
第 10 章: 低階繪圖函式

10: Low-Level Plotting Function

R 的 低階繪圖函式 (low-level plotting functions) 是將 點, 線, 座標與標籤等逐步形成圖形. 低階繪圖指令也可以在一個已經存在的圖形上, 加上其它的圖形元素, 例如, 額外的點, 線與標籤等等. 高階繪圖函式可以畫出常見統計圖形, 但是在某些情況下, 使用者希望畫出一些特殊要求的圖形. 例如坐標軸, 現有圖形上增加另一組資料點, 加入一些文字說明, 多個曲線代表之標簽等, 低階繪圖函式指令會在一個已經存在的繪圖上, 加上其它的圖形元素, 如額外的點, 線與標籤等等. R 低階繪圖過程是起始一個基本圖形 (高階或低階繪圖), 對現有圖形加入其他元素 (低階繪圖).

10.1 低階繪圖過程

低階繪圖必須從繪圖頁面呈現, 座標, 點與線, 多邊形, 顏色 等等考量, 才能繪製良好的統計繪圖. 低階繪圖從函式 `plot.new()` 與 `plot.window()` 起始.

- `plot.new()`: 開始繪製新圖.
- `plot.window()`: 設定座標系統.
- `plot.new()` 與 `plot.window()`: 都未真實繪圖.

底下指令比較使用高階繪圖函式與低階繪圖函式繪製同一圖形，參見圖 ??.

```

1 > ## high-level
2 > x.vec = 1:10
3 > y.vec = c(0.45, 1.46, 1.94, 1.45, 2.35,
4           2.87, 2.39, 2.69, 3.85, 4.07)
5 > par(mar = c(4, 4, 1, 1))
6 > plot(x = x.vec, y = y.vec,
7         ylim = c(0, 6),
8         xlab = "a custom x label",
9         ylab = "a custom y label")
10 > #
11 > ## low-level
12 > x.vec = 1:10
13 > y.vec = c(0.45, 1.46, 1.94, 1.45, 2.35,
14           2.87, 2.39, 2.69, 3.85, 4.07)
15 > plot.new()
16 > par(mar = c(4,4,3,1))
17 > plot.window(xlim = range(x.vec), ylim = c(0, 6))
18 > points(x = x.vec, y = y.vec)
19 > axis(1)
20 > axis(2)
21 > box()
22 > title(xlab = "a custom x label")
23 > title(ylab = "a custom y label")

```

上述低階繪圖函式中，個別指令簡單說明低階繪圖過程如下：

- `par()`: 在 `plot.new()` 之前，函式指令 `par(mar = c(4, 4, 3, 1))` 首先設定繪圖區域邊緣，設定 4 個圖畫邊緣 (margin) 空間，例如，
 - `par(mai = c(2, 2, 1, 1))` margin sizes in inches.
 - `par(mar = c(4, 4, 2, 2))` margin sizes in lines of text.
 - `par(pin = c(5, 4))` plot width and height in pins.
- `plot.window(xlim = xlims, ylim = ylims)` 設定座標軸的範圍：
 - `xlims + ylims`: 2 個元素的數值向量，可以設定座標軸的範圍。例如，`xlim = c(-pi, pi), ylim = c(-1, 1)`.
 - R 實際上繪出比設定值擴張 6% 的座標軸範圍。
 - 使用 `xaxs = "i"` 與 `yaxs = "i"` 可抑制 X-軸, Y-軸 擴張 6%。

```
1 > plot.window(xlim = c(0, 1),
2                  ylim = c(10, 20),
3                  xaxs = "i")
```

- `asp = 1` 設定 X 與 Y 軸為相同單位距離, 繪出正確圓形, 而非橢圓形.
- `log = x, log = y, log = xy` 可改成使用 `log` 對數轉換為單位為座標軸.
- `points(x = x.vec, y = y.vec)` 在 `(x, y)` 座標位置繪製點.
- `axis()` 繪製座標軸.
 - 1 `axis(side)`
 - `side = 1`: 圖下方 x axis.
 - `side = 2`: 圖左方 y axis.
 - `side = 3`: 圖上方 x axis.
 - `side = 4`: 圖右方 y axis.
- `box()` 在繪圖區域 (plot region) 繪製圖框.
- `title(xlab = "a custom x label")` 與 `title(ylab = "a custom y label")` 為繪圖註解.

```
1 title(main = str, sub = str, xlab = str, ylab = str)
```

- `main`: 圖上方, 主要註記.
- `sub`: 圖下方, 次要註記.
- `xlab`: X-軸 註解.
- `ylab`: Y-軸 註解.

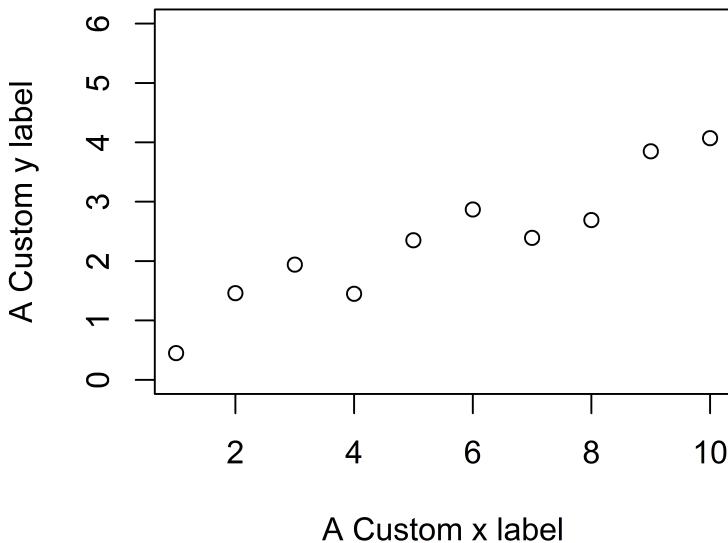


圖 10.1: 比較高階繪圖函式與低階繪圖函式繪製相同圖形

10.2 圖形引數函式 `par()`

當 R 的內部自動設定之繪圖有些時候無法達成使用者之要求，特別是使用者發表論文時，特定期刊之特定形式的圖形要求。R 提供許多“圖形引數”(graphics parameters)可以控制特別需求，如色彩，尺寸等。R 圖形引數都有一特定名稱，透過函式 `par()`，更改設定。圖形引數設定有兩種方式：(a) 對目前的使用的繪圖裝置設定永遠改變設定，所有利用此繪圖裝置輸出之繪圖，都受影響；(b) 暫時改變設定，僅對單一使用的繪圖函式做改變。

在此具體描述一些常用的圖形引數，函式 `par()` 可顯示完整的圖形引數，函式 `par()` 也可用來更改內部自動設定之圖形引數。在上一章，第 ?? 小節中，簡要說明一些常用高階繪圖引數，如表 ??，其中部分引數屬於 `par()` 圖形引數，但有些如 `axes()` 不是圖形引數，而是一些高階繪圖函式，如 `plot()` 函式內的引數。

圖形引數用以下的形式設定

```
1 > par(par.name = par.value)
```

`par.name` 是圖形引數名稱, 它可以用於 `par()` 或某些 (高階或低階) 繪圖函式中當引數, `par.value` 是使用者想給 `par.name` 圖形引數之設定值. 常見繪圖引數為

```

1 xlog      ylog      adj       ann      ask
2 bg        bty       cex       cex.axis cex.lab
3 cex.main  cex.sub   cin       col      col.axis
4 col.lab   col.main  col.sub   cra      crt
5 csi       cxy       din       err      family
6 fg        fig       fin       font    font.axis
7 font.lab  font.main font.sub gamma   lab
8 las       lend      lheight   ljoin   lmitre
9 lty       lwd       mai       mar     mex
10 mfcoll  mfg       mfrow    mpg     mkh
11 new      oma      omd      omi     pch
12 pin      plt      ps       pty     smo
13 srt      tck      tcl      usr     xaxp
14 xaxs    xaxt      xpd      yaxp   yaxt

```

許多圖形引數可以在高階繪圖或低階繪圖函式直接使用, 可依參看常見低階繪圖函式, 但下列圖形引數只能透過 `par()` 改變設定.

```

1 ask  fig   fin   lheight mai mar
2 mex  mfcoll mfrow mfg   new
3 oma  omd   omi   pin
4 plt  ps    pty   usr
5 xlog ylog ylbias

```

10.2.1 永遠改變設定圖形引數

函式 `par()` 可用於查看與修改現有的繪圖裝置的引數列表 (list).

- `par()`

函式 `par()` 內若沒有引數, 則 R 顯示或回傳所有圖形引數列表和現有的繪圖裝置的圖形引數之設定值.

- `par(c("par.name1", "par.name2"))`

在函式 `par()` 內設定某一特定圖形引數名稱之文字向量 `"par.name1", "par.name2"` 等, R 回傳指定的引數名稱之設定值列表, 例如 `par(c("col", "lty"))`.

- `par(par.name1 = par.value1, par.name2 = par.value2)`

在函式 `par()` 內永遠改變某一特定圖形引數之設定值, 使用函式 `par()` 設定圖形引數會永久改變引數設定值, 以後所有在當前中用使繪圖裝置圖形函式都會受這些新設定值所影響. 這種永遠改變不是使用者想要的, 通常使用者只是想設定一些圖形引數, 繪製一些圖片, 然後恢復原始設定值. 因此使用者可以先保存 `par()` 的原始設定值, 繪製一些圖片完成後, 再來重新載入這些原始設定值便可恢復原始設定值.

```
1 > oldpar <- par(no.readonly = TRUE)
2
3     some plotting commands
4
5 > par(oldpar)
```

10.2.2 暫時改變設定圖形引數

圖形引數可以以作為許多繪圖函式之引數的形式, 暫時改變設定僅僅影響現有的繪圖函式使用, 而不影響其他繪圖函式使用. 例如:

```
1 > plot(x, y, pch = "+")
```

產生一個以加號 `+` 為點符號的散點圖, 這指令不會影響後面的繪圖函式原始內部自動設定值. 但並非所有的圖形引數可以成為任何繪圖函式之引數, 有些時候還是需要通過 `par()` 設定和重新設定圖形引數.

10.3 圖畫邊緣

在 R 中, 單獨的圖稱為 圖片 (figure). 單獨的圖片包括一個 邊緣 (margin) 或稱為 內邊緣 (inner margin, margin), 內邊緣可包括 坐標軸標籤, 標題 等. 若圖片排除邊緣, 僅有坐標軸包圍的區域稱為 繪圖區域 (plot region), 參見 圖 10.2.

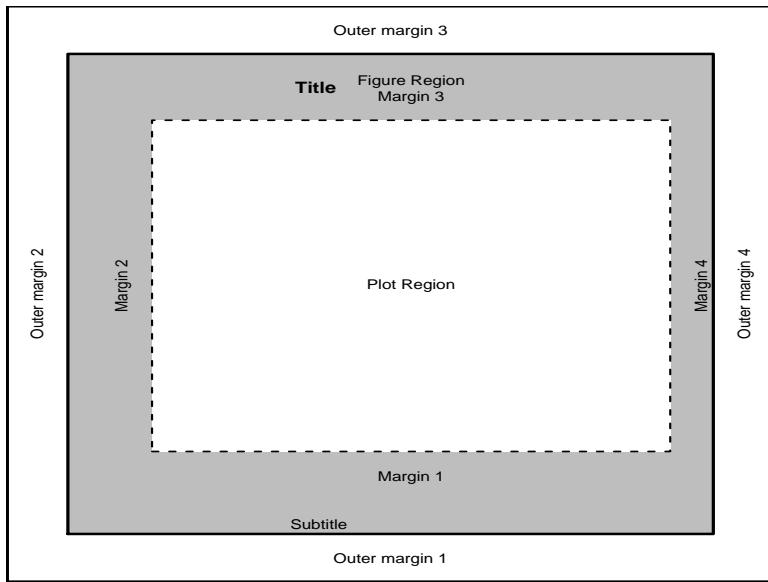


圖 10.2: R 圖片, 繪圖區域 與 圖畫邊緣 的解剖示意圖

低階繪圖函式 `plot.new()` 開始新的繪圖, R 自動設定長方形的繪圖區域 (plot region), 包含設定 4 個圖畫邊緣 (margin). 控制圖片版面的圖形引數有 `mar()`, `mai()` 等, 參見表 10.1. 簡單指令如下:

- `mai = c(0.9792, 0.7872, 0.7872, 0.4032)`, `mai = c(1, 0.5, 0.5, 0)`, ..., etc.

引數 `mai` 設定一個具有 4 個元素的數值向量, 設定 4 個內邊緣 (inner margin), 分別表示內邊緣的底部 (bottom), 左側 (left), 頂部 (top) 和右側 (right) 的空間, 單位是英寸.

- `mar = c(5.1, 4.1, 4.1, 2.1)`, `mar = c(4, 2, 2, 1)`, ..., etc.

引數 `mar` 作用與引數 `mai` 作用相同, 設定 4 個內邊緣但 `mar` 的度量單位採用文本行數為單位 (text lines), 引數 `mar` 內的數值是文本行數的倍數. 這些引數的 R 預設值常常有點大, 右側邊緣常常是不需要的, 沒有標題時, 頂部邊緣也不需要, 但通常保留底部邊緣和左側邊緣, 用來繪製坐標軸和刻度標記.

表 10.1: 常見關於圖畫邊緣的設定引數

引數	使用方式	簡要說明
<code>din, fin, pin</code>	<code>fin = c(6, 4)</code>	絕對尺寸, 分別為 圖形裝置 (graphic device), 圖片 (figure) 與 繪圖區域 (plot region), 以 inches 為單位.
<code>fig</code>	<code>fig = c(0, 0.5, 0, 1)</code>	定義圖片 區域大小 為 圖形裝置 區域大小 的 比例.
<code>font</code>	<code>font = 3</code>	定義字型, 1 = plain, 2 = italic, 3 = bold, 4 = bold italic, 5 = 符號.
<code>mai, mar</code>	<code>mar = c(3, 3, 1, 1) + 0.1</code>	繪圖區域 的 4 個內邊緣尺寸, <code>mai</code> 以 inches 為單位. <code>mar</code> 以文本行數 為單位.
<code>mex</code>	<code>mex = 0.7</code>	定義邊緣文字的相對大小.
<code>mfg</code>	<code>mfg = c(2, 2, 3, 2)</code>	定義多重圖片環境的位置.
<code>mfrow, mfcoll</code>	<code>mfrow = c(2, 2)</code>	定義多重圖片環境, <code>mfrow</code> (<code>mfcoll</code>) 分別以 row (column) 為主依序放置.
<code>new</code>	<code>new = F</code>	邏輯引數, 代表是否要使用一個新的 高階繪圖 或新的 圖形裝置.
<code>oma, omi, omd</code>	<code>oma = c(0, 0, 4, 0)</code>	定義邊緣尺寸, 分別以 文字行數, inches, 或以 圖形裝置 區域的比例 計算.
<code>plt</code>	<code>plt = c(0.1, 0.9, 0.1, 0.9)</code>	定義 繪圖區域 大小 為 形 區域大小 的比例.
<code>pty</code>	<code>pty = "s"</code>	定義 繪圖區域 的形狀, "s" = 正方形, "m" = 使用最大區域.
<code>usr</code>	<code>usr = c(x1, x2, y1, y2)</code>	設定 繪圖區域 X 與 Y 座標軸的範圍上下界.

邊緣設定例如:

- `par()`: 使用在 `plot.new()` 之前, 設定 4 個 圖畫邊緣 (margin) 空間.

- margin sizes in inches:

```
par(mai = c(2, 2, 1, 1))
```

- margin sizes in lines of text:

```
par(mar = c(4, 4, 2, 2))
```

- plot width and height in pins

```
par(pin = c(5, 4))
```

10.4 圖形元素 Graphical Elements

R 圖形由點 (point). 線 (line), 文本 (text) 和 多邊形或閉合區域 (polygons, filled regions) 所構成. 圖形參數就是用來控制這些 圖形元素 (graphical elements), 這些繪圖基本元素 (drawing primitives) 為

- Points
- Connected Line Segments
- Straight Lines Across A Plot
- Disconnected Line Segments
- Arrows
- Rectangles
- Polygons
- Text
- Legends

R 有一些非常有用的低級繪圖函式，參見 表 10.2，可以繪製繪圖基本元素。見 表 10.3，與 圖 10.3 – 圖 10.5。

表 10.2: 常用的低階繪圖函式

函式	說明
<code>points(x, y)</code>	在現有的圖形上加點。
<code>lines(x, y)</code>	在現有的圖形上加線。
<code>text(x, y, labels =)</code>	在現有的圖形的 (x, y) 向量坐標點上加文字。
<code>abline(a, b)</code>	在現有的圖形上加畫一條截距為 a 和斜率為 b 的直線。
<code>axis(side, vec)</code>	在現有的圖形上加上坐標軸。
<code>legend(x, y, legend)</code>	用來在現有的圖形上，在指定的 (x, y) 坐標位置繪製圖例。
<code>title(main = , sub =)</code>	用來在現有的圖形上加入標題。
<code>mtext(text, side = 3, line = 0)</code>	在現有的圖形之邊緣加上文字。
<code>polygon(x, y)</code>	以 x, y 為端點繪製多邊形圖。
<code>segments(x0, y0, x1, y1)</code>	在現有的圖形的點 $(x_0, y_0), (x_1, y_1)$ 繪製線段。
<code>arrows(x0, y0, x1, y1)</code>	在現有的圖形的點 $(x_0, y_0), (x_1, y_1)$ 繪製具有箭頭的線段。
<code>rect(x0, y0, x1, y1)</code>	在現有的圖形的左下點 (x_0, y_0) 與右上方點 (x_1, y_1) 繪製長方形圖。
<code>curve(expression)</code>	在現有的圖形繪製曲線。
<code>frame()</code>	完成現有圖形，並前進到一個新的圖形。
<code>box()</code>	在現有的圖形畫上一個盒狀邊界。
<code>rug(x)</code>	現有圖形上加上 “rug” (類似地毯的直線)，代表圖形上資料的點。
<code>locator(n, type = "n")</code>	使用滑鼠從現有圖形上點選 n 個點，回傳這些點的座標值。
<code>identify(x, y, labels)</code>	使用滑鼠從現有圖形上選擇一點，回傳這點的索引值 (index number) 或加上 <code>label</code> 值。

低階繪圖函式通常需要指定座標位置，座標位置是指即現有的圖形上，或是由高階繪圖函式所建立的圖形上。座標系統中之座標通常用兩個向量 x 和 y 表示，也可以由一個 2 欄 (column) 的矩陣表示。如果在互動式繪圖之下作圖，可以用 `locator()` 函式來從現有圖形中直接傳回與輸入坐標位置。

無論高階繪圖函式或低階繪圖函式常見圖形引數包含 `pch`, `lty`, `lwd`, `lty`, `col`, `cex` 等等.

- `pch` 顯示點的符號, 參見 圖 10.3.

- `pch = "+"`, `pch = "u"`, `pch = "2"`, ..., etc.

使用 `pch` 顯示點的文字符號, 內定預設值常是一個小圓圈, 但預設值會隨圖形列印機器設定而有些差異. 顯示的點有些高於或低於適當的位置, 可用 `pch = "."` 顯示符號, 顯示的點會於中心的位置. 用 `pch = "2"` 顯示文字 "2" 在位置上.

- `pch = 1`, `pch = 2`, ..., etc.

當引數為 `pch = k`, 則設定 k 是一個 0–25 之間的正整數, 顯示一個相對應於 k 之特定符號, 內定預設值為 1.

- `pch = 21–25` 位於 21–25 之間的符號, 看上去是與前面符號重複, 但是用不同的顏色顯示.

- `bg` 設定 `pch = 21:25` 內填滿顏色, 參見顏色的設定.

- `pch` 可以是 32–255 範圍內的字元和數字, 顯示一個相對應字元.

```
1 > plot.new()
2 > legend(locator(1), as.character(0:25), pch = 0:25)
```

- `lty` 設定線條類型. 但預設值會隨圖形列印機器設定而有些差異, 參見 圖 10.4.

- `lty`: 0–6 正整數, 設定線條類型.

`0 = blank`, `1 = solid`, `2 = dashed`, `3 = dotted`,
`4 = dotdash`, `5 = longdash`, `6 = twodash`

- `lty`: 線條形式文字.

`"blank"`, `"solid"`, `"dashed"`, `"dotted"`,
`"dotdash"`, `"longdash"` `"twodash"`.

- `lty = "11"`: 高密度 dotted line.

- `lty = "33"`: 短 dashed line.
- `lty = "1333"`: dot-dashed line.

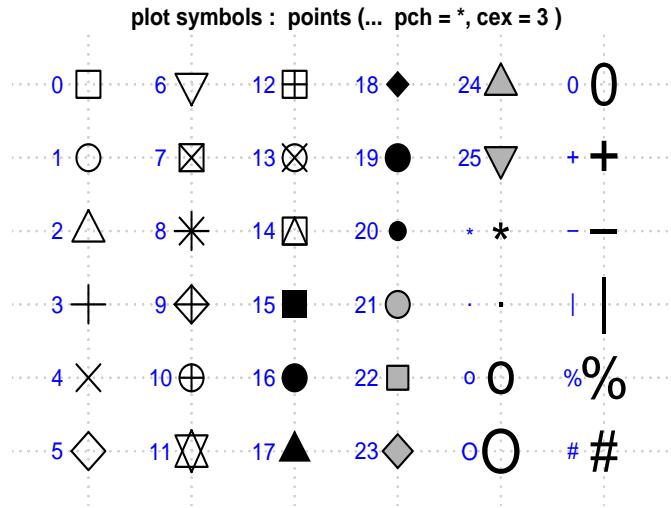
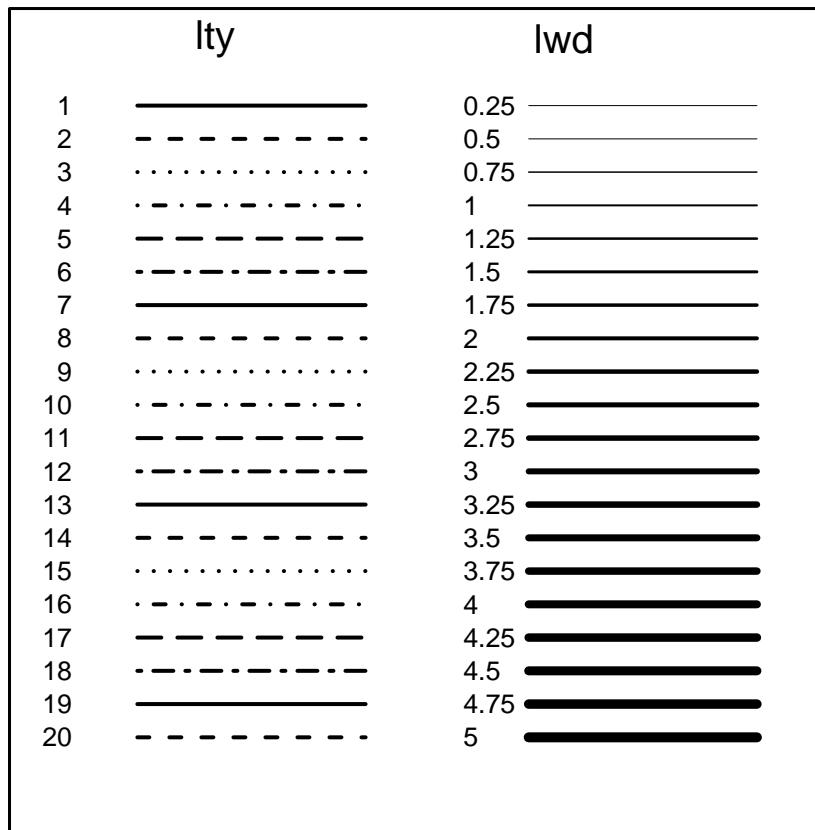


圖 10.3: `pch` 顯示顯示點的特定符號

- `lwd` 設定線條寬度, 參見 圖 10.4.
 - `lwd = 1`: 內定預設值以 `lwd = 1` 為 標準寬度.
 - `lwd = k`: `lwd = 2, …, etc.`
畫線時, 當設定為 `lwd = k` 時, 則設定線條寬度為 標準'寬度 的 k 倍數
寬度. 座標軸線條 和 函數 `lines()` 等產生之線條, 都會受到 `lwd` 設定
的影響, 請參見 圖 10.4.
- `type` 類似用高階繪圖函式 `plot()` 之引數 `type` 設定繪點或繪線. 設定繪圖
在 (x, y) 座標上的顯示方式, 參見 圖 10.5.
 - `p`: 畫點.
 - `l`: 畫線.

圖 10.4: `lty` 與 `lwd` 引數設定高階繪圖函式在畫線與寬度時之顯示方式

- **b**: 畫點同時在點與點之間, 畫線連結.
 - **h**: 從點到 x -橫軸 畫垂直線.
 - **s**, **S**: 階梯函式 (step function), 小 **s** 為左連續函式, 大 **S** 為右連續.

 - **o**: 畫線同時穿過畫點.
 - **n**: 不畫任何點與線, 但容許畫坐標軸且建立坐標系統, 用於後面用低階繪圖函式作圖.
- **col** 設定顏色.
 - **col = 1, ..., col.axis, col.lab, col.main, col.sub, ..., etc.**
引數 **col = 1** 設定點, 線, 文本, 填充區域 (filled region) 和圖像 (image) 的顏色設置, 引數 **col = k** 所設定的顏色值 (value) **k**, 是 R 調色板的數值, 而 R 調色板的詳細數值請參見 [help\(palette\)](#).
 - **col** 也可設定成爲顏色之文字名稱, 例如 **col = "black"**.
 - **col.axis** 設定座標軸線標注.
 - **col.lab** 設定 **x** 和 **y** 座標軸之標籤.
 - **col.main, col.sub** 設定主標題和次標題, 等等的顏色.

```

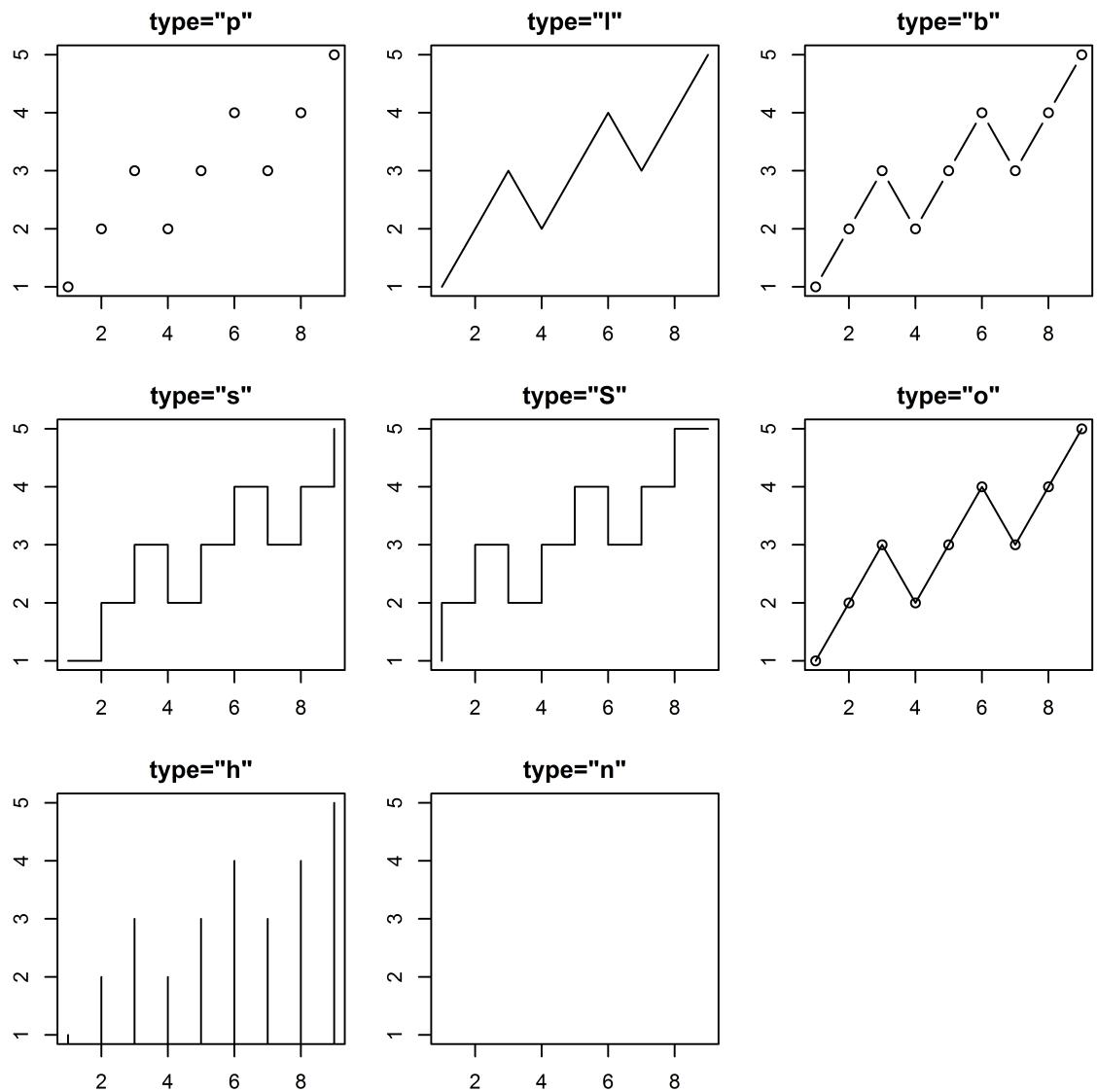
1 > ColorName <- colors()
2 > ColorName
3 [1] "white"           "aliceblue"        "antiquewhite"
4 [4] "antiquewhite1"   "antiquewhite2"   "antiquewhite3"
5 .....
6 [652] "yellow"         "yellow1"          "yellow2"
7 [655] "yellow3"        "yellow4"          "yellowgreen"

```

- **cex = 1, cex = 1.5, ..., etc.**

設定文字字體 (character size) 大小之放大率, 內定預設值爲 1. 這個設定值表示期望字體大小 (包括繪圖字體) 相對內定預設值大小的比率.

- **cex.axis** 設定座標軸標注的字體相對大小的比率.

圖 10.5: `type` 引數設定繪圖函式在 (x, y) 顯示方式

- `cex.lab` 設定座標軸標籤的字體相對大小的比率.
 - `cex.main` 設定主標題的字體相對大小的比率.
 - `cex.sub` 設定次標題的字體相對大小的比率.
- `font = 1, …, etc.`
- 設定圖形中文字的字體類型. 1 對應于純文本, 2 對應粗體, 3 對應斜體, 4 對應粗斜體, 5 對應符號體 (包括希臘字母).
- `font.axis` 設定座標軸標注的字體類型.
 - `font.lab` 設定座標軸標籤的字體類型.
 - `font.main` 設定主標題的字體類型.
 - `font.sub` 設定次標題的字體類型.
- `adj = 0.5, adj = -0.1, …, etc.`
- 設定調整文本 (`text`) 相對於圖形的之相對位置, 內定預設值為 0.5. 0 表示左對齊, 1 表示右對齊, 0.5 表示圖形位置的水準居中, 設定實際值表示出現在圖形位置左側的文本字體寬度的比率, 因此 -0.1 表示會在文本和圖形位置間留下文本字體寬度的 10%.

表 10.3: 常見圖形引數中之圖形元素

圖形元素	用法	說明
<code>pch</code>	<code>pch = "*", pch = 4</code>	設定圖形內“點”的符號或文字.
<code>lty</code>	<code>lty = 2</code>	設定圖形內線條的類型.
<code>lwd</code>	<code>lwd = 2</code>	設定圖形內線條的寬度.
<code>col</code>	<code>col = 2</code>	設定圖形內點, 線, 文字等的顏色.
<code>cex</code>	<code>cex = 1.5</code>	設定圖形內的文字與符號相對大小尺吋.
<code>font</code>	<code>font = 1</code>	設定圖形內文字的字體類型.

10.5 低階繪圖函式: points()

低階繪圖函式 `points(x, y)` 在現有圖形的座標 (`x, y`) 位置上加一組 點 (point). 參見圖 10.6.

```
1 points(x, y = NULL, pch = 1, type = "p", col = str, bg, ...)
```

常見引數為:

- `x, y` 在現有圖形的座標 (`x, y`) 位置上加點, `x, y` 為數值向量.
- `pch` 顯示點的符號, 0–25 之間的正整數, 參見 圖 ??.
- `type` 設定繪點或繪線.
- `col` 設定顏色.
- `bg` 設定 `pch = 21:25` 內填滿顏色.

```
1 ## 1 points()
2 > set.seed(1)
3 > x <- 1:30
4 > z <- x^2
5 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
6 > plot(x,y, type = "n", axes = TRUE, ann = FALSE)
7 > points(x,y,pch = 1,cex = 1)
8 > title(main = "points")
```

10.6 低階繪圖函式: lines()

低階繪圖函式 `lines(x, y)` 函式在現有的圖形上加一條 線 (line). 參見圖 10.6.

```
1 lines(x, y, lty = str, lwd = num, col = str)
```

常見引數為:

- `x, y` 在現有圖形的座標 (`x, y`) 位置上加線, `x, y` 為數值向量.

- `lty` 設定線條類型, 0–6 正整數, 設定線條類型. `0 = blank, 1 = solid, 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash` 參見圖 10.4.
- `lwd` 設定線條寬度, 參見 圖 10.4.
- `type` 設定繪點或繪線, 參見 圖 10.5.
- `col` 設定顏色.

```

1 ## 2 lines()
2 > set.seed(1)
3 > x <- 1:30
4 > z <- x^2
5 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
6 > plot(x,y, type = "n", axes = TRUE, ann = FALSE)
7 > points(x,y,pch = 1,cex = 1)
8 > lines(x,y,lty = 1, lwd = 1.5)
9 > title(main = "lines")

```

10.7 低階繪圖函式: abline()

低階繪圖函式 `abline()` 在現有的圖形上加畫一條截距為 `a` 和斜率為 `b` 的直線.
參見 圖 10.6.

```

1 abline(a = NULL, b = NULL, h = NULL, v = NULL, reg = NULL,
2        coef = NULL, untf = FALSE, ...)

```

常見引數為:

- `h = y`: 函式 `abline(h = y)` 可畫出在 $Y = y$ 平行於 X -軸 之水平線.
- `v = x`: 函式 `abline(v = x)` 可畫出在 $X = x$ 垂直於 X -軸 之垂直線.
- `abline(lm.obj)` 繪出由線性模型函式建立之線性模型物件 `lm.obj` 的線性迴歸線.
- `lwd`] 設定線條寬度.

- `lty`] 設定線條類型.
- `col`] 設定顏色.

```

1 > ## 3 abline()
2 > set.seed(1)
3 > x <- 1:30
4 > z <- x^2
5 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
6 > plot(x, y, type = "n", axes = TRUE, ann = FALSE)
7 > points(x, y, pch = 1, cex = 1)
8 > abline(2, 100, lty = 1, lwd = 2, col = "red")
9 > abline(h = 1000, lty = 2, lwd = 1.5, col = "blue")
10 > abline(v = 10, lty = 3, lwd = 1.5, col = "blue")
11 > abline(lm(y~x), lty = 1, lwd = 2, col = "black")
12 > title(main = "abline()")

```

10.8 低階繪圖函式: symbols()

低階繪圖函式 `symbols()` 在現有的圖形加上特殊符號. 參見圖 10.6.

```

1 symbols(x, y = NULL, circles, squares, rectangles, stars,
2   thermometers, boxplots, inches = TRUE, add = FALSE,
3   fg = par("col"), bg = NA,
4   xlab = NULL, ylab = NULL, main = NULL,
5   xlim = NULL, ylim = NULL, ...)

```

可使用下列常見的引數:

- `x, y` 在現有圖形的座標 (`x, y`) 位置上特殊符號, `x, y` 為數值向量.
- `circles`: 圓半徑向量.
- `squares`: 正方形邊長向量.
- `rectangles`: 方形 “wide” “height” 2 columns 矩陣.
- `stars`: 3 columns 以上矩陣, 從中心到端點距離
- `thermometers`: 2, 3, 4 columns 矩陣 代表 “wide” “height” + W, H 的 “proportion”.

- **boxplots**: 5 columns 矩陣
非別為盒狀圖的 “wide” “height”, “lower” “upper” whiskers, “proportion” median line.
- **add = FALSE** 是否在現有圖形加上 symbols.
- **fg** 盒狀圖的顏色.
- **bg** 盒狀圖填滿的顏色.

```

1 > ## 5 symbols()
2 > set.seed(1)
3 > x <- 1:30
4 > z <- x^2
5 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
6 > plot(x, y, type = "n", axes = TRUE, ann = FALSE)
7 > symbols(x, y, circles = (c(1:30)/15.0),
8           inch = FALSE, lwd = 1.5, add = TRUE)
9 > title(main = "symbols")

```

10.9 低階繪圖函式: segments()

低階繪圖函式 **segments** 在現有圖形的座標點之間繪製線段. 參見圖 10.6.

```

1 segments(x0, y0, x1 = x0, y1 = y0,
2           col = par("fg"), lty = par("lty"), lwd = par("lwd"),
3           ...)

```

常見的引數:

- **(x0, y0, x1, y1)** 在現有圖形的座標點 (x_0, y_0) 與 (x_1, y_1) 之間繪製線段.
- **lty** 設定畫線之類型.
- **lwd** 設定畫線之寬度.
- **col** 設定一個特定色彩.

```
1 > ## 5 segments()
2 > x <- 1:30
3 > z <- x^2
4 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
5 > plot(x, y, type = "n", axes = TRUE, ann = FALSE)
6 > segments(c(1:30), rep(0,30), x, y, lwd = 1.5)
7 > title(main = "segments")
```

10.10 低階繪圖函式: arrows()

低階繪圖函式 `arrows()` 在現有的圖形的點繪製具有箭頭的線段. 參見 圖 10.6 與 圖 10.7.

```
1 arrows(x0, y0, x1 = x0, y1 = y0,
2         length = 0.25, angle = 30, code = 2,
3         col = par("fg"), lty = par("lty"),
4         lwd = par("lwd"), ...)
```

可使用下列常見的引數:

- `(x0, y0, x1, y1)` 在現有圖形的座標點 (x_0, y_0) 與 (x_1, y_1) 之間繪製具有箭頭的線段.
- `length = 0.25` 設定箭頭大小.
-
- `code` 設定箭頭的位置.
 - `code = 1` 設定箭頭的位置在點 (x_0, y_0) .
 - `code = 2` 設定箭頭位置在點 (x_1, y_1) .
 - `code = 3` 設定箭頭位置同時出現在點 (x_0, y_0) 與點 (x_1, y_1) 兩端.
- `angle = 30` 設定箭頭與線段的角度.
- `lwd` 設定線條寬度.
- `lty` 設定線條類型.

- `col` 設定顏色.

```
1 > ## 6 arrows()
2 > x.arrow <- seq(1, 30, by = 3)
3 > plot(x.arrow, x.arrow, type = "n", axes = TRUE, ann = FALSE)
4 > arrows(x.arrow, rep(15,10), x.arrow, x.arrow,
5           lwd = 1.5, length = 0.1, code = 2)
6 > title(main = "arrows")
```

10.11 低階繪圖函式: polygon()

低階繪圖函式 `polygon(x, y, ...)` 畫出以 (x, y) 向量坐標點為端點(頂點)之多邊形(polygon). 參見圖 10.8.

```
1 polygon(x, y = NULL, density = NULL, angle = 45,
2           border = NULL, col = NA, lty = par("lty"),
3           ..., fillOddEven = FALSE)
```

常見的引數:

- `x, y` 在現有圖形的座標 (x, y) 向量坐標點位置上繪製多邊形.
- `lty` 設定畫線之類型.
- `lwd` 設定畫線之寬度.
- `col` 指定一個特定色彩, 來填滿多邊形之內部.

```
1 > ## 8 ploygon()
2 > x <- 1:30
3 > z <- x^2
4 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
5 > plot(x, y, type = "n", axes = TRUE, ann = FALSE)
6 > polygon(c(1, 15, 20, 30, 15),
7           c(1000, 100, 2000, 1500, 3000), lwd = 1.5)
8 > title(main = "polygon")
```

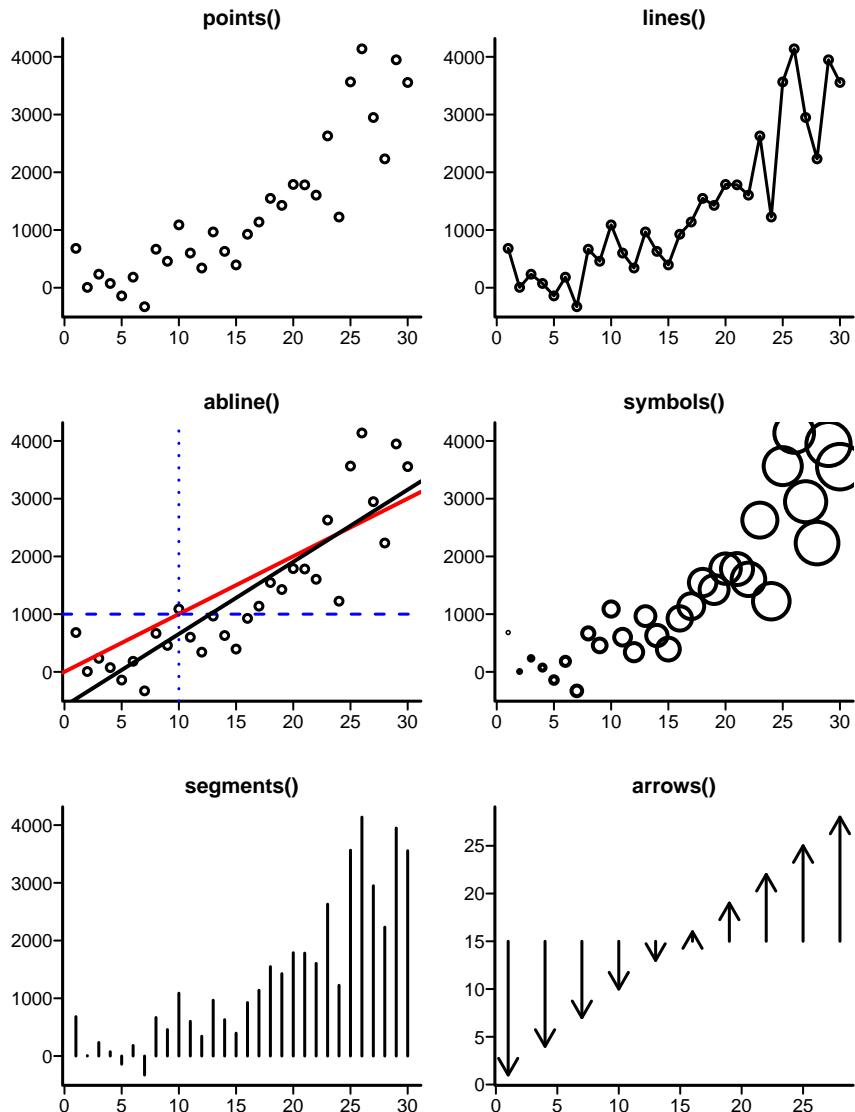


圖 10.6: 常見低階繪圖函式 I

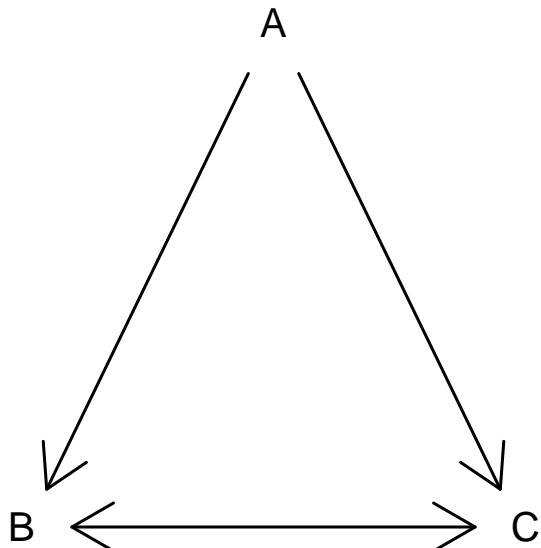


圖 10.7: 低階繪圖函式 arrows()

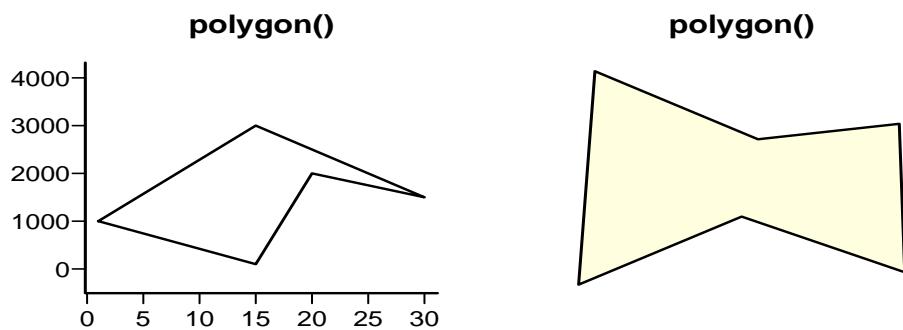


圖 10.8: 低階繪圖函式: polygon()

10.12 低階繪圖函式: rect()

低階繪圖函式 `rect()` 在現有的圖形繪製長方形圖. 參見 圖 10.9.

```
1 rect(xleft, ybottom, xright, ytop, density = NULL, angle = 45,
2       col = NA, border = NULL, lty = par("lty"), lwd = par("lwd"),
3       ...)
```

可使用下列常見的引數:

- `xleft, ybottom` 長方形圖左下座標點.
- `xright, ytop` 長方形圖右下座標點.
- `density = NULL` 設定長方形圖內的線條密度.
`density = NULL, density = 0, density = negative value` 表示長方形圖內沒有線條.
- `angle` 設定長方形圖內的線條呈現的角度.
- `col` 設定長方形圖內的線條之間空隙的顏色.
- `lwd` 設定線條寬度, 參見 圖 10.4.
- `lty` 設定線條類型, 參見 圖 10.4.
- `border = NULL` 設定長方形圖邊界的顏色.

```
1 > plot(c(0, 5), c(0, 100), type = "n", xlab = "", ylab = "")
2 > rect(0.0, 10, 1.2, 20) # transparent
3 > rect(2.0, 10, 4, 30, col = "green", border = "blue", lwd = 2)
4 > rect(0.5, 30, 1.5, 50, col = "red", border = "transparent")
5 > rect(2.5, 45, 3.0, 65, density = 10, border = "red", lwd = 2)
6 > rect(4.0, 50, 5.0, 80, density = 15, col = "blue", lwd = 2,
7       angle = -45, border = "black")
8 > title(main = "rect()")
9 > legend(0.0, 100, legend = 1:5,
10        fill = c("transparent", "green", "red", "transparent", "blue"),
11        density = c(NA, NA, NA, 10, 15),
12        angle = c(NA, NA, NA, 30, -45))
```

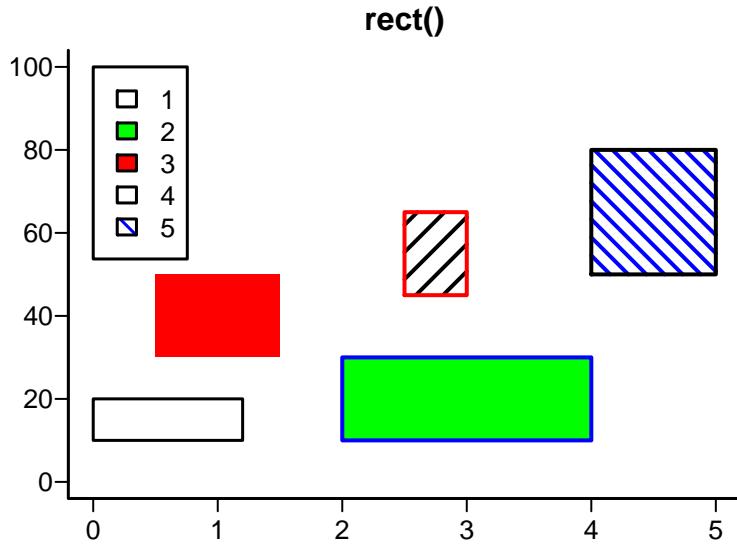


圖 10.9: 低階繪圖函式: rect()

10.13 低階繪圖函式: curve()

低階繪圖函式 `curve()` 在現有的圖形上對數學函數的繪圖繪製曲線圖形. 參見
圖 10.10.

```
1 curve(expr, from = NULL, to = NULL, n = 101, add = FALSE,
2   type = "l", xname = "x", xlab = xname, ylab = NULL,
3   log = NULL, xlim = NULL, ...)
```

可使用下列常見的引數:

- `expr` 數學函數運算式 $f(x)$.
- `from` 與 `to` 運算式 $f(x)$ 的 x 繪圖範圍.
- `n` 正整數, $f(x)$ 的評估點數目.
- `add = FALSE` 在現有圖形繪製曲線, `add = TRUE` 產生新的曲線圖形.
- `type` 設定繪點或繪線, 參見 圖 10.5.
- `xname` X-軸 的名稱.

- `log` 對 x 進行對數轉換.
- `xlab` 與 `ylab` X-軸 與 Y-軸 的名稱.

```
1 > plot(sin, -pi, 3*pi,
2       lty = 1, lwd = 2, col = "red",
3       xlab = "x (unit: pi)")
4 > curve(cos, xlim = c(-pi, 3*pi), n = 100,
5       lty = 2, lwd = 2, col = "blue", add = TRUE)
```

10.14 低階繪圖函式: rug()

低階繪圖函式 `rug()` 在現有的圖形加上資料點在坐標軸上. `rug()` 加上 “rug” (類似地毯的直線), 代表資料點. 參見 圖 10.10.

```
1 rug(x, ticksize = 0.03, side = 1, lwd = 0.5, col = par("fg"),
2      quiet =getOption("warn") < 0, ...)
```

可使用下列常見的引數:

- `x` 資料向量.
- `ticksize` rug 長度.
- `side` 坐標軸位置.
 - `side = 1`: 圖下方.
 - `side = 2`: 圖左方.
 - `side = 3`: 圖上方.
 - `side = 4`: 圖右方.

```
1 > ## rug()
2 > x <- 1:30
3 > z <- x^2
4 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
5 > plot(x, y, type = "p", pch = 16, axes = TRUE, ann = FALSE)
6 > rug(x, side = 1, ticksize = 0.03, lwd = 0.5, col = "red")
7 > rug(y, side = 2, ticksize = -0.05, lwd = 2, col = "blue")
```

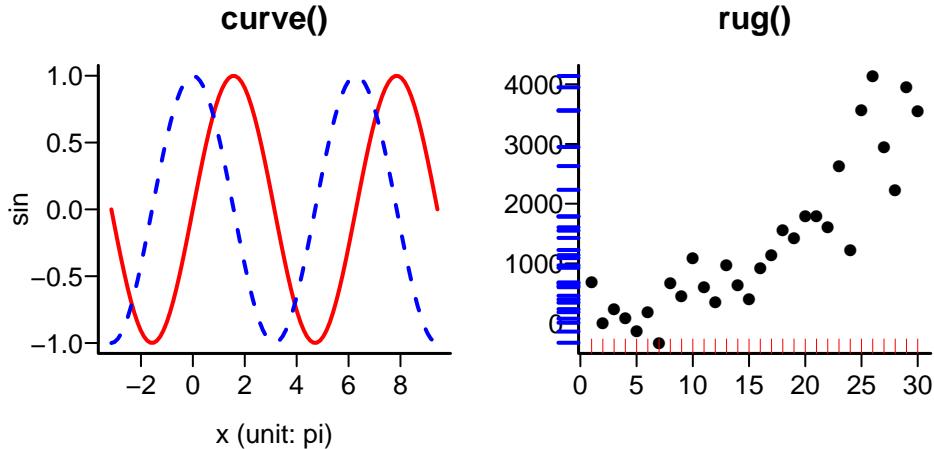


圖 10.10: 常見低階繪圖函式: curve() 與 rug()

10.15 低階繪圖函式: title()

繪圖函式 `title()` 用來在現有的圖形上加入標題。參見 圖 10.11。

```
1 title(main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
2       line = NA, outer = FALSE, ...)
```

常見引數為：

- `main = "Main Title Text"` 設定圖形主標題 (main title), 定義的文字會放在圖形的上方。
- `sub = "Subtitle Text"` 設定圖形次標題 (subtitle), 定義的文字會放在圖形的下方。
- `xlab = "X-Axis Text"` 設定圖形 X-軸 的標題。
- `ylab = "Y-Axis Text"` 設定圖形 XY-軸 的標題。
- `outer = TRUE` 設定標題在圖形的外邊緣 (outer margin)。
- `cex` 設定文字大小之放大率。
- `col` 可以設定顏色。

```

1 > ## title()
2 > set.seed(1)
3 > x <- 1:30
4 > z <- x^2
5 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
6 > plot(x, y, type = "p", pch = 16, axes = TRUE, ann = FALSE)
7 > title(main = "main title", sub = "sub-title",
8       xlab = "x-axis labels", ylab = "outocme")

```

10.16 低階繪圖函式: legend()

繪圖函式 `legend()` 用來在現有的圖形上, 在指定的 (x, y) 坐標位置繪製 圖例 (legend). 圖例的說明文字由向量 `leg.vec` 表示. 參見 圖 10.11.

```

1 legend(x, y = NULL, legend, fill = NULL, col = par("col"),
2         border = "black", lty, lwd, pch,
3         angle = 45, density = NULL, bty = "o", bg = par("bg"),
4         box.lwd = par("lwd"), box.lty = par("lty"), box.col = par("fg"),
5         pt.bg = NA, cex = 1, pt.cex = cex, pt.lwd = lwd,
6         xjust = 0, yjust = 1, x.intersp = 1, y.intersp = 1,
7         adj = c(0, 0.5), text.width = NULL, text.col = par("col"),
8         text.font = NULL, merge = do.lines && has.pch, trace = FALSE,
9         plot = TRUE, ncol = 1, horiz = FALSE, title = NULL,
10        inset = 0, xpd, title.col = text.col, title.adj = 0.5,
11        seg.len = 2)

```

常見引數為:

- `x, y` 在現有圖形的座標 (x, y) 位置上繪製 圖例 (legend). `x, y` 為數值向量.
- 僅設定單一 `x`: 可有下列常用位置可以選擇: "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center".
- `lty` 引數設定畫線之類型.
- `lwd` 引數設定畫線之寬度.
- `pch` 設定畫點之類型.
- `bty = "o"` 設定圖例外框形式 (內設預定值為加外框).

- `title = "title text"` 設定圖例上方之標題.
- `merge = TRUE` 設定圖例同時有點與線時, 合併點與線.
- `cex` 設定文字大小之放大率.
- `col` 可以設定顏色.
- `font` 可以設定字型.

另外尚有一些引數, 參見輔助文件.

```

1 > ## legend()
2 > par(mfrow = c(1, 1), oma = c(0, 0, 0, 0), mar = c(5, 4, 3, 1))
3 > set.seed(1)
4 > x <- 1:30
5 > z <- x^2
6 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
7 > x2 <- x + rnorm(n = 30, mean = 0, sd = 1)
8 > y2 <- y + 1000 + rnorm(n = 30, mean = 0, sd = 1)
9 > plot(x, y, type = "b",
10        pch = 16, lty = 1, lwd = 1, col = "black",
11        main = "legend()")
12 > points(x2, y2, type = "b",
13            pch = 1, lty = 2, lwd = 2, col = "red")
14 > legend(6, 4000, legend = 1:2, title = "group",
15           pch = c(16, 1), lty = c(1, 2), lwd = c(1, 2),
16           col = c("black", "red"))

```

10.17 低階繪圖函式: mtext()

低階繪圖函式 `mtext()` 在現有的圖形之 邊緣 (margin), 加上文字. 參見 圖 10.12.

```

1 mtext(text, side = 3, line = 0, outer = FALSE, at = NA,
2       adj = NA, padj = NA, cex = NA, col = NA, font = NA, ...)

```

常見的引數:

- `text = "words"` 加入所要呈現的文字.
- `side = k`: $k \in \{1, 2, 3, 4\}$, 從圖型底部為 `side = 1`, 逆時鐘旋轉.

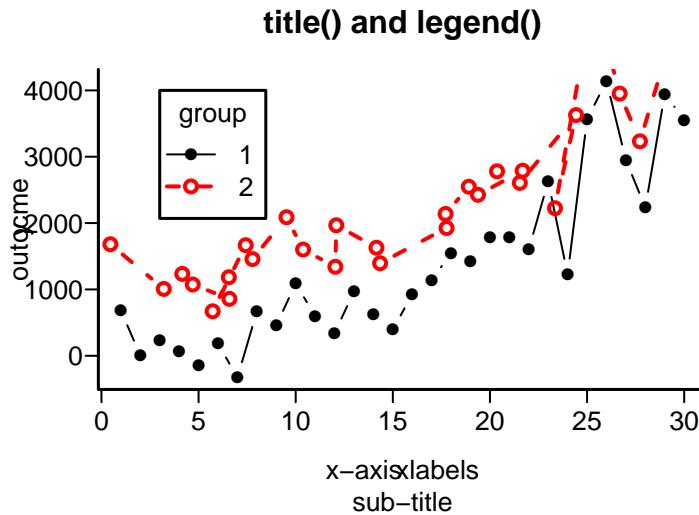


圖 10.11: 低階繪圖函式: title() 與 legend()

- `line = k` 設定從邊緣向外第 k 行, $k = 0, 1, 2, \dots$, 開始呈現文字.
- `outer = TRUE`: 若圖形有外邊緣 (outer margin), 則在外邊緣加入所要呈現的文字.
- `adj` 設定呈現文字的閱讀方向, 若文字的方向與坐標軸平行, 設定 `adj = 0`, 則文字從左下方起始, 設定 `adj = 1`, 則文字從右上方起始.
- `padj` 設定呈現文字的方向, 若文字垂直於閱讀方向 (由引數 `adj` 設定), 且文字的方向與坐標軸平行, 設定 `padj = 0`, 則文字從右上方起始, 設定 `padj = 1`, 則文字從左下方起始.
- `cex` 設定文字大小之放大率.
- `col` 可以設定顏色.
- `font` 可以設定字型.

例如,

```
1 mtext(text, side = 3, line = 0)
```

在現有的圖形之邊緣 (margin), 加上文字 "words".

```

1 > ## mtext()
2 > par(mfrow = c(1,1), oma = c(2, 2, 2, 2), mar = c(5, 5, 5, 5))
3 >
4 > x <- runif(50, 0, 2)
5 > y <- runif(50, 0, 2)
6 > plot(x, y, bty = "o", type = "p", pch = ""),
7 +   main = "main title", sub = "sub-title",
8 +   xlab = "x-lab", ylab = "y-lab")
9 > box(which = "figure", lty = 2, lwd = 2.0, bty = "o")
10 > text(1, 1, "text at (1,1)")
11 > mtext(paste("side", 1:4), side = 1:4, line = -1, font = 2)
12 > mtext(-1:5, side = 1, at = 0.3, line = -1:5)
13 > mtext("mtext()", side = 3, at = 1, line = 4)
14 > mtext("red mtext", side = 2, at = 1, line = 2, col = "red")
15 > mtext("AB", side = 4, at = 0.3, line = c(-1, 1, 3, 5))
16 >
17 > par(mfrow = c(1,1))

```

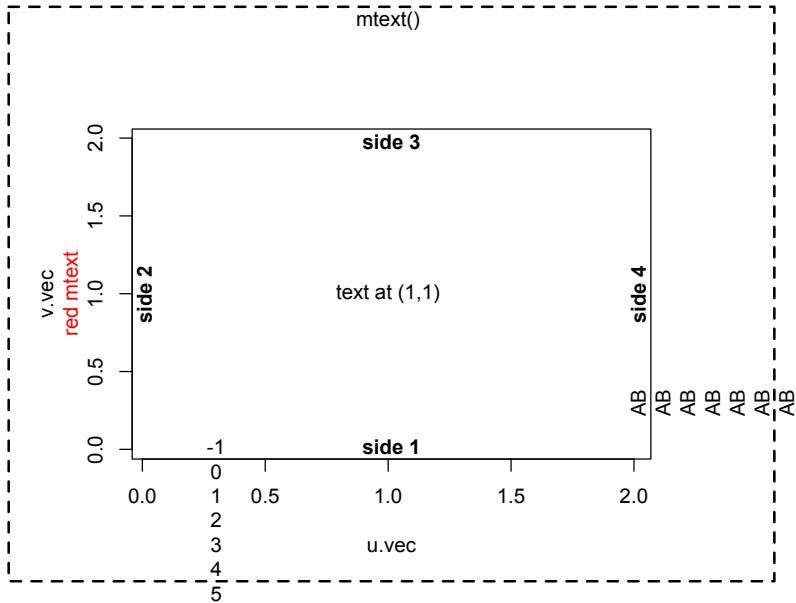


圖 10.12: 低階繪圖函式: mtext()

10.18 低階繪圖函式: `text()`

低階繪圖函式 `text(x, y, labels = char.vec, ...)` 在在現有的圖形的 `(x, y)` 向量坐標點上, 標出由 `labels` 所設定之對應數值向量或文字向量. 參見 圖 10.13.

```
1 text(x, y = NULL, labels = seq_along(x), adj = NULL,
2   pos = NULL, offset = 0.5, vfont = NULL,
3   cex = 1, col = NULL, font = NULL, ...)
```

常見引數為:

- `x, y` 在現有圖形的座標 `(x, y)` 位置上加文字, `x, y` 為數值向量.
- 在 `x[i], y[i]` 坐標點標出 `labels[i]` 之數值或文字.
- 通常在使用 `plot(x, y, type = "n")` 設立坐標軸, 但不畫出點或線, 而由 `text(x, y, z.vec)` 標出數值或文字.
- `col` 可以設定顏色.
- `adj = 0.5, adj = -0.1, ..., etc.`

設定調整文本 (`text`) 相對於圖形的之相對位置, 內定預設值為 0.5. 0 表示左對齊, 1 表示右對齊, 0.5 表示圖形位置的水準居中, 設定實際值表示出現在圖形位置左側的文本字體寬度的比率, 因此 `-0.1` 表示會在文本和圖形位置間留下文本字體寬度的 10%.

```
1 > x <- 1:30
2 > z <- x^2
3 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
4 > plot(x, y, type = "n", axes = TRUE, ann = FALSE)
5 > text(x, y, labels = as.character(c(1:30)), cex = 0.8)
6 > title(main = "text")
7 > ## text()
8 > plot.new()
9 > plot.window(xlim = c(0, 1), ylim = c(0, 1))
10 > abline(h = c(.2, .5, .8),
11 +           v = c(.5, .2, .8), col = "gray")
12 > points(0.5, 0.5, pch = 16)
```

```

13 > points(0.5, 0.8, pch = 16)
14 > points(0.5, 0.2, pch = 16)
15 > points(0.2, 0.5, pch = 16)
16 > points(0.8, 0.5, pch = 16)
17 > text(0.5, 0.5,
18     "srt = 45, adj = c(0.5, 0.5)",
19     srt = 45, adj = c(0.5, 0.5))
20 > text(0.5, 0.8, "adj = c(0, 0.5)", adj = c(0, 0.5))
21 > text(0.5, 0.2, "adj = c(1, 0.5)", adj = c(1, 0.5))
22 > text(0.2, 0.5, "adj = c(1, 1)", adj = c(1, 1))
23 > text(0.8, 0.5, "adj = c(0, 0)", adj = c(0, 0))
24 > axis(1)
25 > axis(2, las = 1)
26 > title(main = "text()")

```

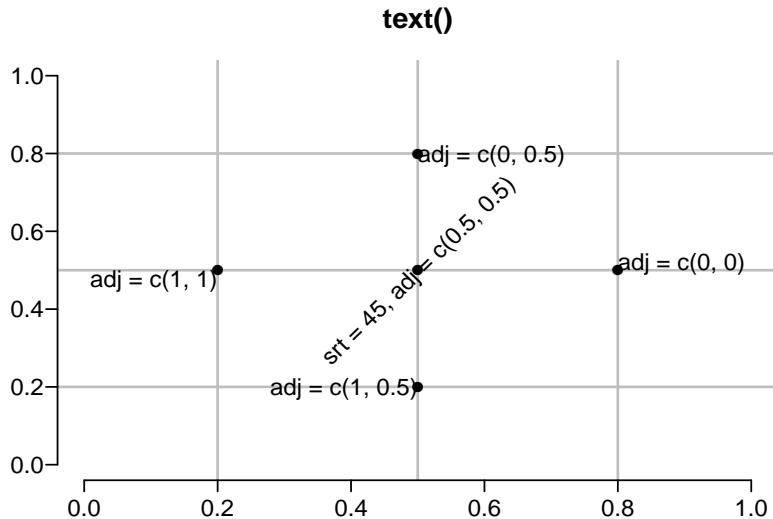


圖 10.13: 低階繪圖函式: text()

10.19 低階繪圖函式: box()

低階繪圖函式 `box()` 在現有的圖形上加上盒狀圖框。參見 圖 10.14。

```
1 box(which = "plot", lty = "solid", ...)
```

可使用下列常見的引數:

- **lty** 設定線條類型. 參見 圖 10.4.
- **box("figure")** 在全圖區域繪製圖框, 全圖區域 (figure region) = plot region + margin.

```
1 > ## box()
2 > oldpar <- par(no.readonly = TRUE)
3 > par(mfrow = c(1,1), mar = c(4, 3, 3, 2), oma = c(1, 1, 1, 1))
4 > set.seed(1)
5 > x <- 1:30
6 > z <- x^2
7 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
8 > plot(x, y, type = "n", axes = TRUE, ann = FALSE, bty = "n")
9 > box(which = "plot", lty = 1, lwd = 2.0, col = "black", bty = "l")
10 > box(which = "figure", lty = 2, lwd = 2.0, col = "gray", bty = "o")
11 > title(main = "box")
12 >
13 > par(oldpar)
```

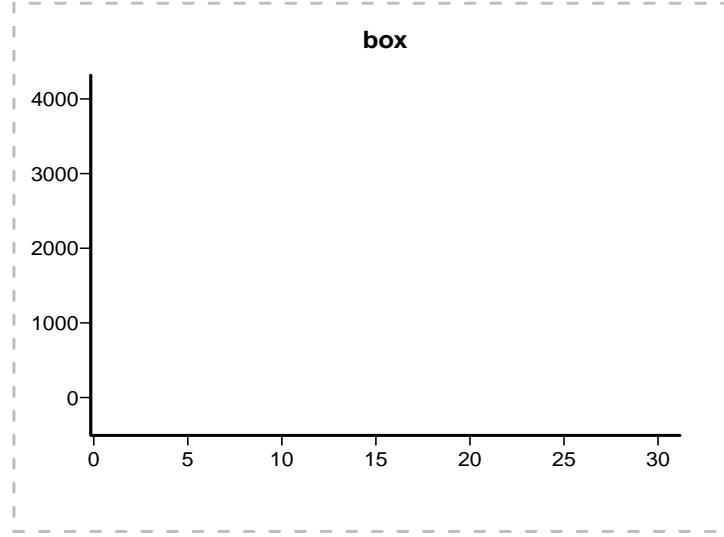


圖 10.14: 低階繪圖函式

10.20 座標軸和座標刻度設定函式: axis()

Axes and Tick Marks: axis()

R 的高階繪圖函式內設 坐標軸 (axis), 可以用低階繪圖函式 `axis()` 設置使用者的坐標軸, 參見 表 10.4 與 圖 ?? 與 圖 10.16.

```
1 axis(side, at = NULL, labels = TRUE, tick = TRUE, line = NA,
2   pos = NA, outer = FALSE, font = NA, lty = "solid",
3   lwd = 1, lwd.ticks = lwd, col = NULL, col.ticks = NULL,
4   hadj = NA, padj = NA, ...)
```

坐標軸包括 3 個主要部分:

1. 坐標軸線 (axis line): 線條格式由圖形引數 `lty`, `lwd` 等設定.
2. 坐標軸刻度 (tick mark): 設定如何劃分坐標軸線上的刻度.
3. 坐標軸刻度標記 (tick label): 設定坐標軸線刻度上的單位.

可使用下列常見的引數:

- `side` 設定座標軸的位置.
 - `side = 1`: 圖下方 X-軸.
 - `side = 2`: 圖左方 Y-軸.
 - `side = 3`: 圖上方 X-軸.
 - `side = 4`: 圖右方 Y-軸.
- `at` 設定坐標軸的刻度的位置的向量.
- `lab = TRUE` 設定坐標軸為數值刻度. 若要自行設定坐標軸的數值或文字, 則輸入與 `at` 位置 同樣長度的數值或文字向量.
- `tick = TRUE` 繪製坐標軸的軸線與刻度.

- `line` 繪製坐標軸的軸線與邊緣的距離, 以文字行數計算.
- `pos` 設定坐標軸的軸線所在數值位置.
- `outer = FALSE` 設定坐標軸的軸線不在圖形的外邊緣.
- `font.axis` 設定坐標軸標註的字型, 字形: 1, 2, 3, 4 = "normal", "bold", "italic", "bold-italic" 等.
- `lty` 設定線條類型, 參見 圖 10.4.
- `lwd, lwd.ticks` 設定軸線線條與刻度的寬度, 參見 圖 ??.
- `col, col.axis, col.ticks` 設定軸線線條, 標註與刻度的顏色.
- `hadj` 設定標註的數值或文字為閱讀方向 (圖形的水平方向).
- `padj` 設定標註的刻度與閱讀方向垂直.
- `las` 設定坐標軸刻度標記的方向, 內定預設值為 0 表示總是平行於坐標軸, 1 表示平行於坐標軸, 2 表示垂直於坐標軸, 3 表示總是垂直於坐標軸, 例如, `las = 0, las = 1, ..., etc..`

其他在高階繪圖函示 或 圖形控制參數環境 `par()` 中常見控制座標軸的繪圖引數參見

- `bty` 設定圖形座標軸外框 (box) 的類型, 選項共有 "`o`", "`l`", "`7`", "`c`", "`u`", 或 "`]`", 座標軸外框的類型類似所選之文字, 例如, `bty = "o"`, `bty = "l"`, ..., `etc..`
- `lab` 前兩個引數分別設定 `x` 和 `y` 坐標軸的刻度間隔數目 (numbers of tick intervals). 第 3 個引數刻度是坐標軸刻度標記的字體長度 (包括小數點); 第 3 個引數若設定太小, 導致所有的坐標軸刻度標記變成一樣的數位. 內定預設值為 `lab = c(5, 5, 7)`.
- `las` 設定坐標軸刻度標記的方向, 內定預設值為 0 表示總是平行於坐標軸, 1 表示平行於坐標軸, 2 表示垂直於坐標軸, 3 表示總是垂直於坐標軸, `las = 0`.

- `mgp` 設定三個座標成分, 坐標軸標籤 (axis label), 坐標軸刻度標記 (tick mark, tick label), 坐標軸 (axis line) 等到軸線的相對距離位置. 第 1 個引數是 坐標軸標籤 相對 坐標軸位置的距離; 第 2 個引數表示 坐標軸刻度標記 相對 坐標軸位置的距離; 第 3 個引數是 坐標軸位置 到軸線的距離, 正值表示在圖形外, 負值表示在圖形內. 距離大小以文本 (text) 文字的行數作為參考單位. 內定預設值為 $mgp = c(3, 1, 0)$.
- `tck` 設定坐標軸刻度的長度, 以畫圖區域大小的比率作為度量. 當 `tck` 比較小 (小於 0.5), `x` 和 `y` 坐標軸上的刻度強制大小相等. 設定值為 1 時, 繪出格線. 負值時刻度在圖形外. 例如, `tck = 0.01` 和 `mgp = c(1, -1.5, 0)` 表示內部刻度. 內定預設值為 `NA`.
- `xaxs` 分別設定 `x` 和 `y` 坐標軸的形式. "`i`" 表示內在的, "`r`" 表示內定預設值, 二者刻度都包含繪圖資料的範圍. 但是 "`r`" 形式的坐標軸刻度會在坐標軸刻度範圍兩邊留一些空隙.

```

1 > par(mfrow = c(1, 2), oma = c(0, 0, 0, 0), mar = c(5, 4, 3, 1))
2 > ## axis()
3 > set.seed(1)
4 > x <- 1:30
5 > z <- x^2
6 > y <- (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))
7 > ##
8 > plot(x, y, type = "b",
9       pch = 16, lty = 1, lwd = 1, col = "black",
10      main = "Dose Response: plot()", sub = "",
11      xlab = "dosage", ylab = "concentration")
12 > ##
13 > plot(x, y, type = "b",
14       pch = 16, lty = 1, lwd = 1, col = "black",
15       ann = FALSE, axes = FALSE)
16 > axis(1, at = c(0, 10, 20, 30),
17       labels = c("0mg", "10mg", "20mg", "30mg"),
18       las = 1, col = "black", col.ticks = "red", lwd = 2, font.axis = 2,
19       tck = -0.05, mgp = c(3, 1, 0), cex = 1)
20 > axis(2, at = c(0, 1000, 2000, 3000, 4000),
21       labels = c("0", "1000", "2000", "3000", "4000"),
22       las = 3, col = "black", col.ticks = "gray", lwd = 2,
23       tck = 0.05, mgp = c(3, 0.5, 0.5), cex = 0.7)
24 > title(main = "Dose Response axis()", sub = "",
```

```
25      xlab = "dosage", ylab = "concentration")  
26 > par(mfrow = c(1,1))
```

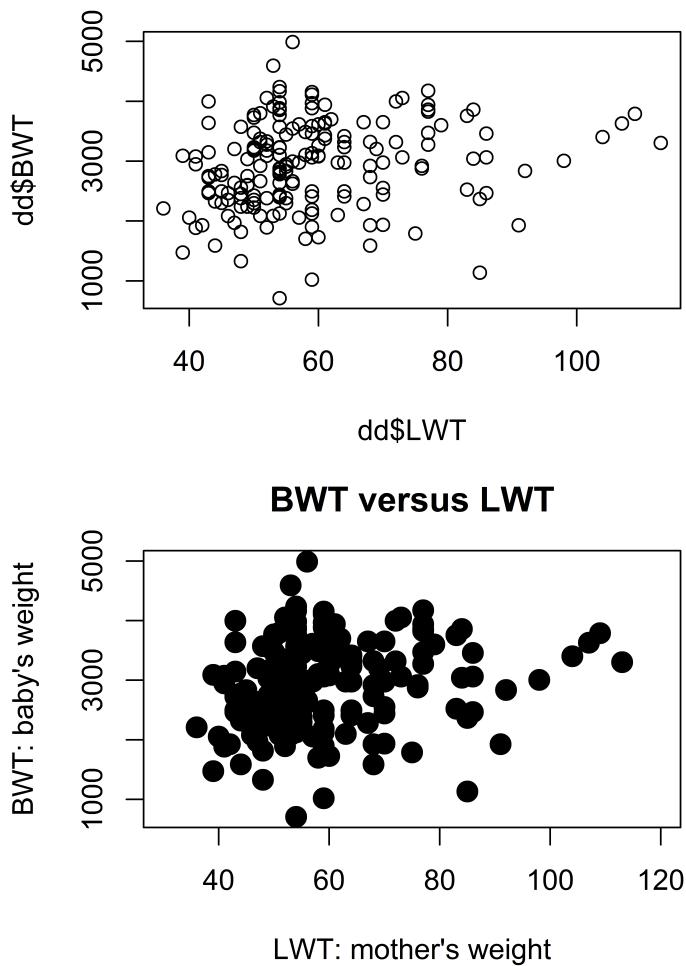


圖 10.15: 高階繪圖函式引數設定之應用

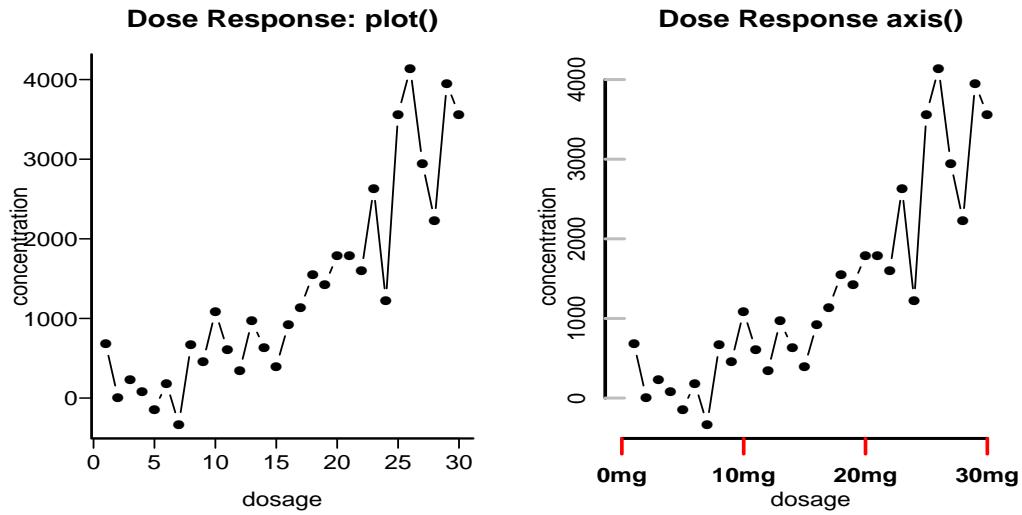


圖 10.16: 低階繪圖函式: axis()

10.21 數學文字與符號 Mathematical Annotation

使用者有時須在現有圖形中，加上數學文字與符號或公式。函式 `expression()` 內使用類似 \LaTeX 數學指令，(非一般文字)，配合 `mtex()`, `axis()`, 或 `title()`，可以加上數學文字與符號或公式，參見 表 10.5 – 表 10.9。例如在二項機率函數加上數學符號：

```
1 text(x, y, expression(paste(bgroup("(", atop(n, x), ")"), p^x, q^{n-x})))
```

更詳盡的內容，參見輔助文件。

```
1 > help(plotmath)
2 > example(plotmath)
3 > demo(plotmath)
```

表 10.4: 常見圖形引數中之座標軸和座標刻度設定

Axes:	
<code>xlim, ylim</code>	<code>xlim = c(0, 25)</code> 座標軸範圍的上下界.
<code>bty</code>	<code>bty = "o",</code> 設定圖形座標軸外框 (box) 的類型, 選項共有 <code>{ "o", "l", "7", "c", "u", "]", "n" }.</code> <code>"n"</code> 則不呈現座標軸外框的外框.
<code>lab</code>	<code>lab = c(3, 7, 4)</code> 坐標軸刻度標記長度與數目.
<code>las</code>	<code>las = 2</code> 坐標軸刻度標記與軸線的相對方向. 0 = 平 形軸線, 1 = 水平, 2 = 垂直.
<code>log</code>	<code>log = "y"</code> 坐標軸使用對數值.
<code>mgp</code>	<code>mgp = c(3, 1, 0)</code> 坐標軸標籤, 坐標軸刻度標記, 坐標軸 等到 軸線的相對距離位置.
<code>tck</code>	<code>tck = -0.01</code> 設定坐標軸刻度的長度.
<code>xaxp, yaxp</code>	<code>xaxp = c(2, 10, 4)</code> 坐標軸刻度標記的極限與數目.
<code>xaxs, yaxs</code>	<code>xaxs = "i"</code> 設定坐標軸的形式.
<code>xaxt, yaxt</code>	<code>xaxt = "s"</code> 坐標軸形式, "n" (無), "s" (標準), "t" (時間) or "l" (對數).

表 10.5: 常見繪圖用數學符號-I

Arithmetic Operation

<code>x + y</code>	$x + y$	x plus y
<code>x - y</code>	$x - y$	x minus y
<code>x * y</code>	xy	juxtapose x and y
<code>x / y</code>	x/y	x forwardslash y
<code>x %+-% y</code>	$x \pm y$	x plus or minus y
<code>x %/% y</code>	$x \div y$	x divided by y
<code>x %*% y</code>	$x \times y$	x times y
<code>- x</code>	$-x$	minus x
<code>+ y</code>	$+y$	plus y

Sub/Subscript

<code>x[i]</code>	x_i	x subscript i
<code>x^2</code>	x^2	x superscript 2

Juxtaposition

<code>x*y</code>	xy	juxtapose x and y
<code>paste(x, y, z)</code>	xyz	juxtapose x, y, and z

Radical

<code>sqrt(x)</code>	\sqrt{x}	square root of x
<code>sqrt(x,y)</code>	$\sqrt[y]{x}$	yth root of x

表 10.6: 常見繪圖用數學符號-II

Relation

<code>x == y</code>	$x = y$	x equals y
<code>x != y</code>	$x \neq y$	x is not equal to y
<code>x < y</code>	$x < y$	x is less than y
<code>x <= y</code>	$x \leq y$	x is less than or equal to y
<code>x > y</code>	$x > y$	x is greater than y
<code>x >= y</code>	$x \geq y$	x is greater than or equal to y
<code>x %~% y</code>	$x \approx y$	x is approximately equal to y
<code>x %~% y</code>	$x \cong y$	x and y are congruent
<code>x %==% y</code>	$x \equiv y$	x is defined as y
<code>x %prop% y</code>	$x \propto y$	x is proportional to y

Typeface

<code>plain(x)</code>	x	draw x in normal font
<code>bold(x)</code>	\mathbf{x}	draw x in bold font
<code>italic(x)</code>	x	draw x in italic font
<code>bolditalic(x)</code>	$\mathbf{\mathit{x}}$	draw x in bolditalic font

List

<code>list(x, y, z)</code>	x, y, z	comma-separated list
<code>...</code>	\dots	ellipsis (height varies)
<code>cdots</code>	\cdots	ellipsis (vertically centred)
<code>ldots</code>	\dots	ellipsis (at baseline)

表 10.7: 常見繪圖用數學符號-III

Set Relation

<code>x %subset% y</code>	$x \subset y$	x is a proper subset of y
<code>x %subeteq% y</code>	$x \subseteq y$	x is a subset of y
<code>x %notsubset% y</code>	$x \not\subset y$	x is not a subset of y
<code>x %supset% y</code>	$x \supset y$	x is a proper superset of y
<code>x %supeteq% y</code>	$x \supseteq y$	x is a superset of y
<code>x %in% y</code>	$x \in y$	x is an element of y
<code>x %notin% y</code>	$x \notin y$	x is not an element of y

Accents

<code>hat(x)</code>	\hat{x}	x with a circumflex
<code>tilde(x)</code>	\tilde{x}	x with a tilde
<code>dot(x)</code>	\dot{x}	x with a dot
<code>ring(A)</code>	\mathring{A}	x with a ring
<code>bar(xy)</code>	\bar{xy}	xy with bar
<code>widehat(xy)</code>	\widehat{xy}	xy with a wide circumflex
<code>widetilde(xy)</code>	\widetilde{xy}	xy with a wide tilde
<code>underline(x)</code>	\underline{x}	draw x underlined

Spacing

`x ~~ y` $x \quad y$

表 10.8: 常見繪圖用數學符號-IV

Arrows		
<code>x % <- >% y</code>	$x \leftrightarrow y$	x double-arrow y
<code>x %->% y</code>	$x \rightarrow y$	x right-arrow y
<code>x % <- % y</code>	$x \leftarrow y$	x left-arrow y
<code>x %up% y</code>	$x \uparrow y$	x up-arrow y
<code>x %down% y</code>	$x \downarrow y$	x down-arrow y
<code>x %< = >% y</code>	$x \Leftrightarrow y$	x is equivalent to y
<code>x % = >% y</code>	$x \Rightarrow y$	x implies y
<code>x %< = % y</code>	$x \Leftarrow y$	y implies x
<code>x %dblup% y</code>	$x \upuparrow y$	x double-up-arrow y
<code>x %dbldown% y</code>	$x \downdownarrow y$	x double-down-arrow y
<hr/>		
Symbolic Names		
<code>alpha - omega</code>	$\alpha - \omega$	Greek symbols
<code>'Sigma' - 'Omega'</code>	$\Sigma - \Omega$	uppercase Greek symbols
<code>theta1, phi1, sigma1, omega1</code>	$\theta, \varphi, \varsigma, \varpi$	cursive Greek symbols
<code>Upsilon1</code>	Υ	capital upsilon with hook
<code>infinity</code>	∞	infinity symbol
<code>partialdiff</code>	∂	partial differential symbol
<code>32*degree</code>	32°	32 degrees
<code>60*minute</code>	$60'$	60 minutes of angle
<code>30*second</code>	$30''$	30 seconds of angle
<hr/>		
Style		
<code>displaystyle(x)</code>	x	draw x in normal size (extra spacing)
<code>textstyle(x)</code>	x	draw x in normal size
<code>scriptstyle(x)</code>	x	draw x in small size
<code>scriptscriptstyle(x)</code>	x	draw x in very small size
<hr/> <hr/>		

表 10.9: 常見繪圖用數學符號- \vee

Functions

<code>x + phantom(0) + y</code>	$x + + y$	leave gap for "0", but don't draw it
<code>x + over(1, phantom(0))</code>	$x + \overline{1}$	leave vertical gap for "0", don't draw
<code>frac(x, y)</code>	$\frac{x}{y}$	x over y
<code>over(x, y)</code>	$\frac{x}{y}$	x over y
<code>atop(x, y)</code>	$\frac{x}{y}$	x over y (no horizontal bar)
<code>sum(x[i], i == 1, n)</code>	$\sum_{i=1}^n x_i$	sum $x[i]$ for i equals 1 to n
<code>prod(plain(P)(X == x), x)</code>	$\prod_x P(X = x)$	product of $P(X = x)$ for all values of x
<code>integral(f(x)*dx, a, b)</code>	$\int_a^b f(x)dx$	definite integral of $f(x)$ wrt x
<code>union(A[i], i == 1, n)</code>	$\bigcup_{i=1}^n A_i$	union of $A[i]$ for i equals 1 to n
<code>intersect(A[i], i == 1, n)</code>	$\bigcap_{i=1}^n A_i$	intersection of $A[i]$
<code>lim(f(x), x %->% 0)</code>	$\lim_{x \rightarrow 0} f(x)$	limit of $f(x)$ as x tends to 0
<code>min(g(x), x > 0)</code>	$\min_{x>0} g(x)$	min. of $g(x)$ for x greater than 0
<code>inf(S)</code>	$\inf S$	infimum of S
<code>sup(S)</code>	$\sup S$	supremum of S
<code>x^y + z</code>	$x^y + z$	normal operator precedence
<code>x^(y + z)</code>	$x^{(y+z)}$	visible grouping of operands
<code>x^{y + z}</code>	x^{y+z}	invisible grouping of operands
<code>group("(",list(a, b),"")")</code>	(a, b)	specify left and right delimiters
<code>bgroup("(",atop(x,y),"")")</code>	$\left(\frac{x}{y}\right)$	use scalable delimiters
<code>group(lceil, x, rceil)</code>	$\lceil x \rceil$	special delimiters
<code>group(lfloor, x, rfloor)</code>	$\lfloor x \rfloor$	special delimiters
<code>group(, x,)</code>	$ x $	special delimiters

10.22 多重圖片環境

Multiple Figure Environment

R 容許在一個單頁上建立一個 $n \times m$ 圖片矩陣。每一個圖片有自己的邊緣，圖片陣列有外邊緣 (outer margin) 包圍，參見 圖 10.17。常見多重圖片相關的圖形引數如下：

- `mfrow = c(1,1), mfrow = c(2,2), mfcoll = c(1, 1), ..., etc.`

引數 `mfrow` 與 `mcol` 都可以設定多重圖片陣列數目。第一個引數值是列數 (row)，第二個值是欄數，差異在於 `mfcoll(n,m)` 圖片以“欄”位為主 (by column) 的方式放置，`mfrow(n,m)` 則以“列”位為主 (by row) 的方式放置，圖 10.17 中版面設計以 `mfrow = c(3,2)` 設定。

```
1 > par(oma = rep(3, 4), mfrow = c(3,2), bg = "white")
2 > .....
3 > plot() .....
4 > .....
5 par(mfrow = c(1,1))
```

- `layout = x.matrix, layout.show(n)`

引數 `layout = x.matrix` 可將圖形裝置 (device) 分割成為 `x.matrix` 型式，`layout.show(n)` 顯示分割情形。可以用引數 `widths` 與 `heights` 設定寬與高，參見 圖 10.18 與 圖 10.19。

- `mfq = c(2, 2, 3, 2), ..., etc.`

引數 `mfq` 將會調整現有的圖片 (current figure) 在一個多重圖片環境中的位置。前兩個數字是現有圖片的 列 (row) 與 欄 (column) 編號，後面兩個數字是多重圖片陣列的列與欄編號。

- `fig = c(4, 9, 1, 4)/10, ..., etc.`

引數 `fig` 設定現有的圖片在頁面上的位置。設定值分別表示左側，右側，下側

和上側的邊緣寬度，並且以左側底部作為參照點，得到的頁面之百分比。

- `oma = c(0, 0, 0, 0), oma = c(2, 0, 3, 0), omi = c(0, 0, 0.8, 0), ... , etc.`

R 內部對圖片沒有設定外邊緣。須使用引數 `oma` 或 `omi` 設定外邊緣 (outer margin) 的大小，類似引數 `mar()` 和 `mai` 設定內邊緣的作用，第一個以文本文字行數度量，第二個以英寸度量。從底部邊緣算起，以順時鐘方向設置。文字可以通過函式 `mtext()` 和引數 `outer = TRUE` 加在外邊緣。

```
1 > # a
2 > layout(matrix(1:4, 2, 2))
3 > layout.show(4)
4 >
5 > # b
6 > layout(matrix(1:6, 3, 2, byrow = TRUE))
7 > layout.show(6)
8 >
9 > # c
10 > layout(matrix(c(1,2,3,3), 2, 2, byrow = TRUE))
11 > layout.show(3)
12 >
13 > # d
14 > layout(matrix(1:4, 2, 2, byrow = TRUE), widths = c(3,1), heights = c(1,3))
15 > layout.show(4)
16 >
17 > # e
18 > layout(matrix(c(1,1,2,1), 2, 2), widths = c(2,1), heights = c(1,2))
19 > layout.show(2)
20 >
21 > # f
22 > layout(matrix(c(0,1,2,3), 2, 2), widths = c(1,3), heights = c(1,3))
23 > layout.show(3)
24 >
25 > # g
26 > ## create single figure which is 5cm square
27 > layout(matrix(1), widths = lcm(5), heights = lcm(5))
28 >
29 > # h
30 > setps(RGraphAnatomyLayout08, h = 5, w = 5)
31 > # setpdf(RGraphAnatomyLayout08, h = 5, w = 5)
32 > par(mfrow = c(1,1))
33 >
34 > ##-- Create a scatterplot with marginal histograms ----
35 > def.par <- par(no.readonly = TRUE) # save default, for resetting...
36 >
```

```
37 >     x <- pmin(3, pmax(-3, rnorm(50)))
38 >     y <- pmin(3, pmax(-3, rnorm(50)))
39 >     xhist <- hist(x, breaks = seq(-3,3,0.5), plot = FALSE)
40 >     yhist <- hist(y, breaks = seq(-3,3,0.5), plot = FALSE)
41 >     top <- max(c(xhist$counts, yhist$counts))
42 >     xrange <- c(-3,3)
43 >     yrange <- c(-3,3)
44 >     nf <- layout(matrix(c(2,0,1,3),2,2,byrow = TRUE),
45 >                   c(3,1), c(1,3), TRUE)
46 >     layout.show(nf)
47 >
48 >     par(mar = c(3,3,1,1))
49 >     plot(x, y, xlim = xrange, ylim = yrange, xlab = "", ylab = "")
50 >     par(mar = c(0,3,1,1))
51 >     barplot(xhist$counts, axes = FALSE, ylim = c(0, top), space = 0)
52 >     par(mar = c(3,0,1,1))
53 >     barplot(yhist$counts, axes = FALSE, xlim = c(0, top),
54 >               space = 0, horiz = TRUE)
```

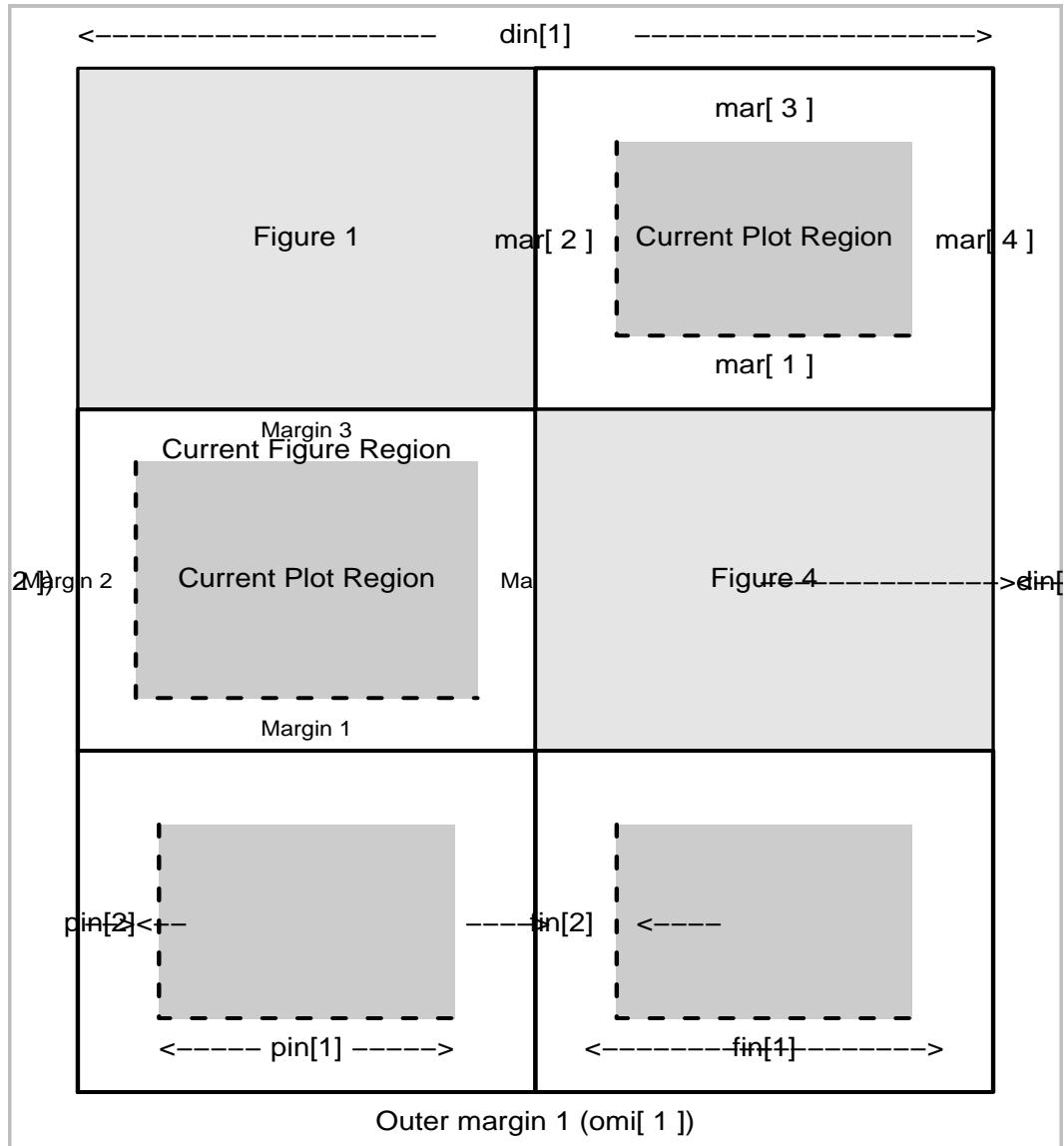


圖 10.17: R 多重圖形與圖畫邊緣解剖

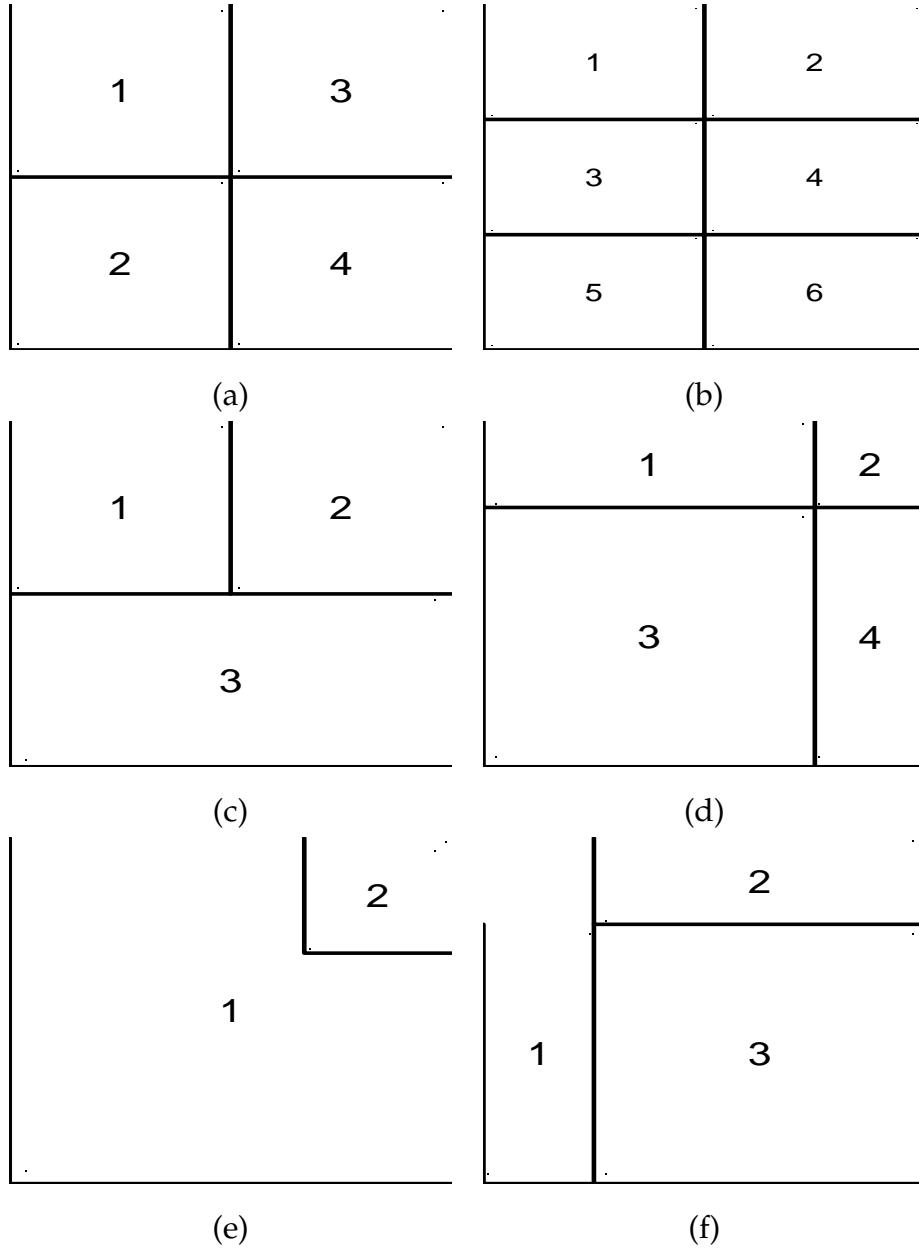


圖 10.18: 顯示多重圖形

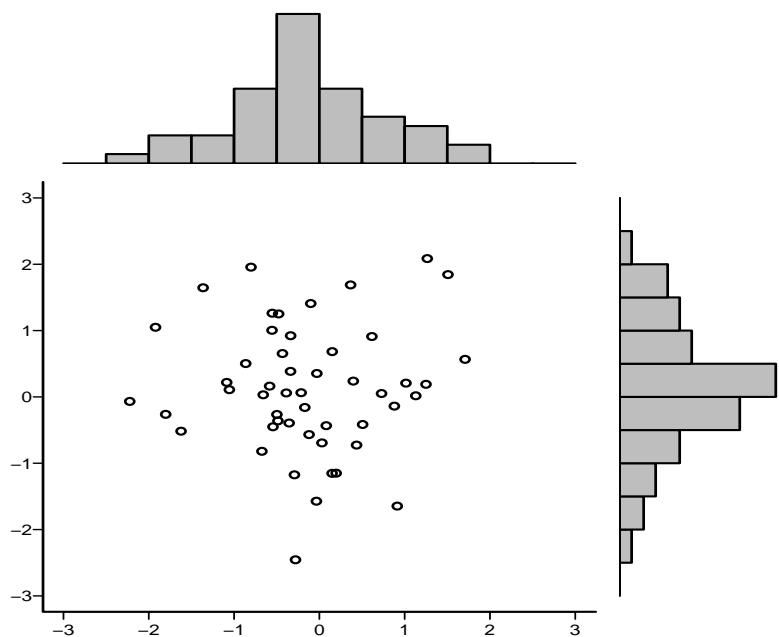


圖 10.19: R 多重圖形與圖畫邊緣使用