số. Chẳng hạn như chúng ta muốn tìm hiểu tính liên tục của triglyceride (`tg`), hàm `stripchart()` sẽ giúp trong mục tiêu này:

```
> stripchart(tg,
             main="Strip chart for triglycerides", xlab="mg/L")
```

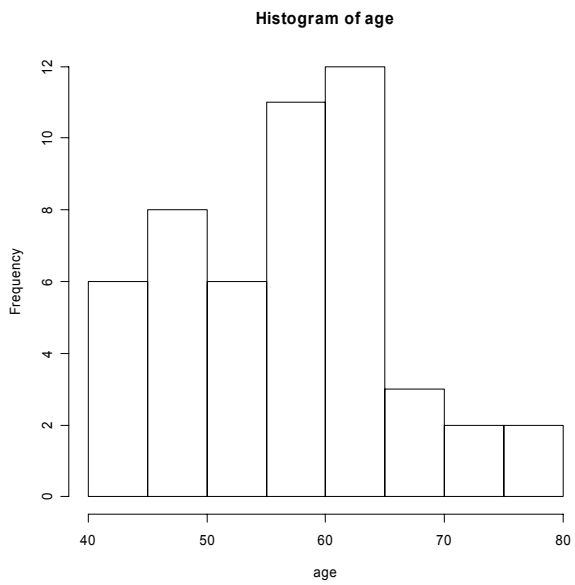


Chúng ta thấy biến số `tg` có sự bất liên tục, nhất là các đối tượng có `tg` cao. Trong khi phần lớn đối tượng có độ `tg` thấp hơn 5, thì có 2 đối tượng với `tg` rất cao (>5).

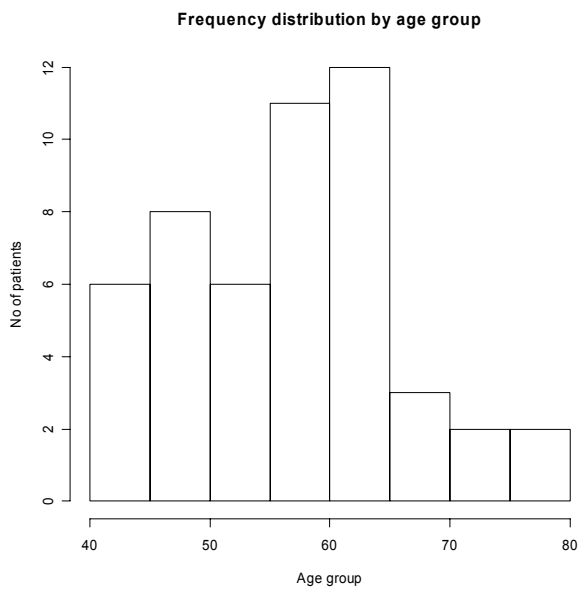
8.6.2 Histogram

Age là một biến số liên tục. Để vẽ biểu đồ tần số của biến số `age`, chúng ta chỉ đơn giản lệnh `hist(age)`. Như đã đề cập trên, chúng ta có thể cải tiến đồ thị này bằng cách cho thêm tựa đề chính (`main`) và tựa đề của trục hoành (`xlab`) và trục tung (`ylab`):

```
> hist(age)
> hist(age, main="Frequency distribution by age group", xlab="Age
group", ylab="No of patients")
```



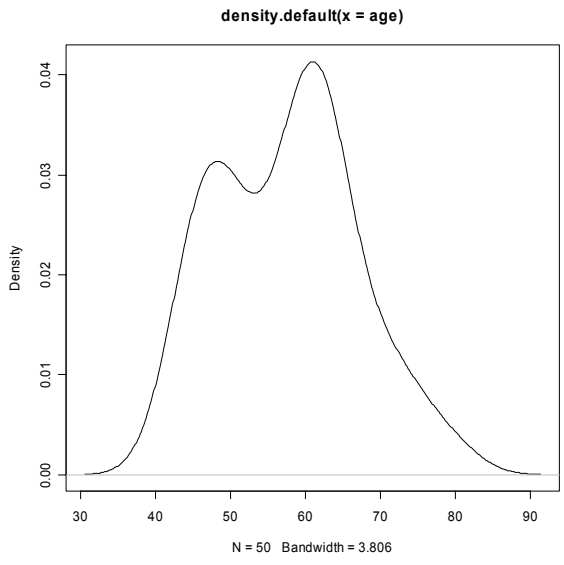
Biểu đồ 9a. Trục tung là số bệnh nhân (đối tượng nghiên cứu) và trục hoành là độ tuổi. Chẳng hạn như tuổi 40 đến 45 có 6 bệnh nhân, từ 70 đến 80 tuổi có 4 bệnh nhân.



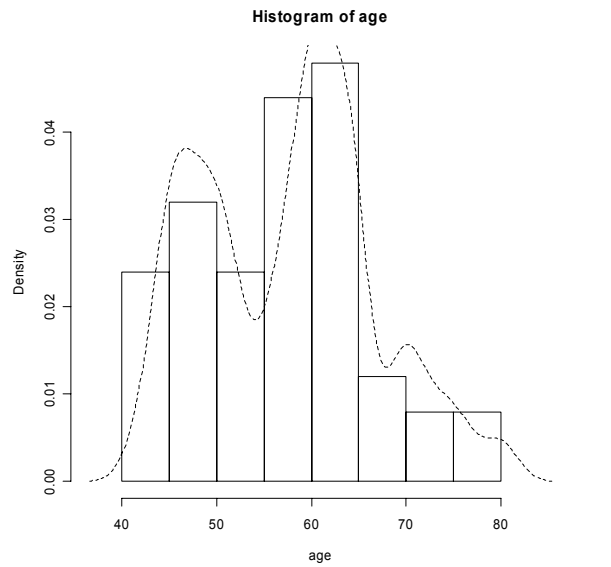
Biểu đồ 9b. Thêm tên biểu đồ và tên của trục trung và trục hoành bằng `xlab` và `ylab`.

Chúng ta cũng có thể biến đổi biểu đồ thành một đồ thị phân phối xác suất bằng hàm `plot(density)` như sau (kết quả trong **Biểu đồ 10a**):

```
> plot(density(age),add=TRUE)
```



Biểu đồ 10a. Xác suất phân phối mật độ cho biến `age` (độ tuổi).



Biểu đồ 10b. Xác suất phân phối mật độ cho biến `age` (độ tuổi) với nhiều interquartile.

Chúng ta có thể vẽ hai đồ thị chồng lên bằng cách dùng hàm interquartile như sau (kết quả xem **Biểu đồ 10b**):

```
> iqr <- diff(summary(age)[c(2,5)])
> des <- density(age, width=0.5*iqr)
> hist(age, xlim=range(des$x), probability=TRUE)
> lines(des, lty=2)
```

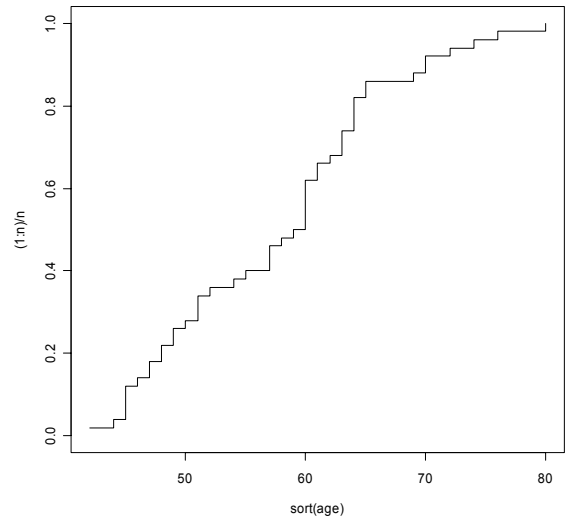
Trong đồ thị trên, chúng ta dùng khoảng cách `0.5*iqr` (tương đối "gần" nhau). Nhưng chúng ta có thể biến đổi thông số này thành `1.5*iqr` để làm cho phân phối thực tế hơn:

```
> iqr <- diff(summary(age)[c(2,5)])
> des <- density(age, width=1.5*iqr)
> hist(age, xlim=range(des$x), probability=TRUE)
> lines(des, lty=2)
```

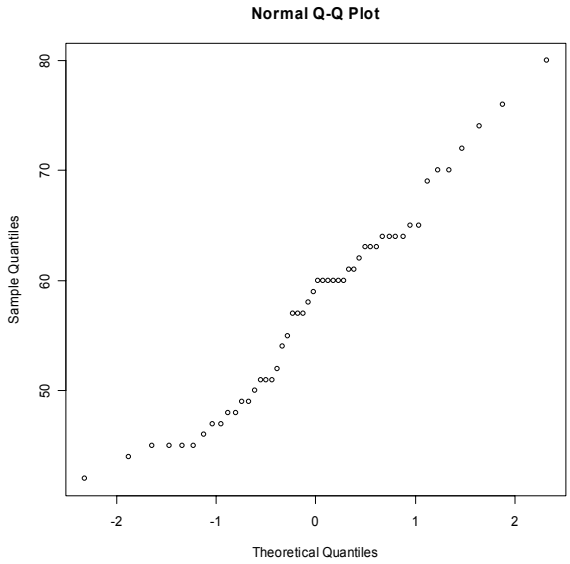
Chúng ta có thể biến đổi biểu đồ thành một đồ thị phân phối xác suất tích lũy (cumulative distribution) bằng hàm `plot` và `sort` như sau:

```
> n <- length(age)
> plot(sort(age), (1:n)/n, type="s", ylim=c(0,1))
```

Kết quả được trình bày trong phần trái của biểu đồ sau đây (**Biểu đồ 11**).



Biểu đồ 11. Xác suất phân phối mật độ cho biến `age` (độ tuổi).



Biểu đồ 12. Kiểm tra biến `age` có theo luật phân phối chuẩn hay không.

Trong đồ thị trên, trục tung là xác suất tích lũy và trục hoành là độ tuổi từ thấp đến cao. Chẳng hạn như nhìn qua biểu đồ, chúng ta có thể thấy khoảng 50% đối tượng có tuổi thấp hơn 60.

Để biết xem phân phối của `age` có theo luật phân phối chuẩn (normal distribution) hay không chúng ta có thể sử dụng hàm `qqnorm`.

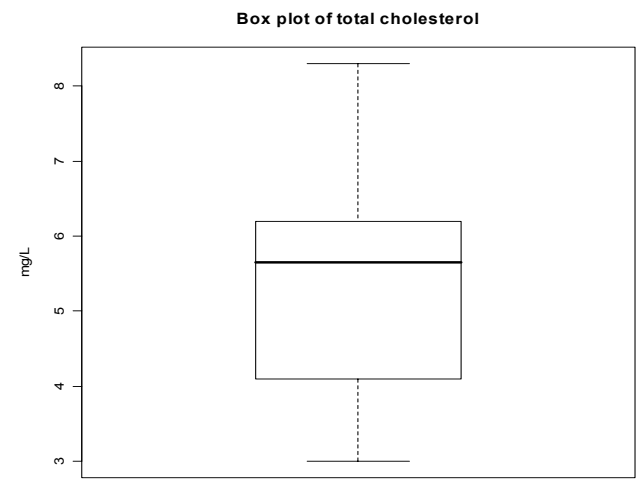
```
> qqnorm(age)
```

Trục hoành của biểu đồ trên là định lượng theo luật phân phối chuẩn (theoretical quantile) và trục hoành định lượng của số liệu (sample quantiles). Nếu phân phối của `age` theo luật phân phối chuẩn, thì đường biểu diễn phải theo một đường thẳng chéo 45 độ (tức là định lượng phân phối và định lượng số liệu bằng nhau). Nhưng qua **Biểu đồ 12**, chúng ta thấy phân phối của `age` không hẳn theo luật phân phối chuẩn.

8.6.3 Biểu đồ hộp (`boxplot`)

Để vẽ biểu đồ hộp của biến số `tc`, chúng ta chỉ đơn giản lệnh:

```
> boxplot(tc, main="Box plot of total cholesterol", ylab="mg/L")
```



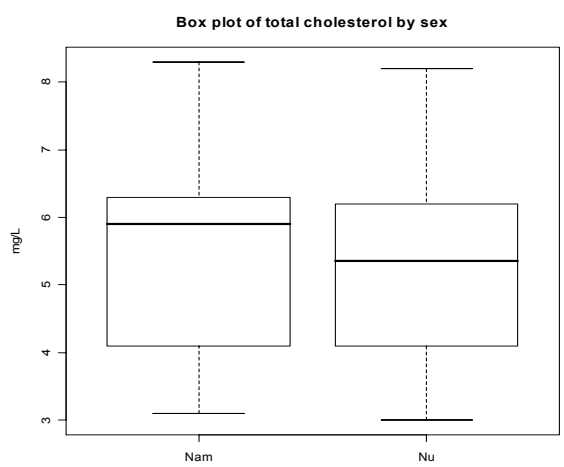
Biểu đồ 13. Trong biểu đồ này, chúng ta thấy median (trung vị) khoảng 5.6 mg/L, 25% total cholesterol thấp hơn 4.1, và 75% thấp hơn 6.2. Total cholesterol thấp nhất là khoang 3, và cao nhất là trên 8 mg/L.

Trong biểu đồ sau đây, chúng ta so sánh `tc` giữa hai nhóm nam và nữ:

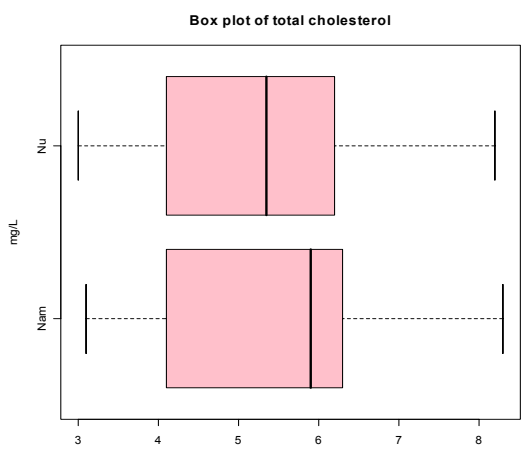
```
> boxplot(tc ~ sex, main="Box plot of total cholestrol by sex",
ylab="mg/L")
```

Kết quả trình bày trong **Biểu đồ 14a**. Chúng ta có thể biến đổ giao diện của đồ thị bằng cách dùng thông số `horizontal=TRUE` và thay đổi màu bằng thông số `col` như sau (**Biểu đồ 14b)**:

```
> boxplot(tc~sex, horizontal=TRUE, main="Box plot of total
cholesterol", ylab="mg/L", col = "pink")
```



Biểu đồ 14a. Trong biểu đồ này, chúng ta thấy trung vị của total cholesterol ở nữ giới thấp hơn nam giới, nhưng độ dao động giữa hai nhóm không khác nhau bao nhiêu.

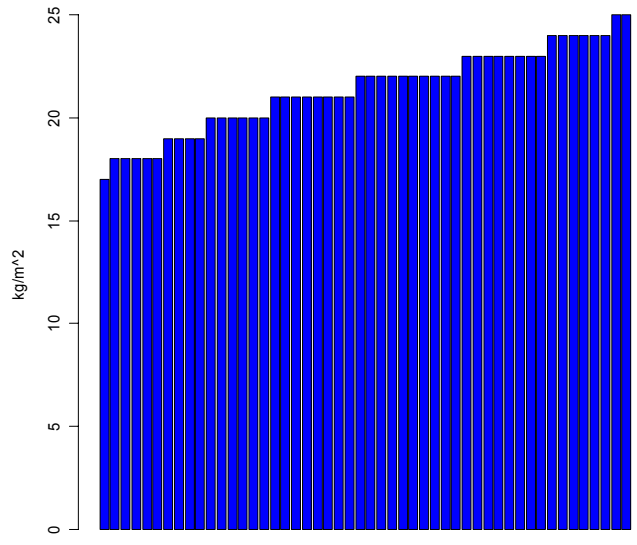


Biểu đồ 14b. Total cholesterol cho từng giới tính, với màu sắc và hình hộp nằm ngang.

8.6.4 Biểu đồ thanh (bar chart)

Để vẽ biểu đồ thanh của biến số `bmi`, chúng ta chỉ đơn giản lệnh:

```
> barplot(bmi, col="blue")
```

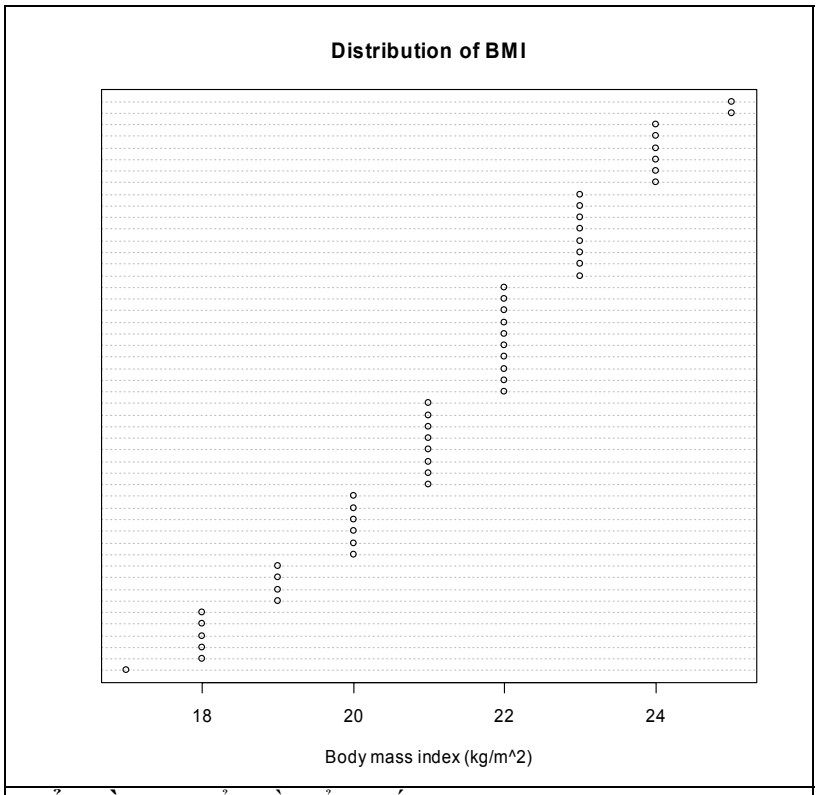


Biểu đồ 15. Biểu đồ thanh cho biến `bmi`.

8.6.5 Biểu đồ điểm (dotchart)

Một đồ thị khác cung cấp thông tin giống như `barplot` là `dotchart`:

```
> dotchart(bmi, xlab="Body mass index (kg/m^2)", main="Distribution of BMI")
```



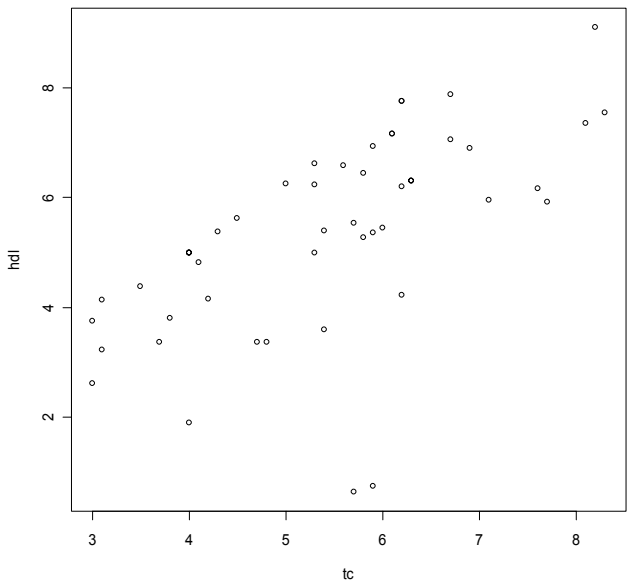
Biểu đồ 16. Biểu đồ điểm biến `bmi`.

8.7 Phân tích biểu đồ cho hai biến liên tục

8.7.1 Biểu đồ tán xạ (scatter plot)

Để tìm hiểu mối liên hệ giữa hai biến, chúng ta dùng biểu đồ tán xạ. Để vẽ biểu đồ tán xạ về mối liên hệ giữa biến số `tc` và `hdl`, chúng ta sử dụng hàm `plot`. Thông số thứ nhất của hàm plot là trục hoành (x-axis) và thông số thứ 2 là trục tung. Để tìm hiểu mối liên hệ giữa `tc` và `hdl` chúng ta đơn giản lệnh:

```
> plot(tc, hdl)
```



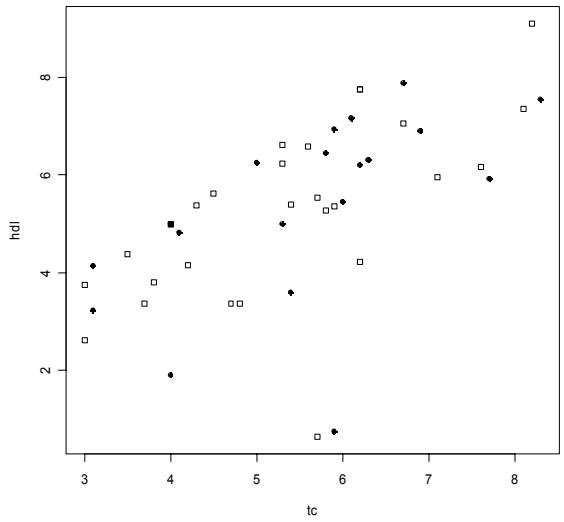
Biểu đồ 17. Mối liên hệ giữa `tc` và `hdl`. Trong biểu đồ này, chúng ta vẽ biến số `hdl` trên trục tung và `tc` trên trục hoành.

Chúng ta muốn phân biệt giới tính (nam và nữ) trong biểu đồ trên. Để vẽ biểu đồ đó, chúng ta phải dùng đến hàm `ifelse`. Trong lệnh sau đây, nếu `sex=="Nam"` thì vẽ kí tự số 16 (ô tròn), nếu không nam thì vẽ kí tự số 22 (tức ô vuông):

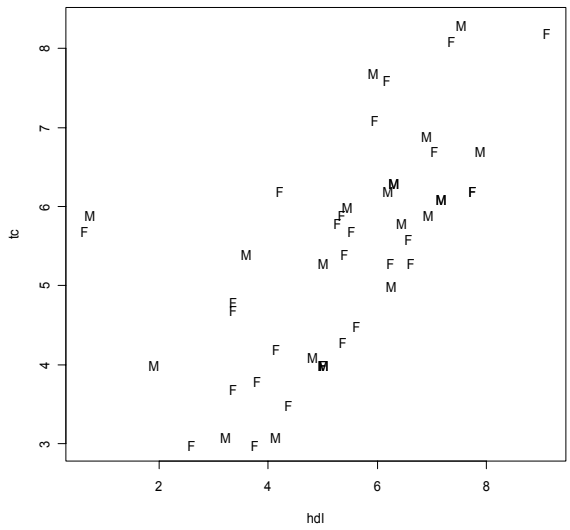
```
> plot(hdl, tc, pch=ifelse(sex=="Nam", 16, 22))
```

Kết quả là **Biểu đồ 18a**. Chúng ta cũng có thể thay kí tư thành "M" (nam) và "F" nữ(xem **Biểu đồ 18b**):

```
> plot(hdl, tc, pch=ifelse(sex=="Nam", "M", "F"))
```



Biểu đồ 18a. Mối liên hệ giữa tc và hdl theo từng giới tính được thể hiện bằng hai kí hiệu dấu.



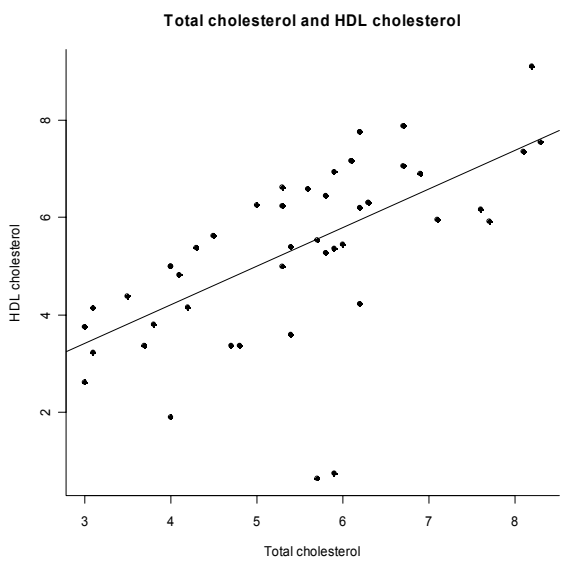
Biểu đồ 18a. Mối liên hệ giữa tc và hdl theo từng giới tính được thể hiện bằng hai kí tự.

Chúng ta cũng có thể vẽ một đường biểu diễn hồi qui tuyến tính (regression line) qua các điểm trên bằng cách tiếp tục ra các lệnh sau đây:

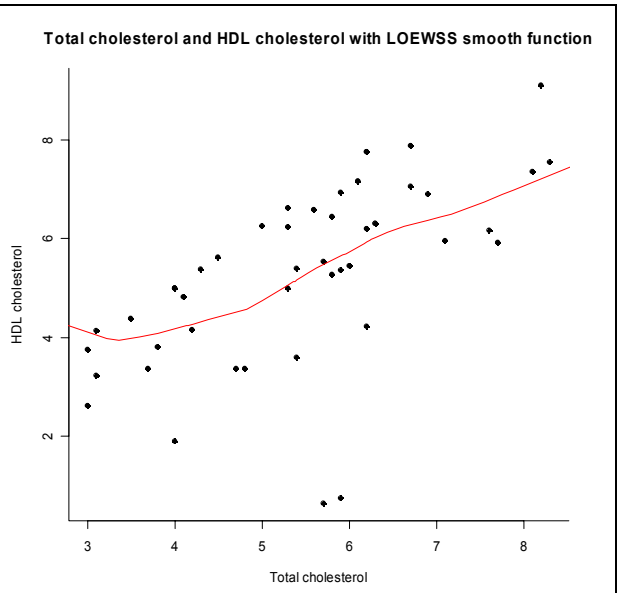
```
> plot(hdl ~ tc, pch=16, main="Total cholesterol and HDL cholesterol",
xlab="Total cholesterol", ylab="HDL cholesterol", bty="l")
> reg <- lm(hdl ~ tc)
> abline(reg)
```

Kết quả là **Biểu đồ 19a** dưới đây. Chúng ta cũng có thể dùng hàm trơn (smooth function) để biểu diễn mối liên hệ giữa hai biến số. Đồ thị sau đây sử dụng `lowess` (một hàm thông thường nhất) trong việc "làm trơn" số liệu `tc` và `hdl` (**Biểu đồ 19b).**

```
> plot(hdl ~ tc, pch=16,
       main="Total cholesterol and HDL cholesterol with LOEWSS smooth
function",
       xlab="Total cholesterol", ylab="HDL cholesterol", bty="l")

> lines(lowess(hdl, tc, f=2/3, iter=3), col="red")
```



Biểu đồ 19a. Trong lệnh trên, `reg<-lm(hdl~tc)` có nghĩa là tìm phương trình liên hệ giữa `hdl` và `tc` bằng linear model (`lm`) và đ8ạt kết quả vào đối tượng `reg`. Lệnh thứ hai `abline(reg)` yêu cầu R vẽ đường thẳng từ phương trình trong `reg`



Biểu đồ 19b. Thay vì dùng abline, chúng ta dùng hàm lowess để thể hiện mối liên hệ giữa `tc` và `hdl`.

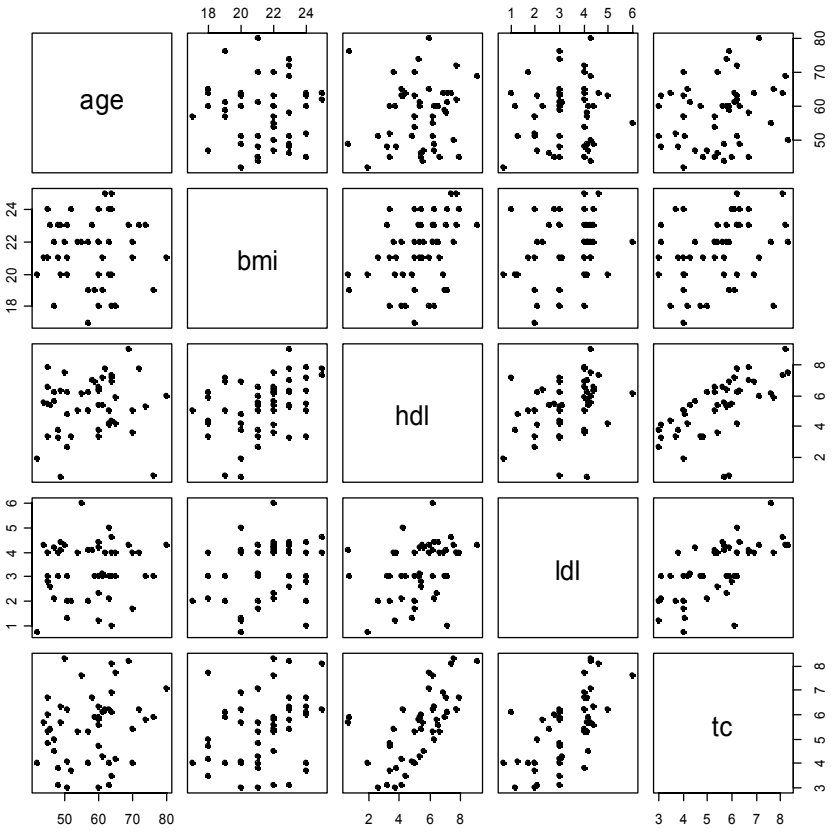
Bạn đọc có thể thí nghiệm với nhiều thông số `f=1/2, f=2/5`, hay thậm chí `f=1/10` sẽ thấy đồ thị biến đổi một cách "thú vị".

8.8 Phân tích Biểu đồ cho nhiều biến: `pairs`

Chúng ta có thể tìm hiểu mối liên hệ giữa các biến số như `age, bmi, hdl, ldl` và `tc` bằng cách dùng lệnh `pairs`. Nhưng trước hết, chúng ta phải đưa các biến số này vào một `data.frame` chỉ gồm những biến số có thể vẽ được, và sau đó sử dụng hàm `pairs` trong R.

```
> lipid <- data.frame(age,bmi,hdl,ldl,tc)
> pairs(lipid, pch=16)
```

Kết quả sẽ là:

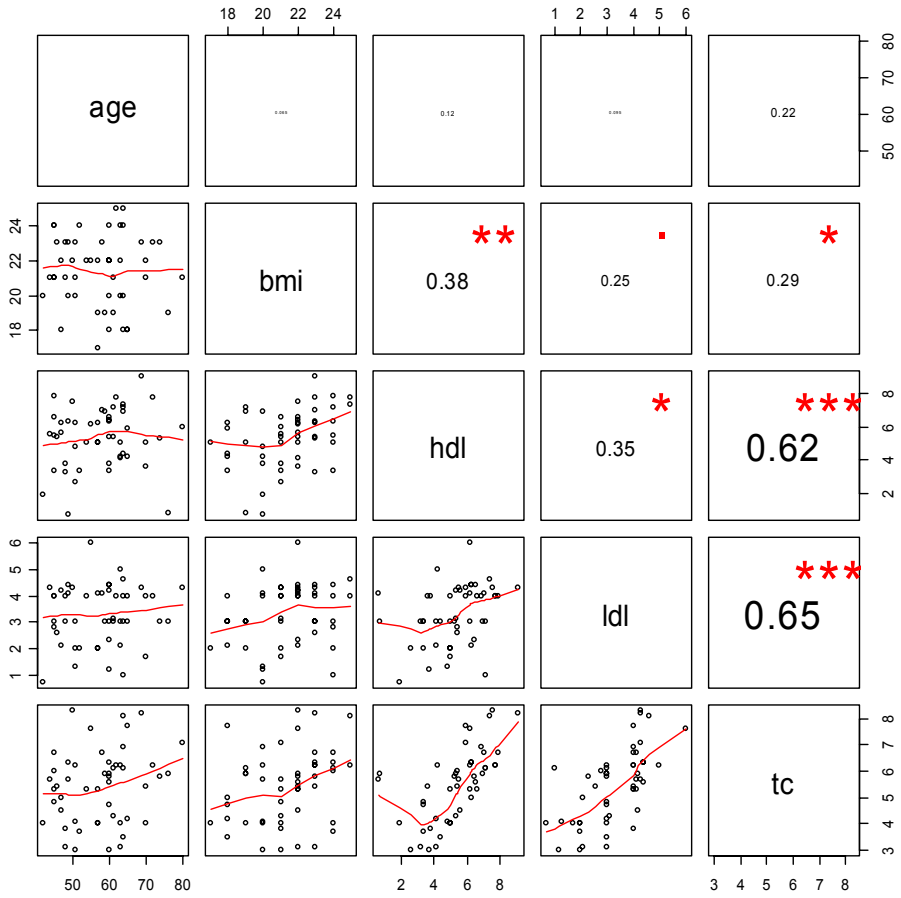


Biểu đồ trên đây có thể cải tiến bằng hàm `matrix.cor` (do một tác giả trên mạng soạn) sau đây để cho ra nhiều thông tin thú vị.

```
matrix.cor <- function(x, y, digits=2, prefix="", cex.cor)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits=digits)[1]
    txt <- paste(prefix, txt, sep="")
    if(missing(cex.cor)) cex <- 0.8/strwidth(txt)

    test <- cor.test(x,y)
    # borrowed from printCoefmat
    Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,
                 cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
                 symbols = c("***", "**", "*", ".", " "))

    text(0.5, 0.5, txt, cex = cex * r)
    text(.8, .8, Signif, cex=cex, col=2)
}
```

Chúng ta quay lại với dữ liệu `lipid` bằng cách gọi hàm `matrix.cor` như sau:

```
pairs(lipid,lower.panel=panel.smooth, upper.panel=matrix.cor)
```



Đồ thị này cung cấp cho chúng ta tất cả hệ số tương quan giữa tất cả các biến số. Chẳng hạn như, hệ số tương quan giữa `age` và `bmi` quá thấp và không có ý nghĩa thống kê; giữa `age` và `hdl` hay giữa `age` và `hdl` cũng không có ý nghĩa thống kê; nhưng giữa `age` và `tc` thì bằng 0.22. Hệ số tương quan cao nhất là giữa `ldl` và `tc` (0.65) và `hdl` và `tc` (0.62). Giữ `hdl` và `ldl`, hệ số tương quan chỉ 0.35, nhưng có ý nghĩa thống kê (có sao!)

Chú ý biểu đồ trên chẳng những cung cấp hai thông tin chính (hệ số tương quan hay correlation coefficient, và vẽ biểu đồ tán xạ cho từng cặp biến số), mà còn cho biết hệ số tương quan nào có ý nghĩa thống kê (những kí hiệu sao). Hệ số tương quan càng cao, kích thước của font chữ càng lớn. Một biểu đồ rất ấn tượng!

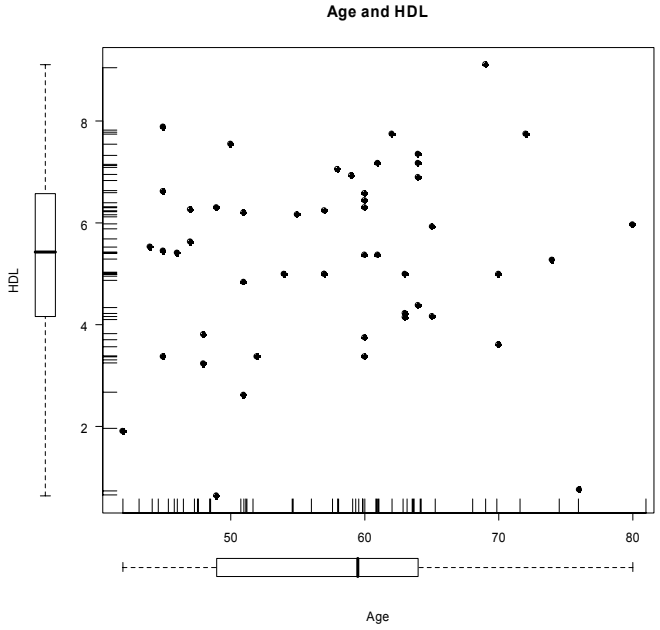
8.9 Một số biểu đồ "đa năng"

8.9.1 Biểu đồ tán xạ và hình hộp

Như trên đã trình bày, biểu đồ tán xạ giúp cho chúng ta hình dung ra mối liên hệ giữa hai biến số liên tục như độ tuổi `age` và `hdl` chẳng hạn. Và để làm việc này, chúng ta dùng hàm `plot`. Để tìm hiểu phân phối cho từng biến `age` hay `hdl` chúng ta có thể dùng hàm `boxplot`. Nhưng nếu chúng ta muốn xem phân phối của hai biến và đồng thời mối liên hệ giữa hai biến, thì chúng ta cần phải viết một vài lệnh để thực hiện việc này. Các lệnh sau đây vẽ biểu đồ tán xạ về mối liên quan giữa `age` và `hdl`, đồng thời vẽ biểu đồ hình hộp cho từng biến.

```
op <- par()
layout( matrix( c(2,1,0,3), 2, 2, byrow=T ),
        c(1,6), c(4,1),
      )

par(mar=c(1,1,5,2))
plot(hdl ~ age,
     xlab='', ylab='',
     las = 1,
     pch=16)
rug(side=1, jitter(age, 5) )
rug(side=2, jitter(hdl, 20) )
title(main = "Age and HDL")

par(mar=c(1,2,5,1))
boxplot(hdl, axes=F)
title(ylab='HDL', line=0)

par(mar=c(5,1,1,2))
boxplot(age, horizontal=T, axes=F)
title(xlab='Age', line=1)

par(op)
```

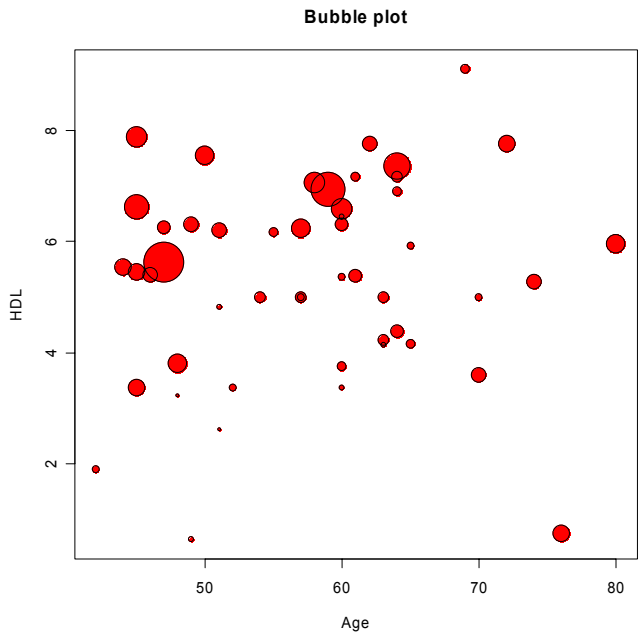
Và kết quả là:



8.9.2 Biểu đồ tán xạ với kích thước biến thứ ba

Biểu đồ trên thể hiện mối liên hệ giữa `age` và `hdl`, với mỗi điểm chấm có kích thước nhau nhau. Nhưng chúng ta biết rằng hdl cũng có liên hệ với triglyceride (`tg`). Để thể hiện một phần nào mối liên hệ 3 chiều này, một cách làm là vẽ kích thước của điểm tùy theo giá trị của `tg`. Chúng ta sẽ sử dụng thông số `cex` đã bàn trong phần đầu để vẽ mối liên hệ ba chiều này như sau:

```
> plot(age, hdl, cex=tg,
     pch=16,
     col=”red”,
     xlab="Age", ylab="HDL",
     main="Bubble plot")

> points(age, hdl, cex=tg)
```



8.9.3 Biểu đồ thanh và xác suất tích lũy

Để vẽ biểu đồ tần số của một biến liên tục chúng ta chủ yếu sử dụng hàm `hist`. Hàm này cho ra kết quả tần số cho từng nhóm (như nhóm độ tuổi chẳng hạn). Nhưng đôi khi chúng ta cần biết cả xác suất tích lũy cho từng nhóm, và muốn vẽ cả hai kết quả trong một biểu đồ. Để làm việc này chúng ta cần phải viết một hàm bằng ngôn ngữ R. Hàm sau đây được gọi là `pareto` (tất nhiên bạn đọc có thể cho một tên khác) được soạn ra để thực hiện mục tiêu trên. Mã cho hàm `pareto` như sau:

```
pareto <- function (x, main = "", ylab = "Value")
   {
    op <- par(mar = c(5, 4, 4, 5) + 0.1,
              las = 2)
   if( ! inherits(x, "table") ) {
     x <- table(x)
   }
   x <- rev(sort(x))
   plot( x, type = 'h', axes = F, lwd = 16,
         xlab = "", ylab = ylab, main = main )
   axis(2)
   points( x, type = 'h', lwd = 12,
           col = heat.colors(length(x)) )
   y <- cumsum(x)/sum(x)
   par(new = T)
   plot(y, type = "b", lwd = 3, pch = 7,
        axes = FALSE,
        xlab='', ylab='', main='')
   points(y, type = 'h')
   axis(4)
   par(las=0)
   mtext("Cumulated frequency", side=4, line=3)
```

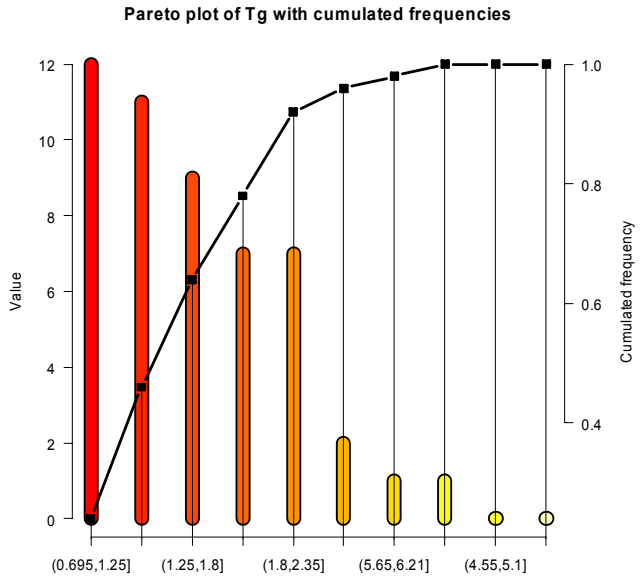
```
  print(names(x))
  axis(1, at=1:length(x), labels=names(x))
  par(op)
}
```

Bây giờ chúng ta sẽ áp dụng hàm pareto vào việc vẽ tần số cho biến tg (triglyceride) như sau. Trước hết, chúng ta chia tg thành 10 nhóm bằng cách dùng hàm `cut` và cho kết quả vào đối tượng `tg.group`.

```
> tg.group  <- cut(tg, 10)
```

Kế đến, chúng ta ứng dụng hàm `pareto`:

```
> pareto(tg.group)
 [1] "(0.695,1.25]" "(2.35,2.9]"   "(1.25,1.8]"   "(2.9,3.45]"   "(1.8,2.35]"
 [6] "(3.45,4]"     "(5.65,6.21]"  "(5.1,5.65]"   "(4.55,5.1]"   "(4,4.55]"
> title(main="Pareto plot of Tg with cumulated frequencies")
```



Trong biểu đồ này, chúng ta có hai trục tung. Trục tung phía trái là tần số (số bệnh nhân) cho từng nhóm `tg`, và trục tung bên phải là tần số tích lũy tích bằng xác suất (do đó, số cao nhất là 1).

8.9.4 Biểu đồ hình đồng hồ (clock plot)

Biểu đồ hình đồng hồ, như tên gọi là biểu đồ dùng để vẽ một biến số liên tục bằng kim đồng hồ. Tức là thay vì thể hiện bằng cột hay bằng dòng, biểu đồ này thể hiện bằng đồng hồ. Hàm sau đây (`clock`) được soạn để thực hiện biểu đồ hình đồng hồ:

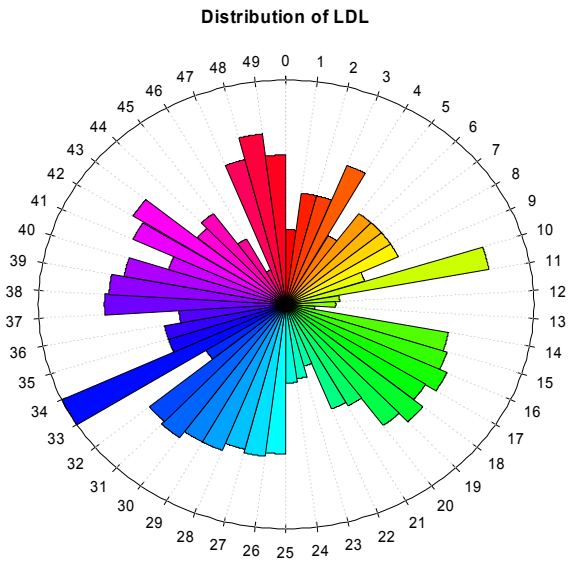
```
clock.plot <- function (x, col = rainbow(n), ...) {
```

```
  if( min(x)<0 ) x <- x - min(x)
  if( max(x)>1 ) x <- x/max(x)
  n <- length(x)
  if(is.null(names(x))) names(x) <- 0:(n-1)
  m <- 1.05
  plot(0,
       type = 'n', # do not plot anything
       xlim = c(-m,m), ylim = c(-m,m),
       axes = F, xlab = '', ylab = '', ...)
  a <- pi/2 - 2*pi/200*0:200
  polygon( cos(a), sin(a) )
  v <- .02
  a <- pi/2 - 2*pi/n*0:n
  segments( (1+v)*cos(a), (1+v)*sin(a),
            (1-v)*cos(a), (1-v)*sin(a) )
  segments( cos(a), sin(a),
            0, 0,
            col = 'light grey', lty = 3)
  ca <- -2*pi/n*(0:50)/50
  for (i in 1:n) {
    a <- pi/2 - 2*pi/n*(i-1)
    b <- pi/2 - 2*pi/n*i
    polygon( c(0, x[i]*cos(a+ca), 0),
             c(0, x[i]*sin(a+ca), 0),
             col=col[i] )
    v <- .1
    text((1+v)*cos(a), (1+v)*sin(a), names(x)[i])
  }
}
```

Chúng ta sẽ ứng dụng hàm clock để vẽ biểu đồ cho biến `ldl` như sau:

```
> clock.plot(ldl,
             main = "Distribution of LDL")
```

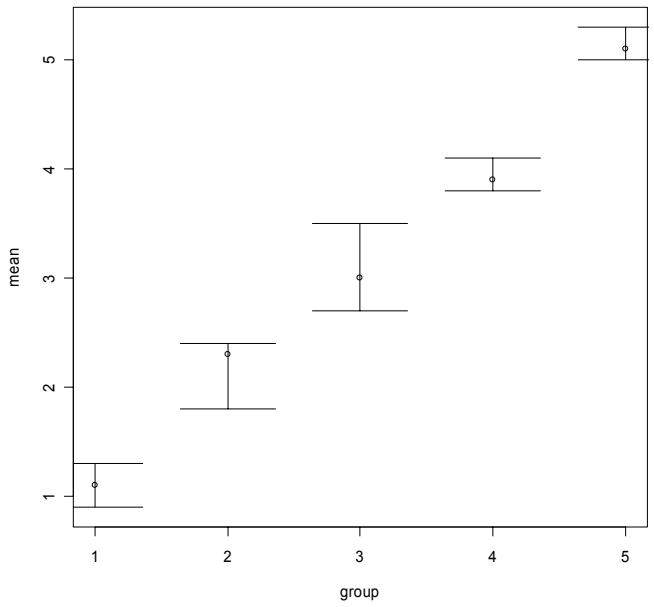
Và kết quả là:



8.9.5 Biểu đồ với sai số chuẩn (standard error)

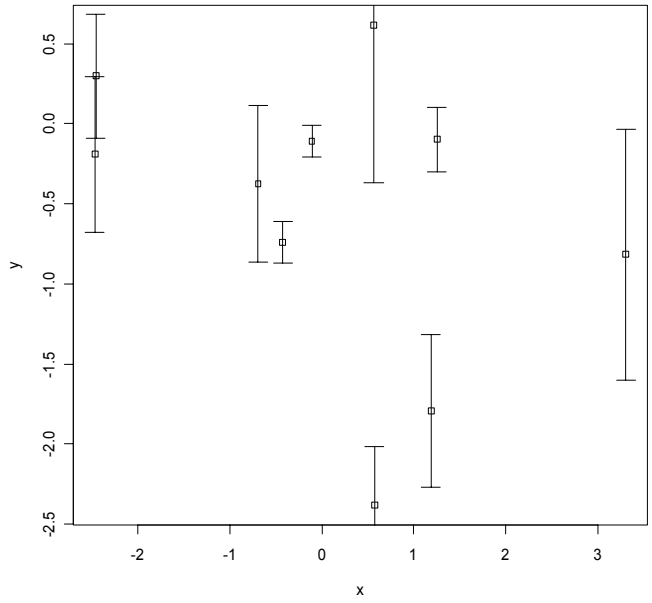
Trong biểu đồ sau đây, chúng ta có 5 nhóm (biến số `x` được mô phỏng chứ không phải số liệu thật), và mỗi nhóm có giá trị trung bình `mean`, và độ tin cậy 95% (`lcl` và `ucl`). Thông thường `lcl=mean-1.96*SE` và `ucl = mean+1.96*SE` (SE là sai số chuẩn). Chúng ta muốn vẽ biểu đồ cho 5 nhóm với sai số chuẩn đó. Các lệnh và hàm sau đây sẽ cần thiết:

```
> group <- c(1,2,3,4,5)
> mean <- c(1.1, 2.3, 3.0, 3.9, 5.1)
> lcl <- c(0.9, 1.8, 2.7, 3.8, 5.0)
> ucl <- c(1.3, 2.4, 3.5, 4.1, 5.3)
> plot(group, mean, ylim=range(c(lcl, ucl)))
> arrows(group, ucl, group, lcl, length=0.5, angle=90, code=3)
```



Sau đây là một mô phỏng khác. Chúng ta tạo ra 10 giá trị `x` và `y` theo luật phân phối chuẩn, và 10 giá trị sai số theo luật phân phối đều (`se.x` và `se.y` uniform distribution).

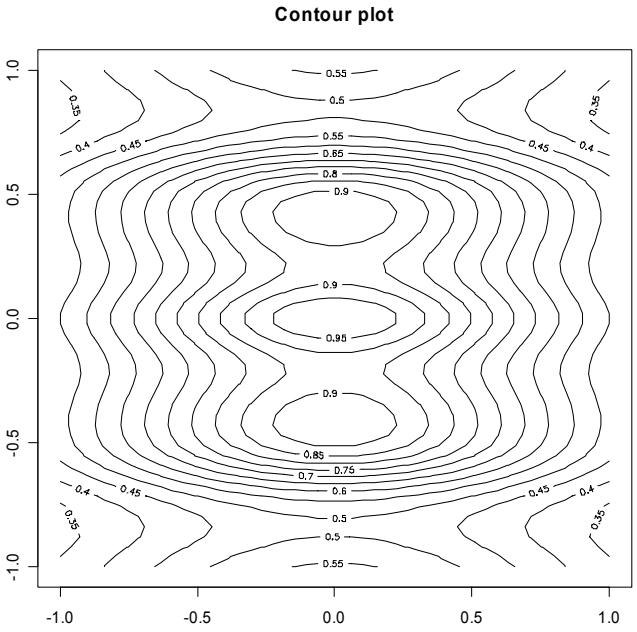
```
> x <- rnorm(10)
> y <- rnorm(10)
> se.x <- runif(10)
> se.y <- runif(10)
> plot(x, ypch=22)
> arrows(x, y-se.y, x, y+se.y, code=3, angle=90, length=0.1)
```



8.9.6 Biểu đồ vòng (contour plot)

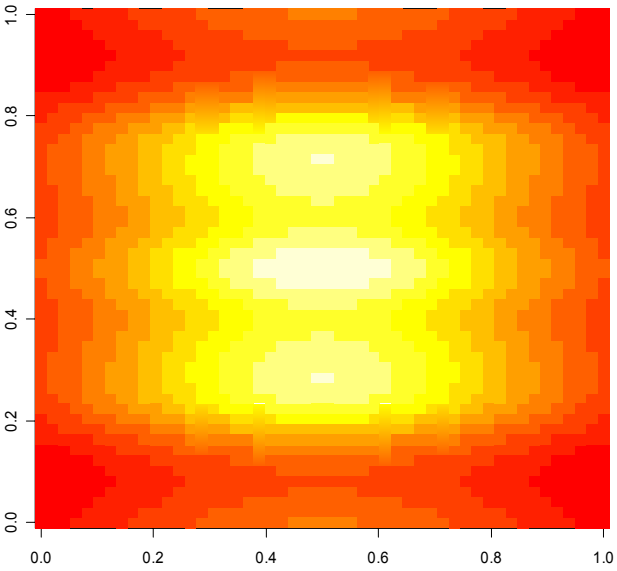
R có thể vẽ các đồ thị vòng với nhiều hình dạng khác nhau, tùy theo ý thích và dữ liệu. Trong các lệnh sau đây, chúng ta sử dụng kĩ thuật mô phỏng để vẽ đồ thị vòng cho ba biến số x, y và z.

```
> N <- 50
> x <- seq(-1, 1, length=N)
> y <- seq(-1, 1, length=N)
> xx <- matrix(x, nr=N, nc=N)
> yy <- matrix(y, nr=N, nc=N, byrow=TRUE)
> z <- 1 / (1 + xx^2 + (yy + .2 * sin(10*yy))^2)
> contour(x, y, z, main = "Contour plot")
```



Đồ thị này có thể chuyển thành một hình (image) bằng hàm `image`.

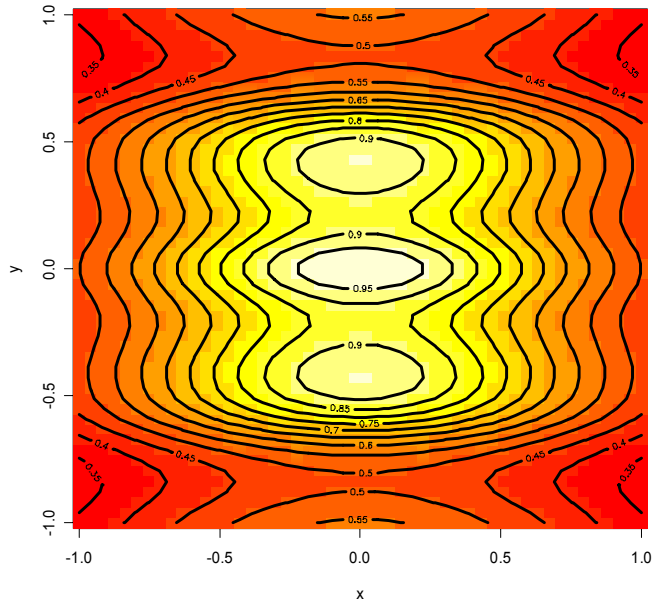
```
> image(z)
```



Một vài thay đổi nhỏ nhưng quan trọng:

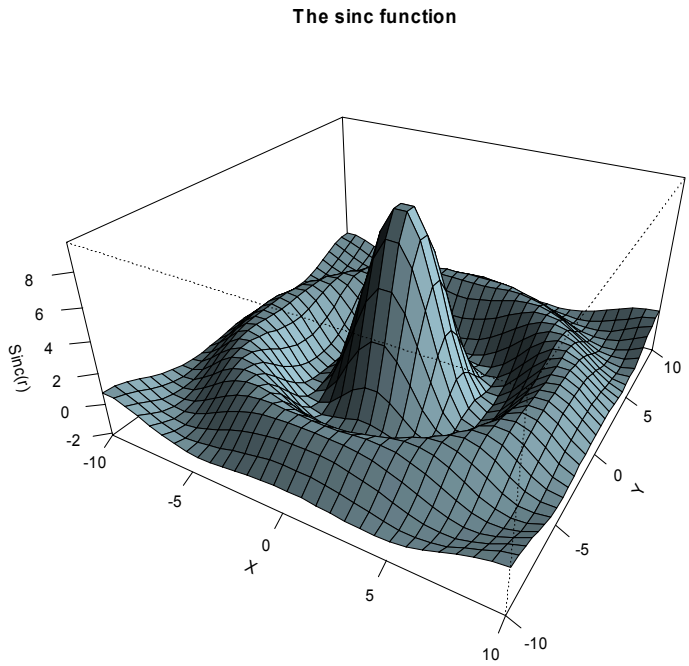
```
> image(x, y, z,
        xlab="x",
        ylab="y")
```

```
> contour(x, y, z, lwd=3, add=TRUE)
```



Sau đây là một vài thay đổi để vẽ biểu đồ theo hàm số sin và 3 chiều. Đồ thị này tuy xem "hấp dẫn", nhưng trong thực tế có lẽ ít sử dụng. Tuy nhiên, tôi trình bày ở đây để cho thấy một ví dụ về tính đa dụng của R.

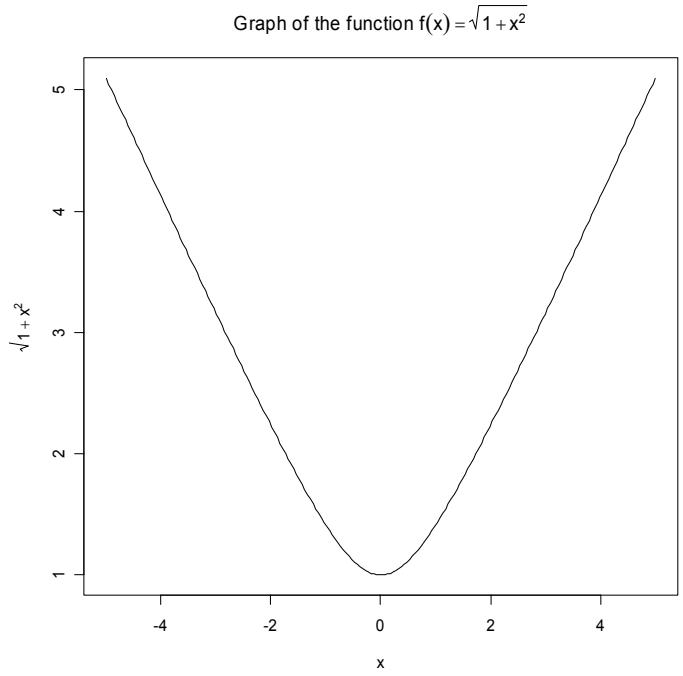
```
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white", mar=c(0,2,3,0)+.1)
> persp(x, y, z,
      theta = 30, phi = 30,
      expand = 0.5,
      col = "lightblue",
      ltheta = 120,
      shade = 0.75,
      ticktype = "detailed",
      xlab = "X", ylab = "Y", zlab = "Sinc(r)",
      main = "The sinc function"
)
> par(op)
```



8.9.10 Biểu đồ với kí hiệu toán

Đôi khi chúng ta cần vẽ biểu đồ với tựa đề có kí hiệu toán học. Trong đồ thị sau đây, chúng ta tạo ra một biến số x với 200 giá trị từ -5 đến 5, và $y = \sqrt{1+x^2}$. Để viết công thức trên, chúng ta cần sử dụng hàm `expression` như sau:

```
> x <- seq(-5,5,length=200)
> y <- sqrt(1+x^2)
> plot(y~x, type='l', ylab=expression(sqrt(1+x^2)))
> title(main=expression("Graph of the function
f"(x)==sqrt(1+x^2)))
```



Ngay cả tiếng Nhật cũng có thể thể hiện bằng R:

```
> plot(1:9, type="n", axes=FALSE, frame=TRUE, ylab="",
       main= "example(Japanese)", xlab= "using Hershey fonts")
> par(cex=3)
> Vf <- c("serif", "plain")
> text(4, 2, "\\#J2438\\#J2421\\#J2451\\#J2473", vfont = Vf)
> text(4, 4, "\\#J2538\\#J2521\\#J2551\\#J2573", vfont = Vf)
> text(4, 6, "\\#J467c\\#J4b5c", vfont = Vf)
> text(4, 8, "Japan", vfont = Vf)
> par(cex=1)
> text(8, 2, "Hiragana")
> text(8, 4, "Katakana")
> text(8, 6, "Kanji")
> text(8, 8, "English")
```

Chương này chỉ giới thiệu một số biểu đồ thông thường trong nghiên cứu khoa học. Ngoài các biểu đồ thông dụng này, R còn có khả năng vẽ những đồ thị phức tạp và tinh vi hơn nữa. Hiện nay, R có một package tên là `lattice` có thể vẽ những biểu đồ chất lượng cao hơn. `lattice`, cũng như bất cứ package nào của R, đều miễn phí, có thể tải về máy tính và cài đặt để sử dụng khi cần thiết.