
第 6 章: 常用的 R 繪圖程式

6: R Graphics

R 內建許多圖形工具函式, 這些圖形工具可以顯示各種統計繪圖並且自建一些全新的圖. 以下指令, 可以先看看 R 的圖形示範.

```
demo(graphics)
demo(image)
demo(lattice)
```

R 圖形工具函式之特色是用同一個繪圖函式, 對不同的類型物件, 可以作出不同的圖形. 圖形工具函式既可互動式使用, 也可以批次處理使用. 在許多情況下, 互動式使用是最有效的. 圖形工具函式所產生之圖形結果, 無法指定成物件, 必須送到 **圖形裝置 (graphic device)**, 圖形裝置可以是一個視窗或某一特定格式之圖形檔案. R 有一系列 **圖形引數 (graphical parameters)**, 這些圖形引數可以修改或制定所需的圖形環境. 繪圖指令可以分成了三個基本的類型:

1. **高階繪圖 (high-level plotting functions)**: 指令在圖形裝置上產生一個新的圖區, 它可以包括坐標軸, 標籤, 標題等等.
2. **低階繪圖 (low-level plotting functions)**: 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.
3. **互動式繪圖 (interactive graphics functions)**: 指令允許互動式地用其他設備 (如滑鼠) 在一個已經存在的繪圖上, 加上圖形資訊或者萃取圖形中的資訊.

在 R 中, **grid** 套件有一個獨立的強大之圖形系統, 根據 **grid** 套件, 所建構的另一個較容易使用之 **lattice** 套件, 可以產生多重漂亮起專業的統計繪圖, 類似 S-PLUS 裏的 **Trellis** 系統.

6.1 繪圖簡介

在此先示範一些常見的 R 繪圖函式與使用方法, 如圖 6.1. 許多時候, 開始用 R 繪圖函式會有一些困惑, 但這些困惑通常會很快消失. 注意: 統計繪圖是一專業學科, 好的統計繪圖是要多次嘗試才能得到較好得結果, 可參考些基本工具書, 如 Cleveland (1993, 1994), Tufte (1990, 1997), Wainer (1984, 1994, 1996)

```

> x.norm<-rnorm(100,mean=0,sd=1)
> x.group<-sample(c(1:4), 100, replace=TRUE)
> x.cat<-sample(c(1:8), 100, replace=TRUE, prob=c(8:1))
> x.cat.tab<-table(x.cat)

> barplot(x.cat.tab)
> pie(x.cat.tab)
> hist(x.norm)
> boxplot(x.norm)
> qqnorm(x.norm)
> plot(density(x.norm))

```

以上圖形指令, 執行結果如圖 6.1.

例 1: NIDDM 臨床試驗比較合成胰島素與一般胰島素對降低血糖和總膽固醇的效應

一位研究者進行一個臨床試驗, 針對成人第二型糖尿病 (NIDDM), 在同時控制其他的因素, 包括性別和發生糖尿病之年齡, 之後, 比較 2 種不同的治療: 新的治療藥物為“合成胰島素 (Syn. Insulin)”與“一般胰島素 (Reg. Insulin)”, 對降低血糖和總膽固醇的效應. 總共 32 位病患, 以性別和發生糖尿病之年齡分成 4 個階層, 隨機地分配到合成胰島素治療組與一般胰島素控制組 (Reg. Insulin), 性別分層為男與女, 發生糖尿病之年齡分層為 20-39 歲和 40-59 歲. 測量基線 (baseline) 之空腹血糖和總膽固醇 (FBS1, TC1), 與測量治療 12 個月之後的空腹血糖和總膽固醇 (FBS2, TC2). 研究的第 1 個目地是比較 2 種不同的治療對降低血糖和總膽固醇的效應, 是否有任何差異. 第 2 個目地是評估性別和發生糖尿病之年齡是否影響治療的結果. 第 3 個目地是評估 FBS 和 TC 在基線時是否相關. 第 4 個目地評估 TC 是否可能解釋 FBS 變化的效應. 表 6.1 顯示變數名稱與定義, 資料在檔案 **CTDMInsu.csv**.

圖 6.2 顯示 (左圖) 治療 12 個月之後的空腹血糖 FBS2 與 (右圖) 降低血糖的效應 (= FBS2-FBS1) 分別對基線測量空腹血糖 FBS1 做散佈圖, 圖中區分 2 種不同的治療, 可以看出合成胰島素治療組比一般胰島素控制組, 對較高的基線空腹血糖病患, 有較好之降低血糖的效應.

```

> CTDM.df<-read.table("CTDMInsu.csv",
  header=TRUE, row.names=NULL, sep="," , dec=".")
> CTDM.df
  Therapy Sex Age TC1 TC2 FBS1 FBS2
1      1   1   1  250 210  140  100
2      1   1   1  260 200  150  110
3      1   1   1  300 250  140   90
.....
> plot(FBS2 ~ FBS1, type = "n",
  main = "FBS: 12-month therapy vs. baseline",
  xlab = "FBS: baseline", ylab = "FBS: 12-month therapy")
> points(FBS1[Therapy == 1], FBS2[Therapy == 1], pch = 1)
> points(FBS1[Therapy == 2], FBS2[Therapy == 2], pch = 19)
> abline(lm(FBS2 ~ FBS1, data = CTDM.df[Therapy == 1,]), lty=1, lwd=1.5)
> abline(lm(FBS2 ~ FBS1, data = CTDM.df[Therapy == 2,]), lty=2, lwd=1.5)

```

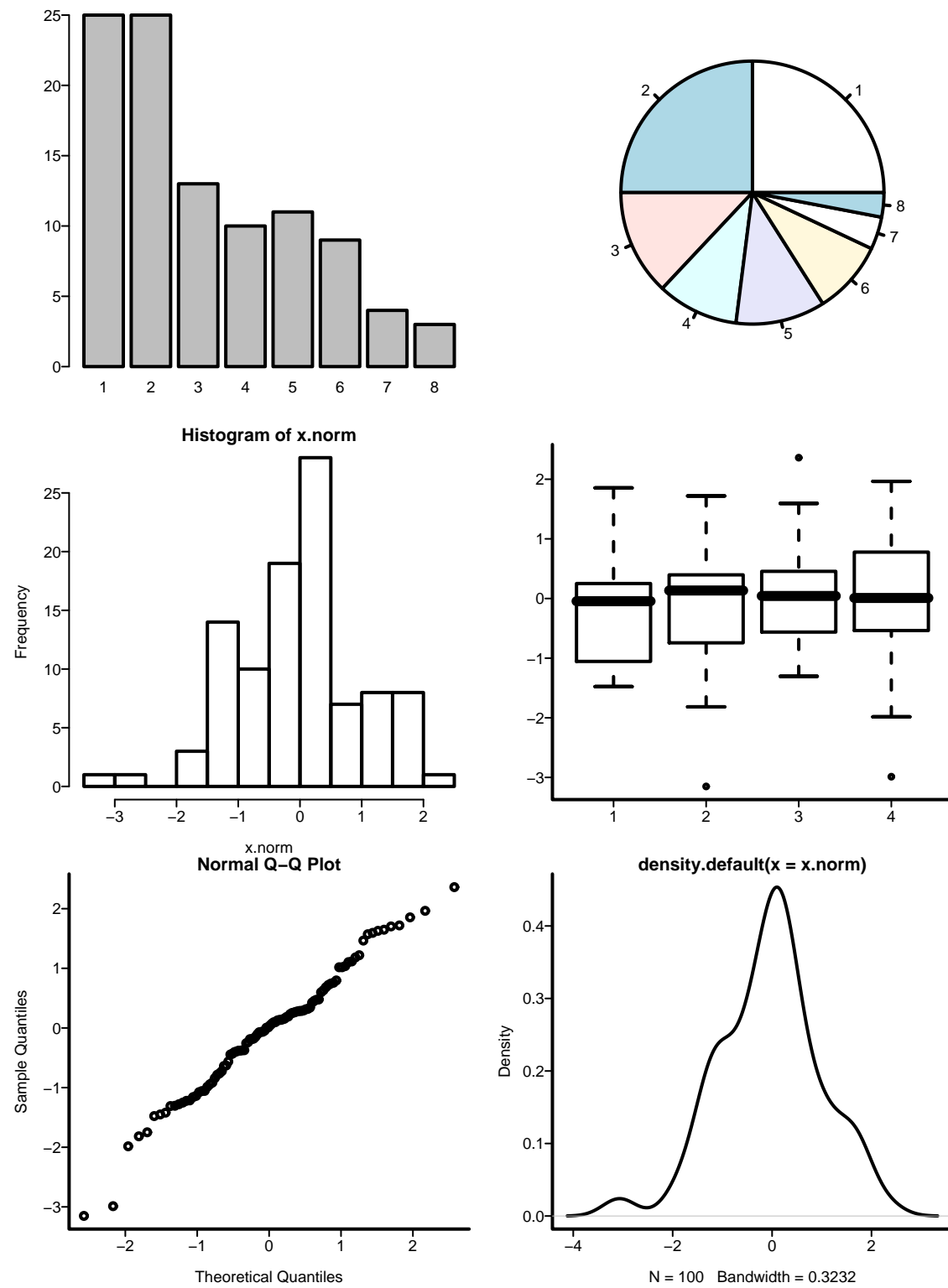


圖 6.1: R 繪圖簡介: 模擬常態分配

```

> legend(130,130, c("Syn. ", "Reg."), pch=c(1,19), lty=1:2, lwd=1.5)
%> #
> plot((FBS2-FBS1) ~ FBS1, type = "n",
      main = "FBS(12-mo Tx - baseline) vs. baseline",
      xlab = "FBS: baseline", ylab = "FBS(12-mo Tx - baseline)")
> points(FBS1[Therapy == 1], (FBS2-FBS1)[Therapy == 1], pch = 1)
> points(FBS1[Therapy == 2], (FBS2-FBS1)[Therapy == 2], pch = 19)
> abline(lm((FBS2-FBS1) ~ FBS1, data = CTDM.df[Therapy == 1,]), lty=1, lwd=1.5)
> abline(lm((FBS2-FBS1) ~ FBS1, data = CTDM.df[Therapy == 2,]), lty=2, lwd=1.5)
> legend(165,-30, c("Syn.", "Reg."), pch=c(1,19), lty=1:2, lwd=1.5)

```

表 6.1: 成人第二型糖尿病 (NIDDM) 臨床試驗比較

合成胰島素與一般胰島素對降低血糖和總膽固醇的效

應: 變數名稱與定義

Therapy	1: 合成胰島素 ; 2: 一般胰島素
Sex	1: 男性; 2: 女性
Age	發生糖尿病之年齡; 1: 20-39; 2: 40-59
TC1	總膽固醇, 基線測量
TC2	總膽固醇, 治療 12 個月後測量
FBS1	空腹血糖, 基線測量
FBS2	空腹血糖, 治療 12 個月後測量

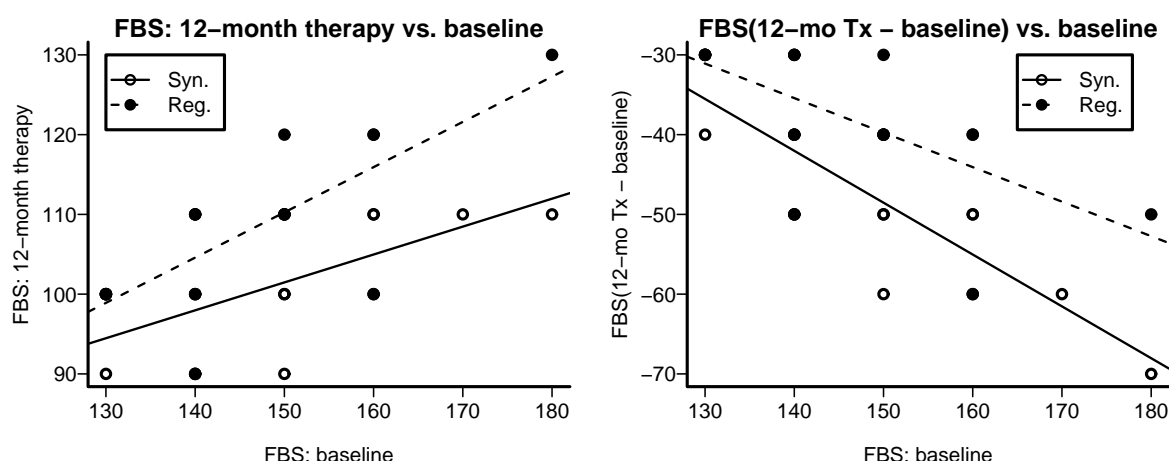


圖 6.2: 臨床試驗比較合成胰島素與一般胰島素對降低血糖和總膽固醇的效應: (左圖) 治療 12 個月之後的空腹血糖 FBS2 與 (右圖) 降低血糖的效應 ($= FBS2 - FBS1$) 分別對基線測量空腹血糖 FBS1 做散佈圖

6.2 高階繪圖 High-Level Plotting Functions

高階繪圖函式對輸入的資料, 產生完整的圖片, 且自動產生坐標軸, 標籤和標題, 當然, 使用者有其他須求, 可以利用函式引數改變. 高階繪圖指令, 會在目前圖形裝置 (graphic device, 如視窗), 開啓一個新的圖區 (plot region), 必要時會清除現有的圖區. 常見的高階繪圖函式參見表 6.2 與圖 6.3.

```
> CTDM.df<-read.table("CTDMInsu.csv", header=TRUE, row.names=NULL, sep=",", dec=".")
> attach(CTDM.df)
> common high-level plot examples
> dotchart(FBS2, main="dotchart()")
> hist(FBS2, main="hist()")
> barplot(FBS2, main="barplot()")
> boxplot(FBS2, main="boxplot()")
> plot(FBS2, FBS1, main="plot()")
> qqnorm(FBS2, main="qqnorm()")
>
> # volcano data
> # Maunga Whau (Mt Eden) is one of about 50 volcanos in the Auckland
> # volcanic field. This data set gives topographic information for
> # Maunga Whau on a 10m by 10m grid.
> # matrix 87 x 61
>
> x.volc<-10*(1:nrow(volcano))
> y.volc<-10*(1:ncol(volcano))
> contour(x.volc, y.volc, volcano, main="contour()")
> image(x.volc, y.volc, volcano, main="image()")
> persp(x.volc, y.volc, volcano, main="persp()")
```

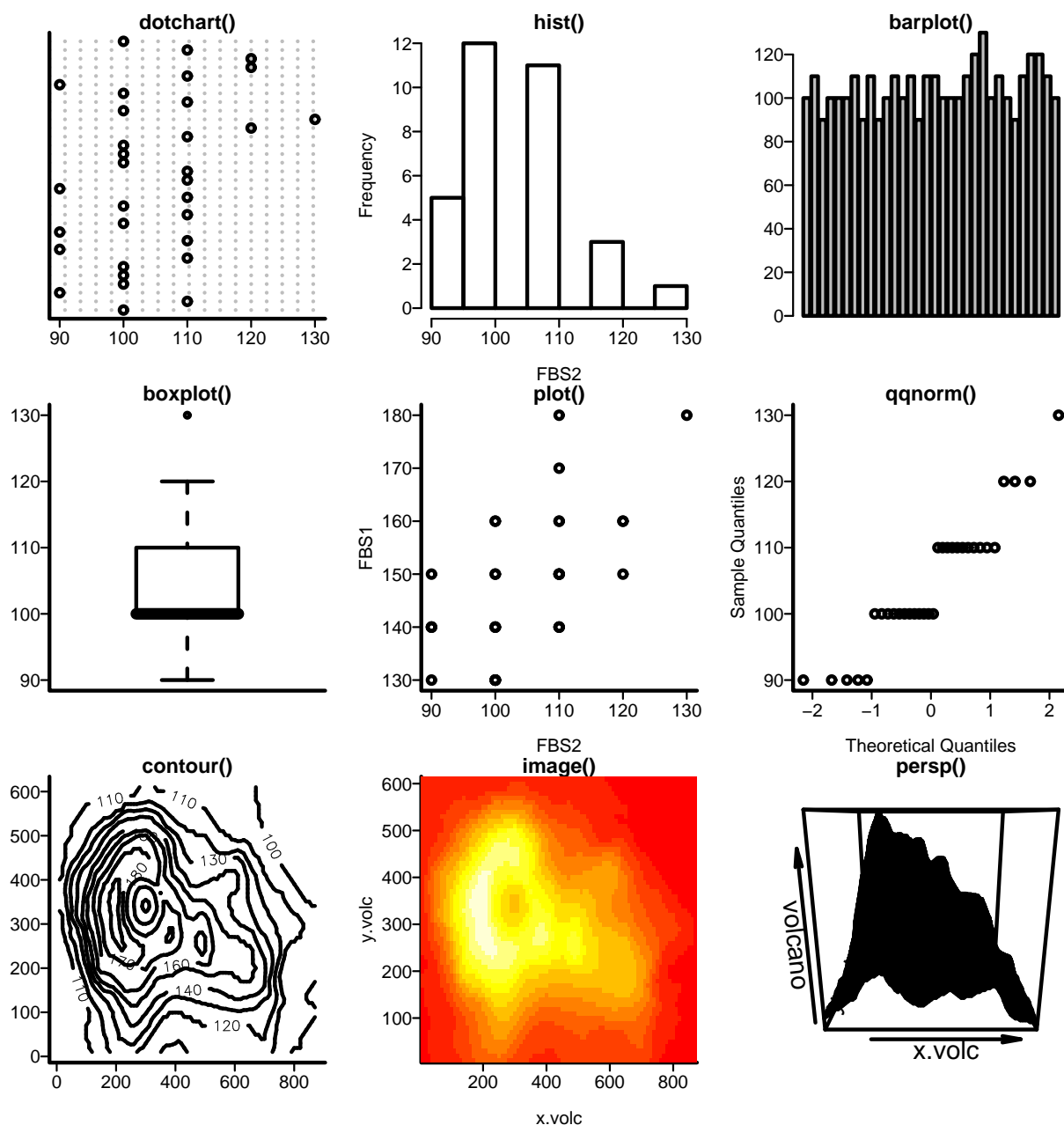


圖 6.3: 常見高階繪圖函式

表 6.2: 常見高階繪圖函式. (符號 † 表示在 **lattice** 套件有相對函式)

R 函式	說明
<code>plot(x)</code>	plot of x (on the y-axis) and ordered on the x-axis
<code>plot(x, y)</code>	X-Y plot scatter plot, bivariate plot of x (on the x-axis) and y (on the y-axis)
<code>sunflowerplot(x,y)</code>	sunflower scatter plot, id. than <code>plot()</code> , but the points with similar coordinates are drawn as flowers which petal number represents the number of points
<code>pie(x)</code>	circular pie-chart
<code>hist(x)</code>	histogram of the frequencies of x
<code>barplot(x)†</code>	bar graphics or histogram of the values of x
<code>boxplot(x)</code>	box-and-whiskers plot
<code>stem(x)</code>	stem and leaf plot
<code>dotchart(x)†</code>	Cleveland dot chart.
<code>qqnorm(x)</code>	Q-Q plot, quantiles of x with respect to the values expected under a normal law
<code>qqplot(x,y)</code>	Q-Q plot, quantiles of y with respect to the quantiles of x
<code>plot.ts(x)</code>	time series plot, if x is an object of class "ts", plot of x with respect to time, x may be multivariate but the series must have the same frequency and dates
<code>ts.plot(x)</code>	time series plot, id. but if x is multivariate the series may have different dates and must have the same frequency
<code>pairs(x)†</code>	scatterplot matrices, pair-wise multivariate plots, if x is a matrix or a data frame, draws all possible bivariate plots between the columns of x
<code>coplot(x ~ y z)</code>	conditioning plot, bivariate plot of x and y for each value of z
<code>interaction.plot(f1, f2, y)</code>	if $f1$ and $f2$ are factors (object), plots the means of y (on the y-axis) with respect to the values of $f1$ (on the x-axis) and of $f2$ (different curves); the option <code>fun</code> allows to choose the summary statistic of y (by default <code>fun=mean</code>)
<code>matplot(x,y)</code>	plot columns of matrices, bivariate plot of the first column of x vs. the first one of y , the second one of x vs. the second one of y , etc.
<code>fourfoldplot(x)</code>	visualizes, with quarters of circles, the association between two dichotomous variables for different populations (x must be an array with <code>dim=c(2, 2, k)</code> , or a matrix with <code>dim=c(2, 2)</code> if $k = 1$)
<code>assocplot(x)</code>	Cohen-Friendly graph showing the deviations from independence of rows and columns in a two dimensional contingency table
<code>mosaicplot(x)</code>	"mosaic" graph of the residuals from a log-linear regression of a contingency table
<code>contour(x,y,z)†</code>	contour plot (data are interpolated to draw the curves), x and y must be vectors and z must be a matrix so that <code>dim(z)=c(length(x), length(y))</code> (x and y may be omitted)
<code>filled.contour(x,y,z)</code>	id. but the areas between the contours are coloured, and a legend of the colours is drawn as well
<code>image(x,y,z)†</code>	id. but with colours (actual data are plotted)
<code>persp(x,y,z)†</code>	id. but in perspective (actual data are plotted)
<code>stars(x)</code>	if x is a matrix or a data frame, draws a graph with segments or a star where each row of x is represented by a star and the columns are the lengths of the segments
<code>symbols(x, y, ...)</code>	draws, at the coordinates given by x and y , symbols (circles, squares, rectangles, stars, thermometres or boxplots) which sizes, colours ... are specified by supplementary arguments
<code>termplot(mod.obj)</code>	plot of the (partial) effects of a regression model (<code>mod.obj</code>)

6.2.1 plot() 函式

在 R 最常用的一個圖形函式 `plot()`。這是一個“通用函數” (generic function): `plot()` 產生的圖形與第一個引數的類型 (class) 有關。

`plot(x, y)`

若 `x` 和 `y` 是數值向量, `plot(x, y)` 產生 `y` 對 `x` 的 散佈圖 (scatter plot)。

`plot(x)`

若 `x` 是一個 時間序列物件 (time series object), 產生一個 時間序列圖 (time series plot)。

若 `x` 是一個數值向量 (numerical), 它將產生一個向量元素值對該向量下標 (index) 的圖。若 `x` 是一個複數向量 (complex), 產生一個複數向量元素的虛部對實部的圖。

`plot(f), plot(f, y)`

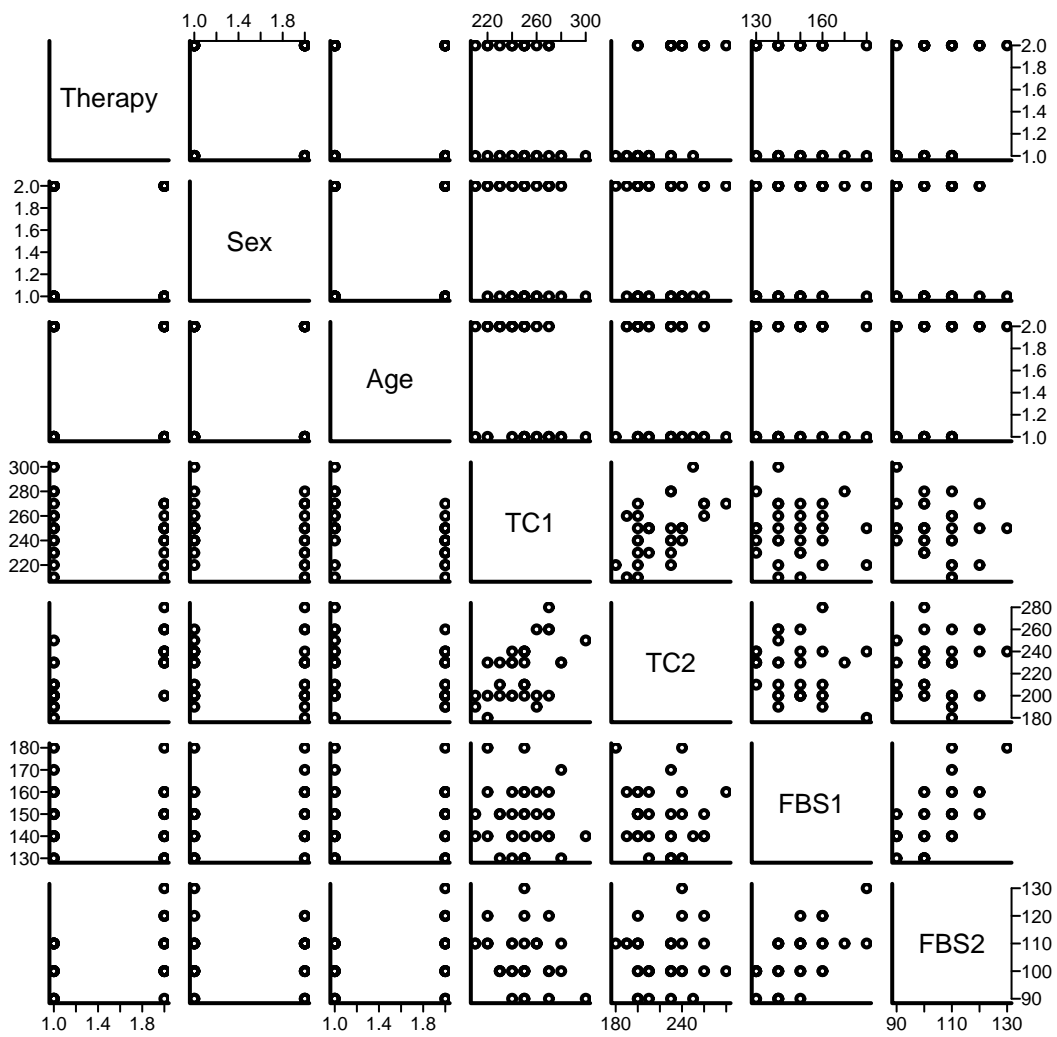
若 `f` 是一個 因數 (因子) 物件 (factor), `y` 是一個數值向量, `plot(f)` 產生 `f` 的 直方圖 (bar graphics), `plot(f, y)` 產生 `y` 在 `f` 的各種水準下的 盒狀圖 (box plot)。

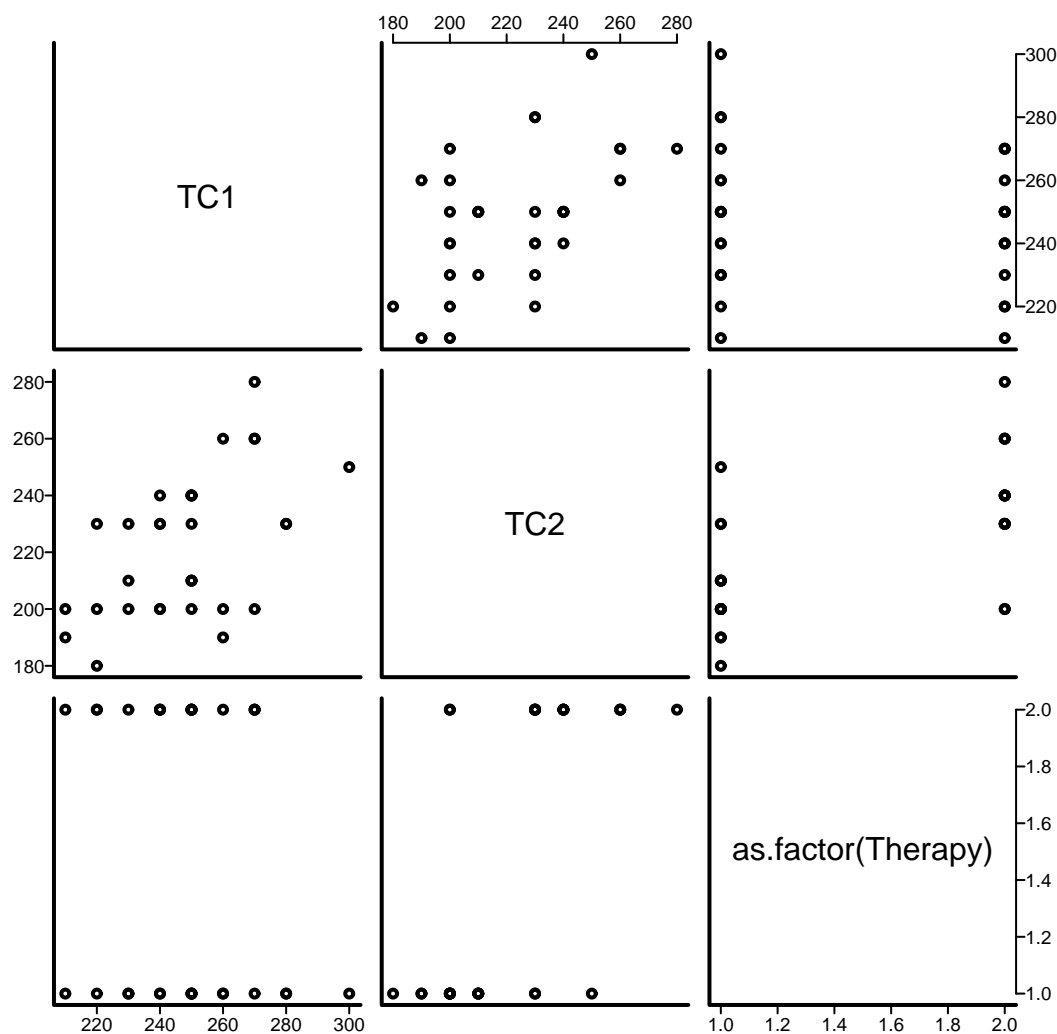
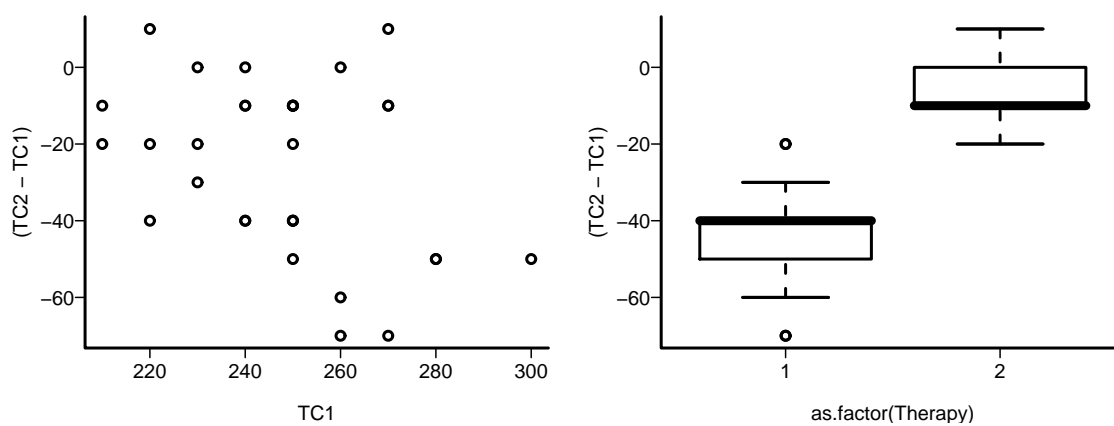
`plot(df), plot(~expr), plot(y~expr)`

若 `df` 是一個資料框物件 (data frame), `y` 是任何物件, `expr` 是一個物件名以 ‘+’ 連結的表列 (如: `a + b + c`, `age + sex`). `plot(df)` 產生資料框中變數的分佈圖; `plot(~expr)` 產生一系列物件的分佈圖; `plot(y~expr)` 產生 `y` 相對 `expr` 中各個物件的分佈圖。

圖 6.4, 6.5, 6.6, 是利用下列部分指令產生。

```
> CTDM.df<-read.table("CTDMInsu.csv", header=TRUE, row.names=NULL, sep="," , dec=".")
> attach(CTDM.df)
> plot(CTDM.df)
>
> plot(CTDM.df)
>
> plot(~ TC1 + TC2 + as.factor(Therapy))
>
> plot((TC2-TC1) ~ TC1 + as.factor(Therapy)) # two plots
```


圖 6.4: `plot(CTDM.df)`

圖 6.5: `plot(TC1 + TC2 + as.factor(Therapy))`圖 6.6: `plot((TC2-TC1) TC1 + as.factor(Therapy))`

6.2.2 高階繪圖函式引數

高階繪圖函式對輸入的資料, 產生完整的圖片, 且自動產生坐標軸, 標籤和標題, 使用者若有其他需求, 可以利用高階繪圖函式引數改變, `plot()` 函式引數的內部設定值如下. 參見表 6.3 與圖 6.11.

```
> plot(x, y,
      type = "p",
      bty="o",
      pch =
      lty =
      cex =
      lwd =
      col =
      bg =
      xlim = NULL, ylim = NULL,
      log = "",
      main = NULL,
      sub = NULL,
      xlab = NULL, ylab = NULL,
      cex.main =
      col.lab =
      font.sub =
      ann = par("ann"),
      axes = TRUE,
      frame.plot = axes,
      panel.first = NULL,
      panel.last = NULL, asp = NA,
      ...)
```

其中

`type="p", type="l", type="b", ..., etc.`

設定繪圖在 (x, y) 顯示方式, 參見圖 6.7.

p: 畫點,

l: 畫線,

b: 畫點同時在點與點之間, 畫線連結

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,

h: 從點到 x -橫軸畫垂直線,

n: 不畫任何點與線, 但容許畫坐標軸且建立坐標系統, 用於後面用低階圖形函式作圖.

```
> CTDM.df<-read.table("CTDMInsu.csv", header=TRUE, row.names=NULL, sep="," , dec=".")
> attach(CTDM.df)
> # compare different types
> FBS.diff<-FBS2-FBS1
> CTDM.id<-c(1:32)
```

```

> plot(CTDM.id, FBS.diff, type="p", main='type="p"', bty="o")
> plot(CTDM.id, FBS.diff, type="l", main='type="l"', bty="o")
> plot(CTDM.id, FBS.diff, type="b", main='type="b"', bty="o")
> plot(CTDM.id, FBS.diff, type="s", main='type="s"', bty="o")
> plot(CTDM.id, FBS.diff, type="S", main='type="S"', bty="o")
> plot(CTDM.id, FBS.diff, type="o", main='type="o"', bty="o")
> plot(CTDM.id, FBS.diff, type="h", main='type="h"', bty="o")
> plot(CTDM.id, FBS.diff, type="n", main='type="n"', bty="o")

```

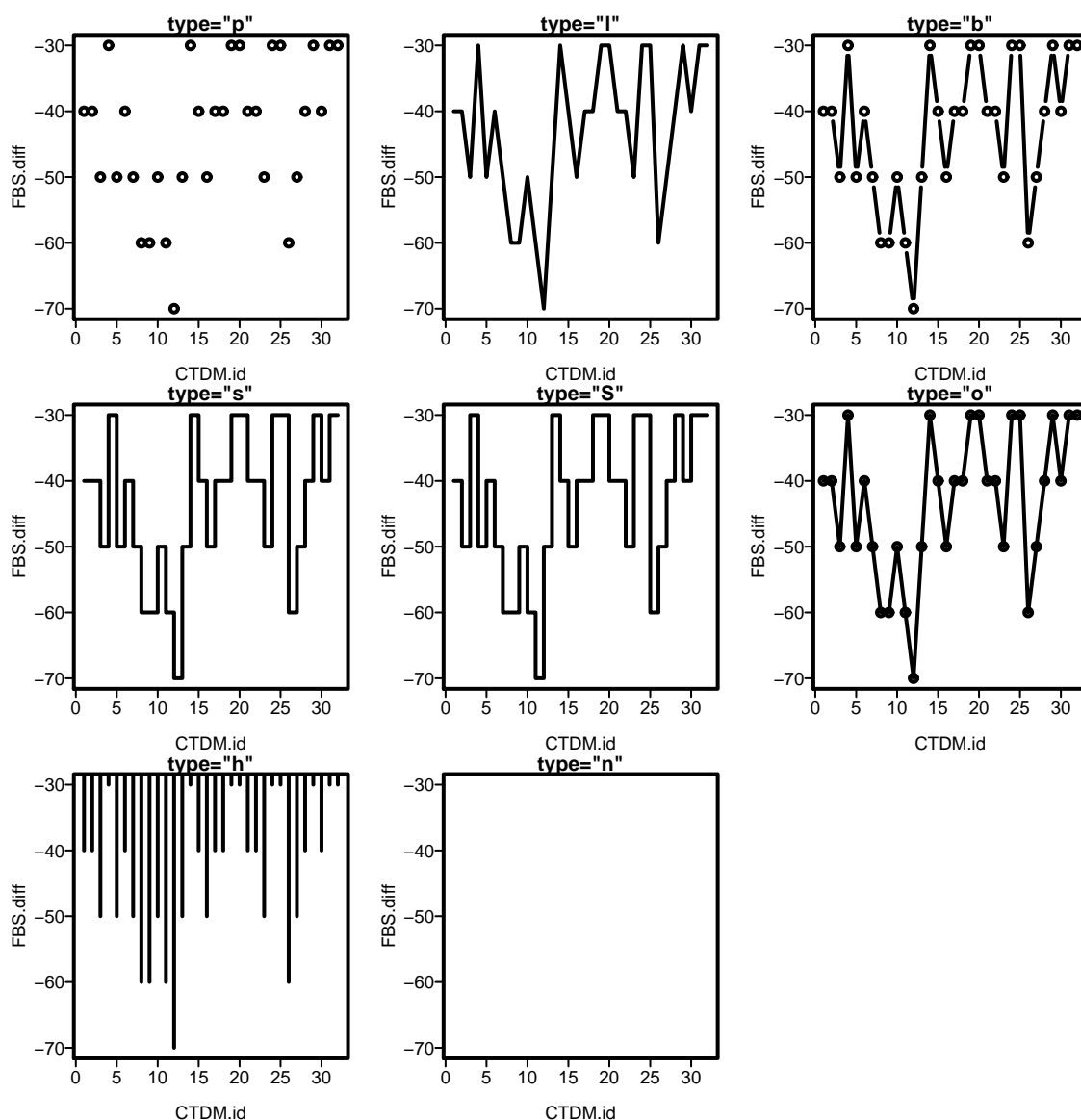


圖 6.7: type 引數設定高階繪圖函式在 (x, y) 顯示方式

bty="o", bty="l", ..., etc.

設定圖形座標軸外框 (box) 的類型, 選項共有 { "o", "l", "7", "c", "u", "]" }, 座標軸外框的類型類似所選之文字, 參見圖 6.8.

```

> CTDM.df<-read.table("CTDMInsu.csv", header=TRUE, row.names=NULL, sep=",", dec=".")
> attach(CTDM.df)
> # compare different bty, box types
> FBS.diff<-FBS2-FBS1
> CTDM.id<-c(1:32)
> plot(CTDM.id, FBS.diff, type="p", main='bty="o"', bty="o")
> plot(CTDM.id, FBS.diff, type="l", main='bty="l"', bty="l")
> plot(CTDM.id, FBS.diff, type="b", main='bty="7"', bty="7")
> plot(CTDM.id, FBS.diff, type="s", main='bty="c"', bty="c")
> plot(CTDM.id, FBS.diff, type="S", main='bty="u"', bty="u")
> plot(CTDM.id, FBS.diff, type="o", main='bty="]"', bty="]")

```

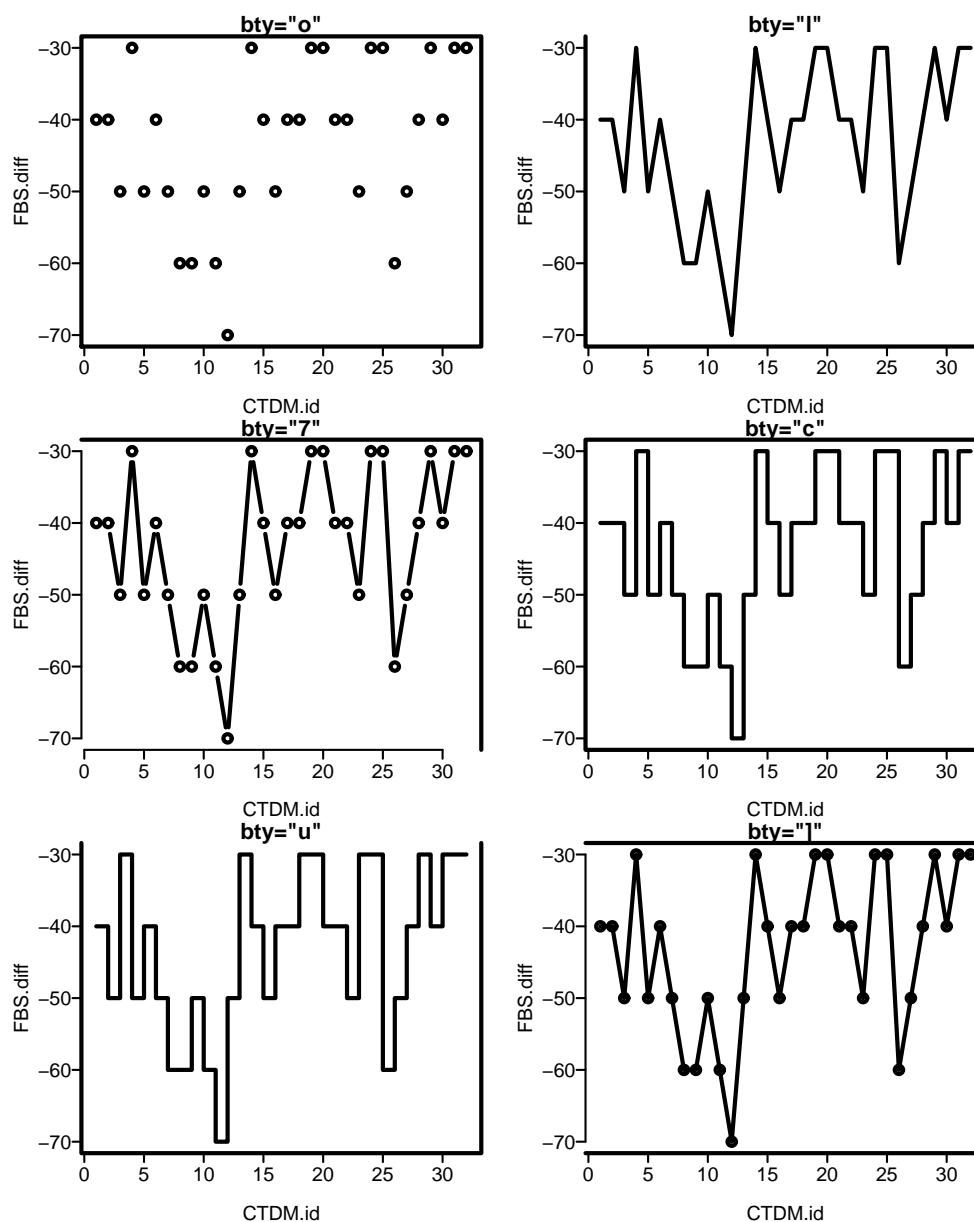


圖 6.8: bty 引數設定高階繪圖函式在圖形座標軸外框外框之顯示方式

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

畫點時, 當 `pch=k` 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, 參見圖 6.9.

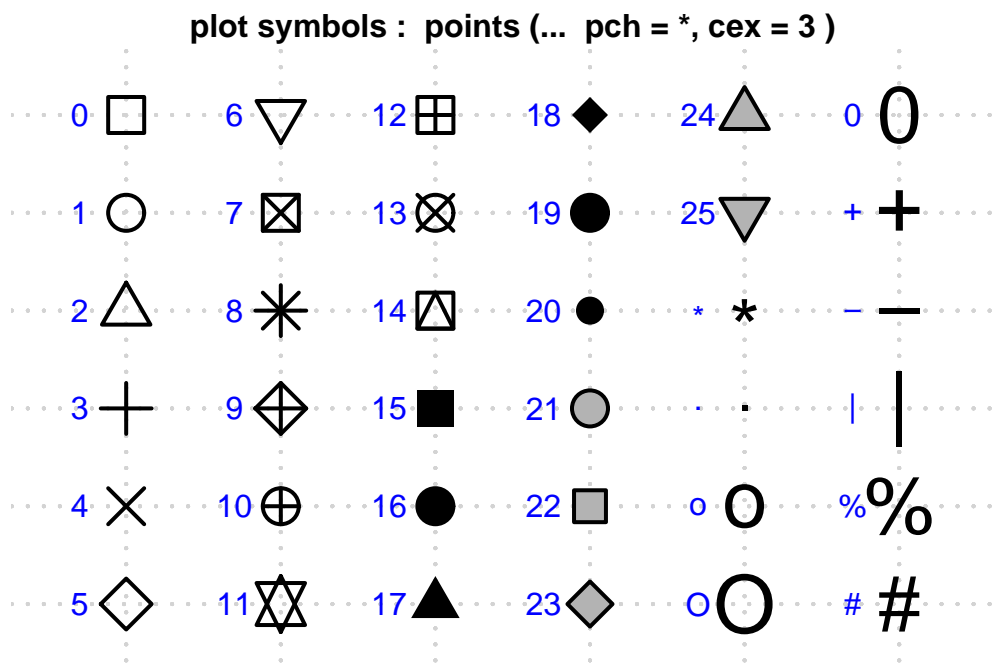


圖 6.9: `pch` 引數設定高階繪圖函式在畫點時之顯示方式

`lty=1, lty=2, ..., etc.`

畫線時, 設定線條類型, 內定預設值為 1. 類型 1 常常是實線 (solid line), 類型 2 和其他常常是不同程度之虛線, 類型 0 是不可見的線條, 但預設值會隨圖形列印機器設定而有些差異, 參見圖 6.10.

`lwd=1, lwd=2, ..., etc.`

畫線時, 設定線條寬度, 以“標準” (`lwd=1`), 線條寬度的倍數設定線條寬度. 座標軸線條和函數 `lines()` 等產生之線條都會受影響, 參見圖 6.10.

`col=1`

`col=1` 設定點, 線的顏色設置, 內定預設值為 `black`. 設定顏色值 (value) 是調色板的數值 (見 `help(palette)`), 或者顏色之文字名稱.

`bg="white"`

設定圖形背景顏色, 內定預設值為 `white`.

`xlim=, ylim=`

設定座標軸範圍之上下界, 如 `xlim=c(1, 10)` 或 `xlim=range(x)`.

`log="x", log="y", log="xy", log="yx"`

設定座標軸是否取對數值, `log="xy", log="yx"` 表示 x 與 y 座標軸都取對數值.

```
main="Title",sub="Subtitle"
```

`main="Title"` 設定圖形主標題 (main title) 定義的文字放在圖形的上方, `sub="Subtitle"` 設定圖形次標題 (subtitle) 定義的文字放在圖形的下方.

```
xlab="X",ylab="Y"
```

設定 x - 與 y - 座標軸標記.

```
ann=TRUE
```

`ann=TRUE`, 當邏輯設定為真 (TRUE) 時, 畫出自動設定之主標題, 座標軸標記.

```
axes=TRUE
```

`axes=TRUE`, 當邏輯設定為真 (TRUE) 時, 畫出自動設定之座標軸與座標軸外框.

```
frame.plot=axes
```

當邏輯設定為真 (TRUE) 時, 設定自動畫出座標軸外框.

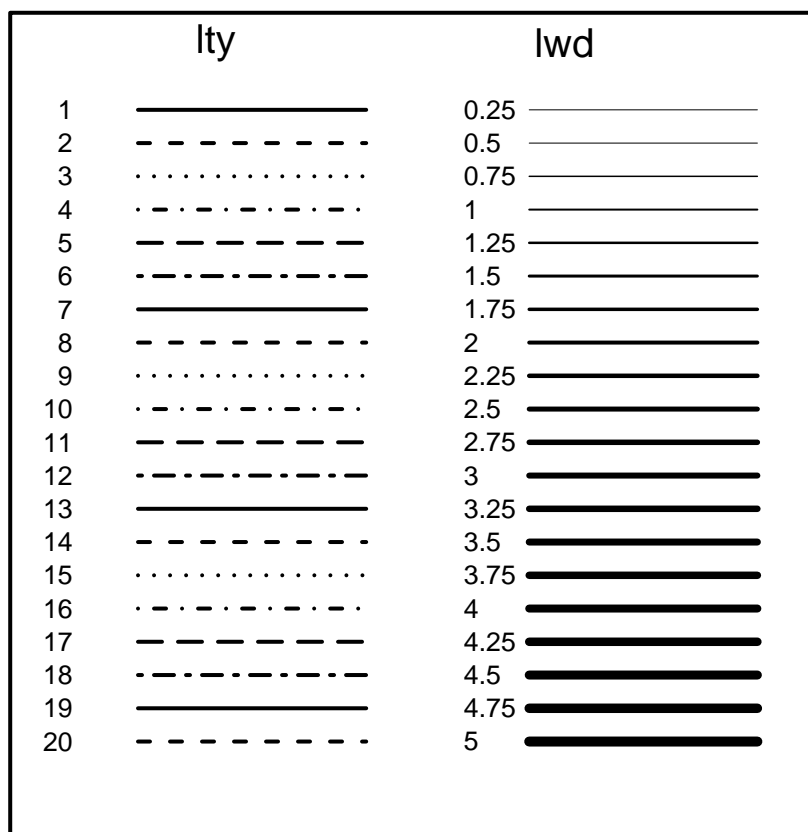


圖 6.10: `lty` 與 `lwd` 引數設定高階繪圖函式在畫線與寬度時之顯示方式

```
> # default
> plot(TC1, TC2-TC1)
>
> # plus some common arguments
> plot(TC1[order(TC1)], (TC2-TC1)[order(TC1)],
       type = "b",
```

```

bty = "o",
pch = 19,
lty = 2,
cex = 1.5,
lwd = 1.5,
xlim = c(200, 320), ylim = c(-100, 20),
main = "TC diff. (= TC2-TC1) vs. TC1",
sub = "High-level plot with arguments",
xlab = "TC at baseline", ylab = "TC diff. (= TC2-TC1)",
axes = TRUE,
ann=TRUE,
frame.plot=TRUE
)

```

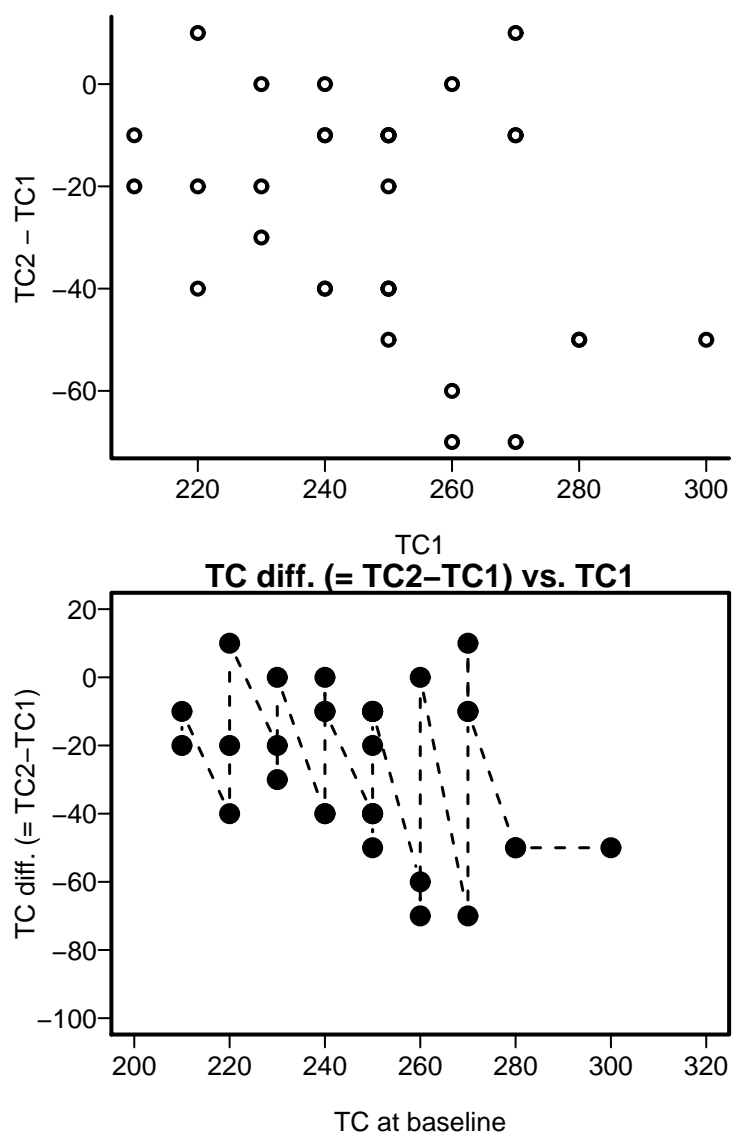


圖 6.11: 高階繪圖函式引數設定之應用

表 6.3: 常用高階繪圖引數

引數	說明
<code>type="p"</code>	specifies the type of plot,
<code>"p":</code>	points;
<code>"l":</code>	lines;
<code>"b":</code>	points connected by lines;
<code>"o":</code>	id. but the lines are over the points;
<code>"h":</code>	vertical lines;
<code>"s":</code>	steps, the data are represented by the top of the vertical lines;
<code>"S":</code>	id. but the data are represented by the bottom of the vertical lines.
<code>bty="l"</code>	A character string which determined the type of "box" which is drawn about plots. If <code>bty</code> is one of "o", "l", "7", "c", "u", or "]" the resulting box resembles the corresponding upper case letter. A value of "n" suppresses the box.
<code>pch=1</code>	specifies the type of points, number 1 to 25 and others.
<code>lty=1</code>	specifies the type of lines. number 0 to 6. (0=blank, 1=solid, 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash", where "blank" uses "invisible lines" (i.e., doesn't draw them).
<code>cex=1.0</code>	a numerical vector giving the amount by which plotting text and symbols should be scaled relative to the default.
<code>lwd=1.0</code>	the line width scaled relative to the default.
<code>col=</code>	The colors for lines and points
<code>bg=</code>	a vector of background colors for open plot symbols.
<code>xlim=, ylim=</code>	specifies the lower and upper limits of the axes, for example with <code>xlim=c(1, 10)</code> or <code>xlim=range(x)</code>
<code>log=</code>	a character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic.
<code>main="Title"</code>	main title, must be a variable of mode character
<code>sub="Sub"</code>	sub-title (written in a smaller font)
<code>xlab="X", ylab="Y"</code>	annotates the axes, must be variables of mode character
<code>ann=TRUE</code>	a logical value indicating whether the default annotation (title and x and y axis labels) should appear on the plot.
<code>axes=TRUE</code>	if FALSE does not draw the axes and the box
<code>frame.plot=axes</code>	a logical indicating whether a box should be drawn around the plot.
<code>panel.first=</code>	an expression to be evaluated after the plot axes are set up but before any plotting takes place. This can be useful for drawing background grids or scatterplot smooths.
<code>panel.last=</code>	an expression to be evaluated after plotting has taken place.
<code>asp=</code>	the y/x aspect ratio, see 'plot.window'.

6.2.3 常用高階繪圖函式

常用高階繪圖函式, `stem()` 函式可以用來繪製莖葉圖 (stem-and-leaf Plot), `boxplot()` 函式可以用來繪製盒形圖 (box plot), `hist()` 函式可以繪製直方圖 (histogram), `qqnorm()` 函式繪製常態機率分位圖 (normal quantile plot) 等, 請參見圖 6.3.

`stem(x)`

`stem(x)` 函式可以用來繪製數值向量 `x` 的“莖葉圖” (stem-and-leaf Plot).

`dotchart(x, ...)`

`dotchart(x, ...)` 函式繪製數值向量 `x` 的“點圖” (dot chart).

`boxplot(x), boxplot(x~f)`

`boxplot(x)` 函式繪製盒形圖, 如圖 6.1, 其中 `x` 為一數值向量, 若 `f` 是因子 (factor) 向量, `boxplot(x~f)` 產生 `x` 在 `f` 的各種水準下的 盒狀圖 (box plot).

`hist(x), hist(x, nclass=n), hist(x, breaks=b, ...)`

`hist(x)` 函式產生數值向量 `x` 的“直方圖” (histogram), 如圖 6.1 `hist(x)` 函式會自動選擇適合的分類 (class) 數目, 也可以通過設定引數 `nclass=k` 或 `breaks=k` 來改變成 `k` 個分類數目. 另一種方法是設定引數 `breaks=c(vector)` (如 `breaks=c(10, 20, 30, 40)`) 精確地設置中斷點 (breakpoint), 內設中斷點是 `right=TRUE`, 為左中括號, 右小括號區間 {如 `[10,20)` }. 如果設定引數 `probability=TRUE`, 直方圖高度表示相對頻率而不是頻數.

`qqnorm(x), qqline(x), qqplot(x, y)`

`qqnorm(x)` 函式繪製常態機率分位圖, `qqnorm(x)` 顯示數值向量 `x` 與 相對常態順序分數期望值 (Normal order scores) 的常態機率分位圖. `qqline(x)` 函式是在這個 `qqnorm(x)` 圖上加一條理論上的直線 (常態分配為 45 度角斜率的直線). `qqplot(x, y)` 函式產生數值向量 `x` 的分位對數值向量 `y` 的分位的圖以比較二者的機率分配是否一致, 若數值向量 `y` 是以之的某一特定機率分配, 則 `qqplot(x, y)` 函式可用來檢驗數值向量 `x` 是否與數值向量 `y` 來自同一機率分配.

`contour(x, y, z, ...), image(x, y, z, ...), persp(x, y, z, ...),`

`contour(x, y, z, ...)` 函式繪製“等高線圖” (contour plot), 顯示數值向量 `x` 與 `y`, 以等高線 (contour line) 來表示數值向量 `z` 的值. `image(x, y, z, ...)` 繪出一個類似等高線圖, 但用不同的顏色表示 `z` 的值; `persp(x, y, z, ...)` 繪出 3-D 表面等高等高.

```
> # common high-level plot examples
> dotchart(FBS2, main="dotchart()")
> hist(FBS2, main="hist()")
> barplot(FBS2, main="barplot()")
> boxplot(FBS2, main="boxplot()")
> plot(FBS2, FBS1, main="plot()")
> qqnorm(FBS2, main="qqnorm()")
>
> # 3-D plot
> # volcano data
> # Maunga Whau (Mt Eden) is one of about 50 volcanos in the Auckland
```

```

> # volcanic field. This data set gives topographic information for
> # Maunga Whau on a 10m by 10m grid.
> # matrix 87 x 61
>
> x.volc<-10*(1:nrow(volcano))
> y.volc<-10*(1:ncol(volcano))
> contour(x.volc, y.volc, volcano, main="contour()")
> image(x.volc, y.volc, volcano, main="image()")
> persp(x.volc, y.volc, volcano, main="persp()")

```

6.2.4 高階繪圖函式對多變量資料繪圖

R 還有一些高階繪圖函式, 可以對多變量資料繪圖。

`pairs(X.mat)`

若 `X.mat` 是一個數值矩陣 (matrix) 或者資料框架 (data frame), `pairs(X.mat)` 函式指令產生 `X.mat` 的“欄與欄” (column) 之間兩兩成對之散佈圖矩陣 (pairwise scatter-plot matrix); `X.mat` 的每一欄相對 `X.mat` 的所有其他欄而產生 $n \times (n - 1)$ 個圖, 並且把這些圖以矩陣形式顯示。這個矩陣形式圖形的尺度是一致的。

`coplot(x~y | z)`

如果 `x` 和 `y` 是數值向量, `z` 是長度都一致之數值向量或者因子物件 (factor), `coplot()` 產生一系列在特定的 `z` 值下 `x` 對 `y` 的散點圖。若 `z` 是因子物件, 這就簡單的表示 `x` 在 `z` 各個水準下對 `y` 畫的散點圖; 當 `z` 是數值向量時, 它將會被分割成一系列條件區間 (conditioning intervals), 對於任一區間, 區間內 `z` 對應的 `x, y` 值繪製 `x` 對 `y` 的散點圖。區間的數值和位置由 `coplot()` 的引數 `given.values=` 控制。

函數 `coplot()` 和 `pairs()` 都有一個引數 `panel=`。這個引數可以用來設置各個面板中的圖形樣式。內定預設值產生散點圖。

以下利用例子 6.1 與 R 內建之料框架 `state`, 關於美國 50 洲的一些相關資料, 用 `coplot()`, `pairs()` 範例, 參見圖 6.12 與圖 6.13。

```

> # coplot()
> CTDM.df<-read.table("CTDMInsu.csv", header=TRUE, row.names=NULL, sep=",", dec=".")
> attach(CTDM.df)
> coplot(FBS2~FBS1 | Therapy)      # (a)
> coplot(FBS2~FBS1 | Therapy*Age) # (b)
>
> # pairs
> pairs(state.x77[,1:5], main="pairs() and pael=smooth",
        pch=16, panel=panel.smooth, lwd=1.5)

```

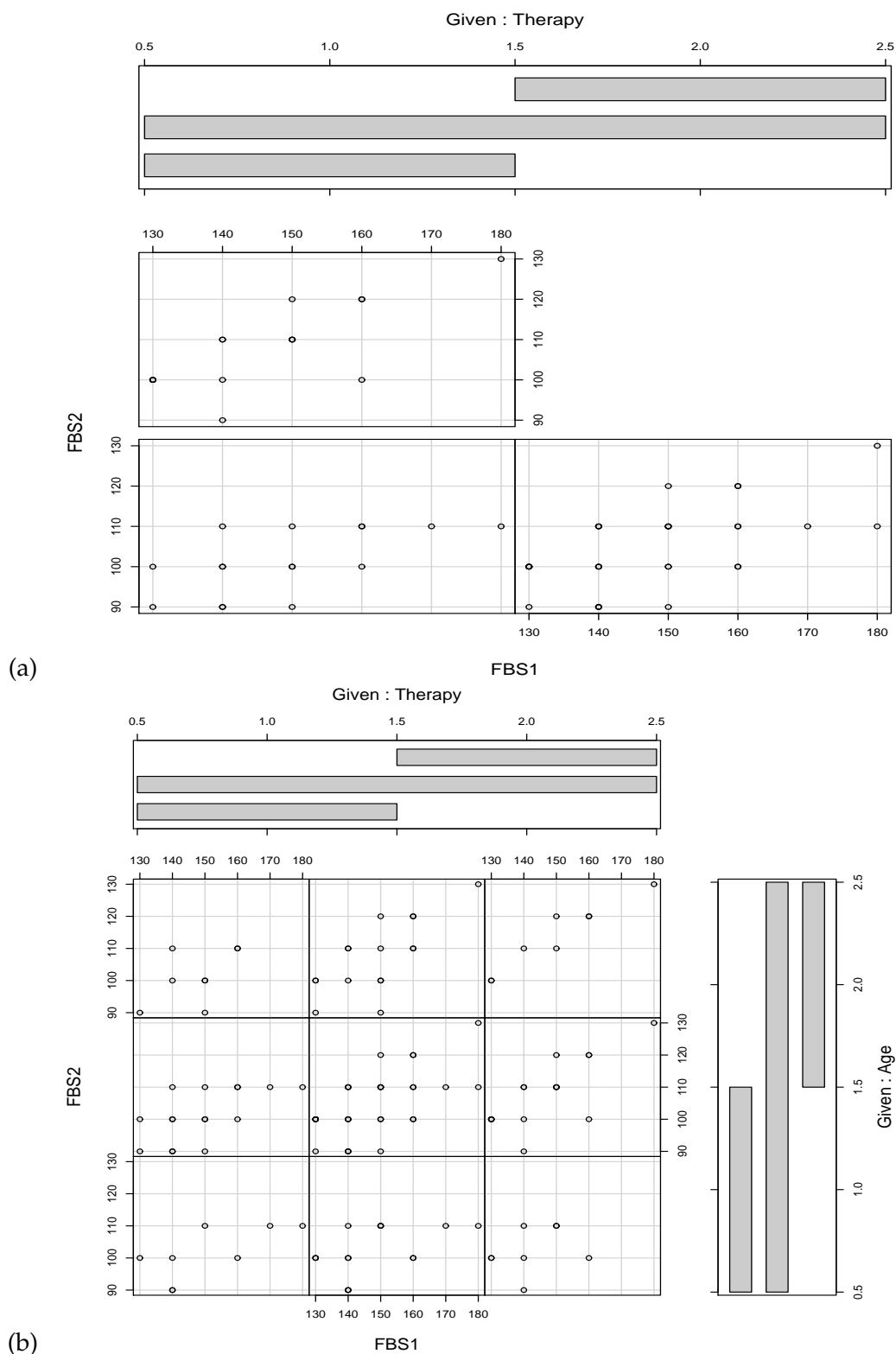
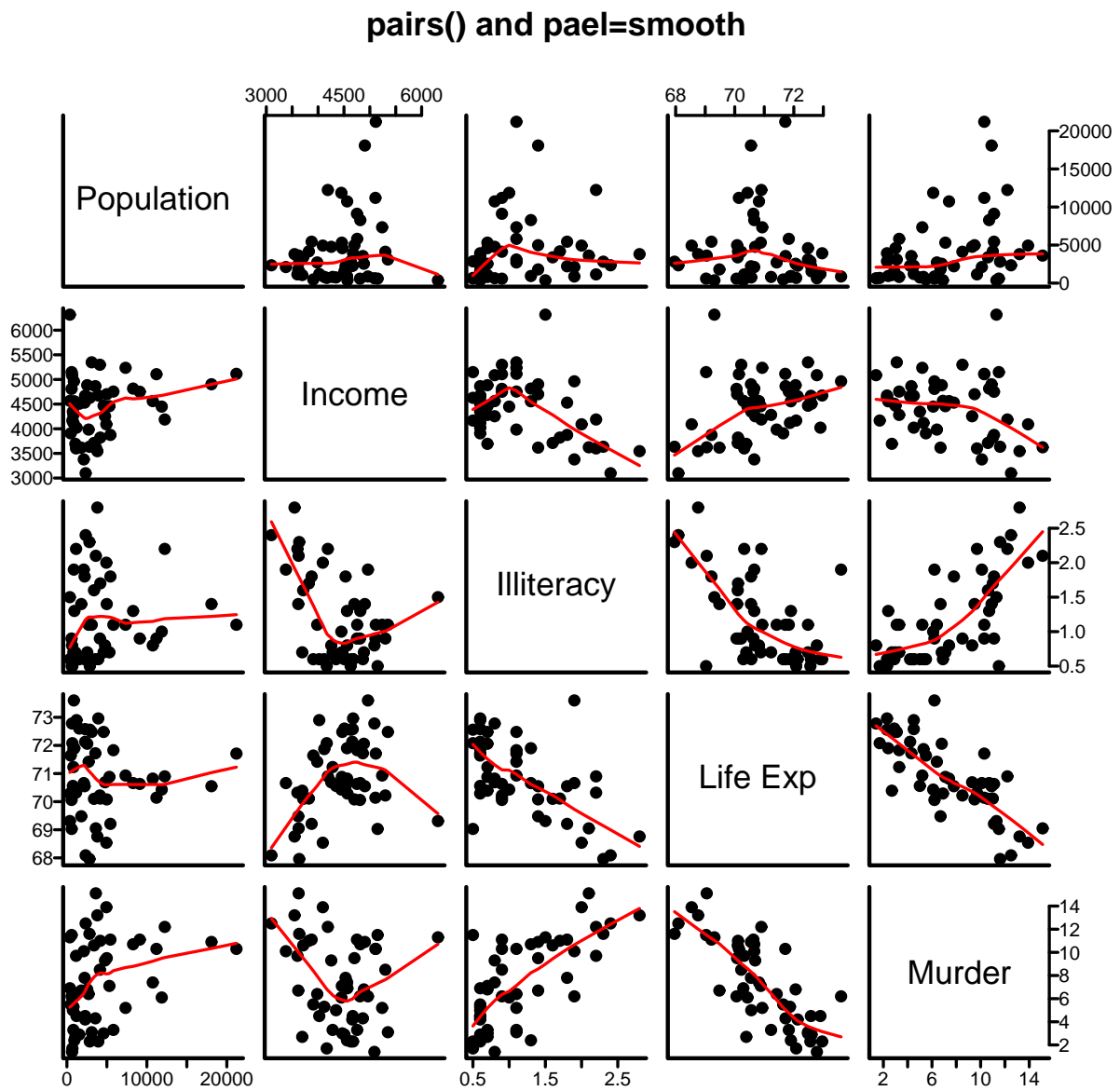


圖 6.12: Therapy 與 Age 是因子物件: (a) `coplot(FBS2 FBS1 | Therapy)`, (b) `coplot(FBS2 FBS1 | Therapy*Age)`

圖 6.13: `pairs()` 高階多變量資料繪圖函式

6.3 低階繪圖函式 Low-Level Plotting Functions

高階圖形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,多個曲線代表之標籤等,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.

6.3.1 常用的低階繪圖函式 Common Low-level plotting Functions

一些非常有用的低級圖形函式,參見表 6.4 與圖 6.14. 例如 `points(x,y)` 與 `lines(x,y)` 在現有的圖形上 (多數由進階圖形函式所建立的圖形) 增加點或者連接線; `abline(a,b)` 可以增加一個斜率為 b 截距為 a 的直線; `title(main=, sub=)` 將 `main` 定義的標題放在圖形的上方,將 `sub` 定義放在圖形的下方; `legend(x,y,legend)` 在指定坐標位置上畫出圖例等,參見 6.15.

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, (多數由進階圖形函式所建立的圖形), 坐標系統中之坐標. 坐標通常用兩個向量 x 和 y 表示,也可以由一個兩欄 (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.

`points(x, y), lines(x, y)`

`points(x, y)` 函式在現有的圖形上加一組點, `lines(x, y)` 函式在現有的圖形上加一組點線. 也可用 `plot()` 之引數 `type=` 設定繪點或繪線.

`text(x, y, labels=z.vec, ...)`

`text(x, y, labels=char.vec, ...)` 在 (x, y) 向量坐標點, 標出由 `labels` 所設定之對應 `z.vec` 數值型或文字型向量, 在 $x[i], y[i]$ 點標出 `labels[i]` 之數值或文字. 通常使用 `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.

`abline(a, b), abline(h=y), abline(v=x), abline(lm.obj)`

在函式在現有的圖形上加畫一條截距為 a 和斜率為 b 直線; `abline(h=y)` 中 `h=y` 引數可畫出在 $Y = y$ 平行於 x -坐標軸之水平線; `abline(v=x)` 中 `v=x` 引數可畫出在 $X = x$ 垂直於 y -坐標軸之垂直線. `abline(lm.obj)` 繪出由線性模型函式建立之線性模型物件的迴歸線.

`polygon(x, y, ...)`

`polygon(x, y, ...)` 畫出以 (x, y) 向量坐標點為端點 (頂點) 之 **多邊形 (polygon)**, 可以用 `col=` 引數指定一個特定色彩, 來填滿多邊形之內部.

`legend(x, y, leg.vec, ...)`

`legend(x, y, leg.vec, ...)` 函式用來在現有的圖形的指定 (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.

`cex` 引數設定文字大小之放大率;

`lty` 引數設定畫線之類型;

`lwd` 引數設定畫線之寬度;

`pch` 引數設定畫點之類型;

`bty="o"` 引數設定圖例外框形式 (內設預定值為加外框);

`title="title text"` 引數設定圖例上方之標題;

`merge=TRUE` 引數設定圖例同時有點與線時, 合併點與線; 另外尚有一些引數, 參見輔助文件.

`title(main, sub)`

`main="Main Title Text"` 設定圖形主標題 (main title) 定義的文字放在圖形的上方, `sub="Subtitle Text"` 設定圖形次標題 (subtitle) 定義的文字放在圖形的下方.

`axis(side, ...)`

`axis(side, ...)` 函式在現有的圖形上加上坐標軸, 其中引數 `side=k`, $k \in \{1, 2, 3, 4\}$, 從圖型底部為 `side=1`, 逆時鐘旋轉, 引數 `at=` 設定刻度 (tick) 位置, 引數 `labels=` 設定刻度處之標籤. 通常使用 `plot(x, y, axes=FALSE)`, 然後用 `axis()` 函式加上坐標軸.

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

`mtext(text, side=3, line=0)` 函式在現有的圖形之邊緣, 加上文字.

```
> # 1 points()
> plot(x,y, type="n", axes=TRUE, ann=FALSE)
> points(x,y,pch=1,cex=1)
> title(main="points")
>
> # 2 lines()
> plot(x,y, type="n", axes=TRUE, ann=FALSE)
> points(x,y,pch=1,cex=1)
> lines(x,y,lty=1, lwd=1.5)
> title(main="lines")
>
> # 3 abline()
> plot(x,y, type="n", axes=TRUE, ann=FALSE)
> points(x,y,pch=1,cex=1)
> abline(2,100, lty=1, lwd=1.5)
> abline(h=1000, lwd=1.5)
> abline(v=10, lwd=1.5)
> abline(lm(y~x), lty=2)
> title(main="abline")
>
> # 4 text()
> plot(x,y, type="n", axes=TRUE, ann=FALSE)
> text(x,y, labels=as.character(c(1:30)),cex=0.8)
> title(main="text")
>
> # 5 symbols()
> plot(x,y, type="n", axes=TRUE, ann=FALSE)
> symbols(x,y, circles=(c(1:30)/15.0), inch=FALSE, lwd=1.5, add=TRUE)
> title(main="symbols")
>
> # 6 segments()
> plot(x,y, type="n", axes=TRUE, ann=FALSE)
> segments(c(1:30), rep(0,30), x, y, lwd=1.5)
```

```

> title(main="segments")
>
> # 7 arrows()
> x.arrow<-seq(1,30,by=3)
> plot(x.arrow, x.arrow, type="n", axes=TRUE, ann=FALSE)
> arrows(x.arrow, rep(15,10), x.arrow, x.arrow, lwd=1.5, length=0.1, code=2)
> title(main="arrows")
>
> # 8 ploygon()
> plot(x,y, type="n", axes=TRUE, ann=FALSE)
> polygon(c(1,15,20,30,15), c(1000, 100, 2000, 1500, 3000), lwd=1.5)
> title(main="polygon")
>
> # 9 box()
> plot(x,y, type="n", axes=TRUE, ann=FALSE)
> box(which="plot", lty=1, lwd=1.5, bty="o")
> box(which="figure", lty=2, lwd=2.0, bty="o")
> title(main="box")

> # low-level parameters
> plot(TC1, TC2-TC1)
>
> ###
> # plus some common arguments
> plot(TC1, TC2-TC1,
       type="n",
       bty="o",
       xlim=c(200, 350), ylim = c(-100, 50),
       axes=FALSE,
       ann=FALSE,
       frame.plot=FALSE
       )
> points(TC1[Therapy==1], (TC2-TC1)[Therapy==1], pch=1)
> points(TC1[Therapy==2], (TC2-TC1)[Therapy==2], pch=15)
>
> TC11<-TC1[Therapy==1]
> TC12<-TC1[Therapy==2]
> TC.diff1<-TC2[Therapy==1]-TC1[Therapy==1]
> TC.diff2<-TC2[Therapy==2]-TC1[Therapy==2]
> lines(TC11[order(TC11)],TC.diff1[order(TC11)], lty=1, lwd=1.5)
> lines(TC12[order(TC12)],TC.diff2[order(TC12)], lty=2, lwd=1.5)
>
> abline(lm(TC.diff1~TC11), lty=1, lwd=1)
> abline(lm(TC.diff2~TC12), lty=2, lwd=1)
>
> axis(1, labels=c("200mg", "250mg", "300mg", "350mg"), at=c(200, 250, 300, 350))
> axis(2, labels=c("-100mg", "-50mg", "0mg", "50mg"), at=c(-100, -50, 0, 50), las=3)

```



```

>
> title(main="TC difference vs. TC at baseline",
        sub="Low-Level example",
        xlab="TC at baseline",
        ylab="TC difference")
> mtext("adding texts", side=4, cex=1.2, las=3)
>
> legend(275, 50, legend=c("1=Synthetic", "2=Regular"), pch=c(1,15), lty=c(1,2),
        title="Insulin")
>
> box(which="plot", lty=1, lwd=1.5, bty="o")
> box(which="figure", lty=2, lwd=1.5, bty="o")

```

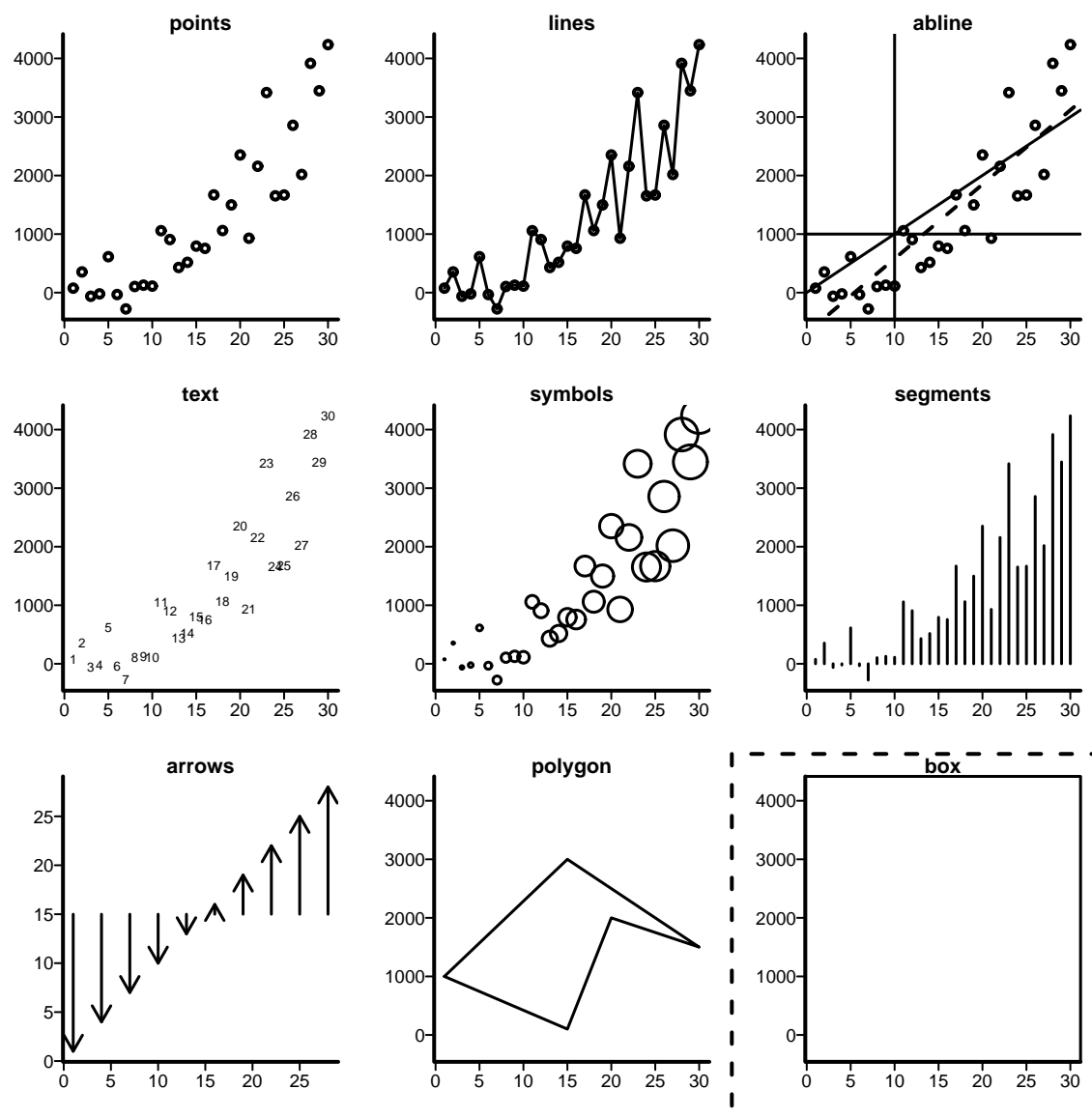


圖 6.14: 常見低階繪圖函式

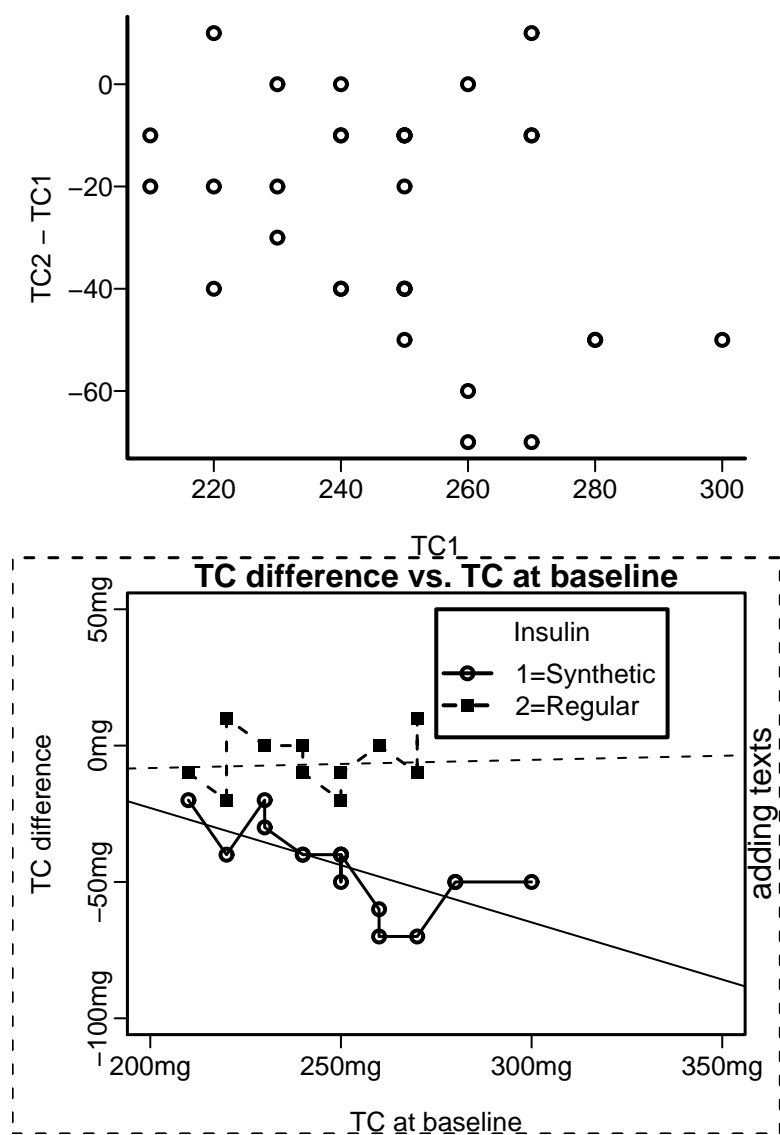


圖 6.15: 低階繪圖函式使用方式

表 6.4: 常用的低階繪圖函式

R Commands	Short description
<code>points(x,y)</code>	Add points to the current plot.
<code>lines(x,y)</code>	Add lines to the current plot.
<code>abline(a,b)</code>	Add lines to the current plot in slope-intercept form.
<code>axis(side,vec)</code>	Add an axis to the plot.
<code>title(main=, sub=)</code>	Add title(s).
<code>mtext(text,side=3,line=0)</code>	Add text in the margins.
<code>text(x,y,labels=)</code>	Add text symbols to the current plot.
<code>legend(x,y,legend)</code>	Add a legend to the current plot.
<code>segments(x0,y0,x1,y1)</code>	Draw line segments on the current plot.
<code>arrows(x0,y0,x1, angle=30,code=2)</code>	Draw arrows on the current plot.
<code>rect(x1,y1,x2,y2)</code>	Draw a rectangle.
<code>polygon(x,y)</code>	Add polygon(s) to the present plot, possibly filled.
<code>frame()</code>	Complete the current plot and advance to a new graphics frame.
<code>box()</code>	Draw a box on the current plot.
<code>locator(n, type="n")</code>	Use mouse to locate <i>n</i> points and return the co-ordinates.
<code>identify(x)</code>	Locate the mouse-pointer position return the coordinates.
<code>rug(x)</code>	Adds a “rug” representation (1-D plot) of the data to the plot.

6.3.2 互動式繪圖函式 Interactive Graphics Functions

R 有兩個函式 `locator()` 和 `identify()` 可以使用者用滑鼠遊標在圖形中, 點擊所想知道的坐標位置, 然後 R 回傳圖形中滑鼠遊標所點擊的位置的確定坐標。

`locator(n, type)`

函式 `locator(n, type)` 指令執行時, 會停下來等待使用者在現有圖形中, 用滑鼠左鍵與遊標點擊現有的圖上的特定位置, 這個過程容許 *n* 個點 (內佈自動設定為 512 個點), 用滑鼠中鍵點擊可以選擇停止, 引數 `type` 容許在選擇的特定之點位置上作指定繪圖, 在滑鼠遊標點擊處繪點 (線) 等, 高級繪圖指令的功能. `locator()` 回傳是一個列表 (list), 有兩個成分, *x* 和 *y*, 分別表示點擊位置的 *x*-(橫) 坐標和 *y*-(縱) 坐標. 當使用者需要設定一些圖形元素 (如圖例與標籤) 在現有的圖上之放置位置時, 互動式選取特定位置, 是一種可行的方法。

```
text(locator(1), "Outlier", adj=0)
```

```
identify(x, y, labels)
```

函式 `identify(x, y, labels)` 容許使用者將 `labels` 的定義, 利用滑鼠左鍵, 放置在滑鼠遊標點擊處決定的 `x` 和 `y` 附近, 在 `labels` 未設定時, 內部自動設定為點的索引值 (`index number`) 或序號, 回傳選擇的點之索引值; 或當使用引數 `labels="樣本名字"` (特定文字資訊) 並使用引數 `plot=FALSE`, 加上使用滑鼠右鍵點擊時, 可回傳選擇的點之索引值. 使用時機通常是想選定現有的圖上之一些特定點, 然後以某一種方式加以轉換或分析處理, 因此需要選擇的點之索引值. 而不是選擇的點之坐標位置.

```
> CTDM.df<-read.table("CTDMInsu.csv", header=TRUE, row.names=NULL, sep="," , dec=".")
> attach(CTDM.df)

> plot(TC1, TC2-TC1)
> locator(5, type="p", pch=15, cex=1.5)
> text(locator(2), "Outlier", adj=0)

> identify(250, 0, labels="id", pos=TRUE)
```

6.3.3 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 6.5. 例如在二項機率函數加上數學符號:

```
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.

```
help(plotmath)
example(plotmath)
demo(plotmath)
```

表 6.5: 常見繪圖用數學符號-I

Arithmetic Operation		
$x + y$	$x + y$	x plus y
$x - y$	$x - y$	x minus y
$x * y$	xy	juxtapose x and y
x / y	x/y	x forwardslash y
$x \%+ \% y$	$x \pm y$	x plus or minus y
$x \%/% y$	$x \div y$	x divided by y
$x \%* \% y$	$x \times y$	x times y
$- x$	$-x$	minus x
$+ y$	$+y$	plus y
Sub/Scuperscript		
$x[i]$	x_i	x subscript i
x^2	x^2	x superscript 2
Juxtaposition		
$x*y$	xy	juxtapose x and y
$\text{paste}(x, y, z)$	xyz	juxtapose x, y, and z
Radical		
$\text{sqrt}(x)$	\sqrt{x}	square root of x
$\text{sqrt}(x,y)$	$\sqrt[y]{x}$	yth root of x
Relation		
$x == y$	$x = y$	x equals y
$x != y$	$x \neq y$	x is not equal to y
$x < y$	$x < y$	x is less than y
$x <= y$	$x \leq y$	x is less than or equal to y
$x > y$	$x > y$	x is greater than y
$x >= y$	$x \geq y$	x is greater than or equal to y
$x \%~\% y$	$x \approx y$	x is approximately equal to y
$x \%=\% y$	$x \cong y$	x and y are congruent
$x \%==\% y$	$x \equiv y$	x is defined as y
$x \%prop\% y$	$x \propto y$	x is proportional to y

表 6.6: 常見繪圖用數學符號-II

Typeface		
plain(x)	x	draw x in normal font
bold(x)	x	draw x in bold font
italic(x)	<i>x</i>	draw x in italic font
bolditalic(x)	<i>x</i>	draw x in bolditalic font
List		
list(x, y, z)	<i>x, y, z</i>	comma-separated list
...	...	ellipsis (height varies)
cdots	...	ellipsis (vertically centred)
ldots	...	ellipsis (at baseline)
Set Relation		
x %subset% y	$x \subset y$	x is a proper subset of y
x %subseteq% y	$x \subseteq y$	x is a subset of y
x %notsubset% y	$x \not\subseteq y$	x is not a subset of y
x %supset% y	$x \supset y$	x is a proper superset of y
x %supseteq% y	$x \supseteq y$	x is a superset of y
x %in% y	$x \in y$	x is an element of y
x %notin% y	$x \notin y$	x is not an element of y
Accents		
hat(x)	\hat{x}	x with a circumflex
tilde(x)	\tilde{x}	x with a tilde
dot(x)	\dot{x}	x with a dot
ring(A)	\AA	x with a ring
bar(xy)	\bar{xy}	xy with bar
widehat(xy)	\widehat{xy}	xy with a wide circumflex
widetilde(xy)	\widetilde{xy}	xy with a wide tilde
underline(x)	\underline{x}	draw x underlined
Spacing		
x ~~ y	$x \quad y$	

表 6.7: 常見繪圖用數學符號-III

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>	$x \Uparrow y$	x double-up-arrow y
<code>x %dbldown% y</code>	$x \Downarrow y$	x double-down-arrow y
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>	$\vartheta, \varphi, \varsigma, \varpi$	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
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

表 6.8: 常見繪圖用數學符號-IV

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>	$x + \frac{1}{}$	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>	$\cup_{i=1}^n A_i$	union of A[i] for i equals 1 to n
<code>intersect(A[i], i==1, n)</code>	$\cap_{i=1}^n A_i$	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>	$\min_{x > 0}^{(x)} g(x)$	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

6.4 圖形引數 Graphics Parameters: `par()` Function

當 R 的內部自動設定之繪圖無法達成使用者之要求，特別是發表論文之圖形要求，R 提供許多“圖形引數”(graphics parameters) 可以控制特別需求，如色彩，此尺寸等。R 圖形引數都有一特定名稱，透過 `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。

在此將要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。在本章第 6.2.2 小節中，簡要說明一些常用高階繪圖引數，如表 6.3，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()` 函式的引數。

圖形引數用以下的形式設定

```
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.

6.4.1 永遠改變設定圖形引數 與 `par()` 函式

Permanent Changes and `par()` Function

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()
```

沒有引數, 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"))
```

設定某一特定圖形引數名稱之文字向量, 回傳指定的引數名稱之設定值列表, 如 `par(c("col", "lty"))`.

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

永遠改變某一特定圖形引數之設定值, 顯示或回傳特定圖形引數之的初史設定值 (original values). 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.

```
> oldpar<-par(no.readonly=TRUE)
```

```
some plotting commands
```

```
> par(oldpar)
```

6.4.2 暫時改變設定圖形引數 Temporary Changes

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:

```
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. 但並不是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.

6.4.3 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), 見表 6.9, 與圖 6.9, 圖 6.8, 圖 6.10, 圖 6.11.

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

用於顯示點的符號. 內定預設值常是一個小與圓圈, 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.

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

當 `pch=k` 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. 另外, `pch` 可以是 32–255 範圍內的字元和數字, 顯示一個相對應字元. 參見圖 6.9.

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

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.

`lty=1, lty=2, ..., etc.`

設定線條類型, 內定預設值為 1. 類型 1 常常是實線 (solid line), 類型 2 和其他常常是不同程度之虛線, 類型 0 是不可見的線條, 但預設值會隨圖形列印機器設定而有些差異, 參見圖 6.10.

`lwd=1, lwd=2, ..., etc.`

設定線條寬度, 以“標準” (`lwd=1`), 線條寬度的倍數設定線條寬度. 座標軸線條和函數 `lines()` 等產生之線條都會受影響, 參見圖 6.10.

`col=1, ..., col.axis, col.lab, col.main, col.sub, ..., etc.`

`col=1` 設定點, 線, 文本, 填充區域 (filled region) 和圖像 (image) 的顏色設置, 內定預設值為 black. 設定顏色值 (value) 是調色板的數值 (見 `help(palette)`), 或者顏色之文字名稱.

`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.

```
> ColorName<-colors()
```

```
> ColorName
```

[1] "white"	"aliceblue"	"antiquewhite"
[4] "antiquewhite1"	"antiquewhite2"	"antiquewhite3"
.....		
[652] "yellow"	"yellow1"	"yellow2"
[655] "yellow3"	"yellow4"	"yellowgreen"

`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, font.lab, font.main, font.sub`

設定座標軸標注, x 和 y 座標軸標籤, 主標題和次標題文字的字體類型.

`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%.

`cex=1, cex=1.5, ..., etc.`

設定文字字體 (`character size`) 大小之放大率, 內定預設值為 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.

表 6.9: 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>	Plotting character or mark.
<code>lty</code>	<code>lty=2</code>	Line type: solid, dashed, dotted, etc.
<code>lwd</code>	<code>lwd=2</code>	Line width, usually as a multiple of default width.
<code>col</code>	<code>col=2</code>	Colour for symbol, line or region.
<code>cex=1.0</code>	<code>cex=1.5</code>	a numerical vector giving the amount by which plotting text and symbols should be scaled relative to the default.

6.5 座標軸和座標刻度設定 Axes and Tick Marks

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 6.10 與 圖 6.15.

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)** (劃分坐標軸線上的刻度).
3. **坐標軸刻度標記 (tick label)** (坐標軸線刻度上的單位).

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

`axis` 引數如下;

`bty="o", bty="l", ..., etc.`

設定圖形座標軸外框 (`box`) 的類型, 選項共有 "o", "l", "7", "c", "u", or "]", 座標軸外框的類型類似所選之文字.

`lab=c(5, 5, 7), ..., etc.`

`lab=()` 前兩個引數分別設定 x 和 y 座標軸的刻度間隔數目 (numbers of tick intervals). 第 3 個引數刻度是座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, 導致所有的座標軸刻度標記變成一樣的數位. 內定預設值為 (5,5,7).

`las=0, las=1, ..., etc.`

設定座標軸刻度標記的方向, 內定預設值為 0. 0 表示總是平行於座標軸, 1 表示平行於座標軸, 2 表示垂直於座標軸, 3 表示總是垂直於座標軸.

`mgp=c(3, 1, 0), ..., etc.`

設定三個座標成分, 座標軸標籤, 座標軸刻度標記, 座標軸相對到軸線的距離的位置. 第 1 個引數是座標軸標籤相對座標軸位置的距離; 第 2 個引數表示座標軸刻度標記相對座標軸位置的距離; 第 3 個引數是座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 (3,1,0).

`tck= NA, tck=0.01, ..., etc.`

設定座標軸刻度的長度, 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 座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. `tck=0.01` 和 `mgp=c(1, -1.5, 0)` 表示內部刻度. 內定預設值為 NA.

`xaxs="r", yaxs="r", xaxs="r", yaxs="i", ..., etc.`

分別設定 x 和 y 座標軸的形式. "i" (內在的); "r" (內定預設值) 刻度都包含繪圖資料的範圍. 但是 "r" 形式的座標軸刻度會在座標軸刻度範圍兩邊留一些空隙.

表 6.10: 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>	Axis limits.
<code>bty</code>	<code>bty="o",</code>	Box type, one of ("o", "l", "7", "c", "n")
<code>lab</code>	<code>lab=c(3, 7, 4)</code>	Tick marks and labels with numerical values.
<code>las</code>	<code>las=2</code>	Label orientation. 0 = parallel to axis, 1 = horizontal, 2 = vertical.
<code>log</code>	<code>log="y"</code>	Control log axis scales.
<code>mgp</code>	<code>mgp=c(3, 1, 0)</code>	Axis location.
<code>tck</code>	<code>tck=-0.01</code>	Tick mark length as signed fraction of the plot region dimension.
<code>xaxp, yaxp</code>	<code>xaxp = c(2, 10, 4)</code>	Tick mark limits and frequency.
<code>xaxs, yaxs</code>	<code>xaxs = "i"</code>	Style of axis limits.
<code>xaxt, yaxt</code>	<code>xaxt="s"</code>	Axis type. "n" (null), "s" (standard), "t" (time) or "l" (log).

6.5.1 圖畫邊緣 Figure Margins

R 裏單獨的圖被稱為 **figure**, 並且還包括一個邊緣 (**margin**) (可包括坐標軸標籤, 標題等), 坐標軸包圍的繪圖區域 (**plot region**), 參見表 6.11 與圖 6.16. 控制圖畫版面的圖形引數有 `mar()`, `mai()`.

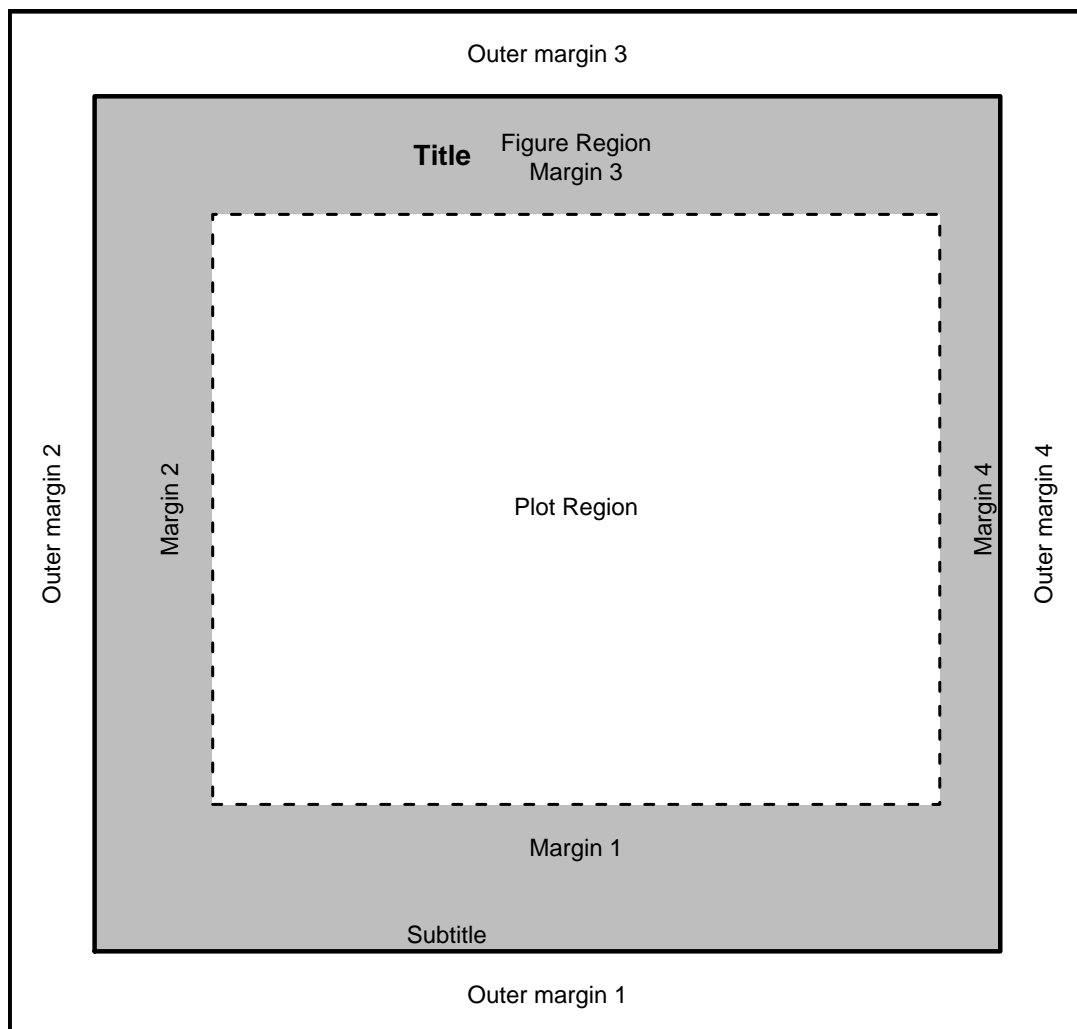


圖 6.16: R 圖形與圖畫邊緣解剖

表 6.11: 常見圖形引數中之圖畫邊緣設定

din, fin, pin	fin=c(6, 4)	Absolute device size, figure size and plot region size in inches.
fig	fig=c(0, 0.5, 0, 1)	Define the figure region as a fraction of the device region.
font	font=3	Small positive integer determining a text font for characters and hence an interline spacing. In R font 1 is plain, font 2 italic, font 3 bold, font 4 bold italic and font 5 is the symbol font.
mai, mar	mar=c(3, 3, 1, 1) + 0.1	The four margin sizes, in inches (mai), or in text line units (mar, that is, relative to the current font size). Note that mar need not be an integer.
mex	mex=0.7 & Number of text	Lines per interline spacing.
mfg	mfg=c(2, 2, 3, 2)	Define a position within a specified multi-figure display.
mfrow, mfcpl	mfrow=c(2, 2)	Define a multi-figure display.
new	new=T	Logical value indicating whether the current figure has been used.
oma, omi, omd	oma=c(0, 0, 4, 0)	Define outer margins in text lines or inches, or by defining the size of the array of figures as a fraction of the device region.
plt	plt=c(0.1, 0.9, 0.1, 0.9)	Define the plot region as a fraction of the figure region.
pty	pty="s"	Plot type, or shape of plotting region, "s" or "m"
usr	usr=c(0.5, 1.5, 0.75, 10.25)	Limits for the plot region in user coordinates

6.5.2 多重圖形環境 Multiple Figure Environment

R 容許在一個單頁上建立一個 $n \times m$ 圖形矩陣. 每一個圖有自己的邊緣, 圖形陣列有外邊緣 (outer margin) 包圍, 參見圖 6.17.

常見多重圖形相關的圖形引數如下:

`mfrow=c(1,1), mfrow=c(2,2), mfc col=c(1, 1), ..., etc.`

設定多重圖形陣列數目. 第一個引數值是列數 (row), 第二個值是欄數, 差異在於 `mfc col(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 6.17 中版面設計以 `mfrow=c(3,2)` 設定.

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

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

`layout=x.matrix` 將圖形裝置分割程 `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 6.18 與圖 6.19.

`mfg=c(2, 2, 3, 2), ..., etc.`

設定 `mfg` 引數將會調整現有的圖片 (current figure) 在一個多重圖形環境中的位置. 前兩個數字是現有的圖片的列 (row) 與欄 (column) 編號, 後面兩個數字是多重圖形陣列的列與欄編號.

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

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 內部設定沒有外邊緣. 須用 `oma` 或 `omi` 設定 **外邊緣 (outer margin)** 的大小, 類似 `mar()` 和 `mai` 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.

```
> # a
> layout(matrix(1:4, 2, 2))
> layout.show(4)
>
> # b
> xlayout(matrix(1:6, 3, 2, byrow=TRUE))
> layout.show(6)
>
> # c
> layout(matrix(c(1,2,3,3), 2, 2, byrow=TRUE))
> layout.show(3)
>
> # d
```

```

> layout(matrix(1:4, 2, 2, byrow=TRUE), widths=c(3,1), heights=c(1,3))
> layout.show(4)
>
> # e
> layout(matrix(c(1,1,2,1), 2, 2), widths=c(2,1), heights=c(1,2))
> layout.show(2)
>
> # f
> layout(matrix(c(0,1,2,3), 2, 2), widths=c(1,3), heights=c(1,3))
> layout.show(3)
>
> # g
> ## create single figure which is 5cm square
> layout(matrix(1), widths=lcm(5), heights=lcm(5))
>
> # h
> setps(RGraphAnatomyLayout08, h=5, w=5)
> # setpdf(RGraphAnatomyLayout08, h=5, w=5)
> par(mfrow = c(1,1))
>
> ##-- Create a scatterplot with marginal histograms -----
> def.par <- par(no.readonly = TRUE) # save default, for resetting...
>
>     x <- pmin(3, pmax(-3, rnorm(50)))
>     y <- pmin(3, pmax(-3, rnorm(50)))
>     xhist <- hist(x, breaks=seq(-3,3,0.5), plot=FALSE)
>     yhist <- hist(y, breaks=seq(-3,3,0.5), plot=FALSE)
>     top <- max(c(xhist$counts, yhist$counts))
>     xrange <- c(-3,3)
>     yrange <- c(-3,3)
>     nf <- layout(matrix(c(2,0,1,3),2,2,byrow=TRUE), c(3,1), c(1,3), TRUE)
>     layout.show(nf)
>
>     par(mar=c(3,3,1,1))
>     plot(x, y, xlim=xrange, ylim=yrange, xlab="", ylab="")
>     par(mar=c(0,3,1,1))
>     barplot(xhist$counts, axes=FALSE, ylim=c(0, top), space=0)
>     par(mar=c(3,0,1,1))
>     barplot(yhist$counts, axes=FALSE, xlim=c(0, top), space=0, horiz=TRUE)

```

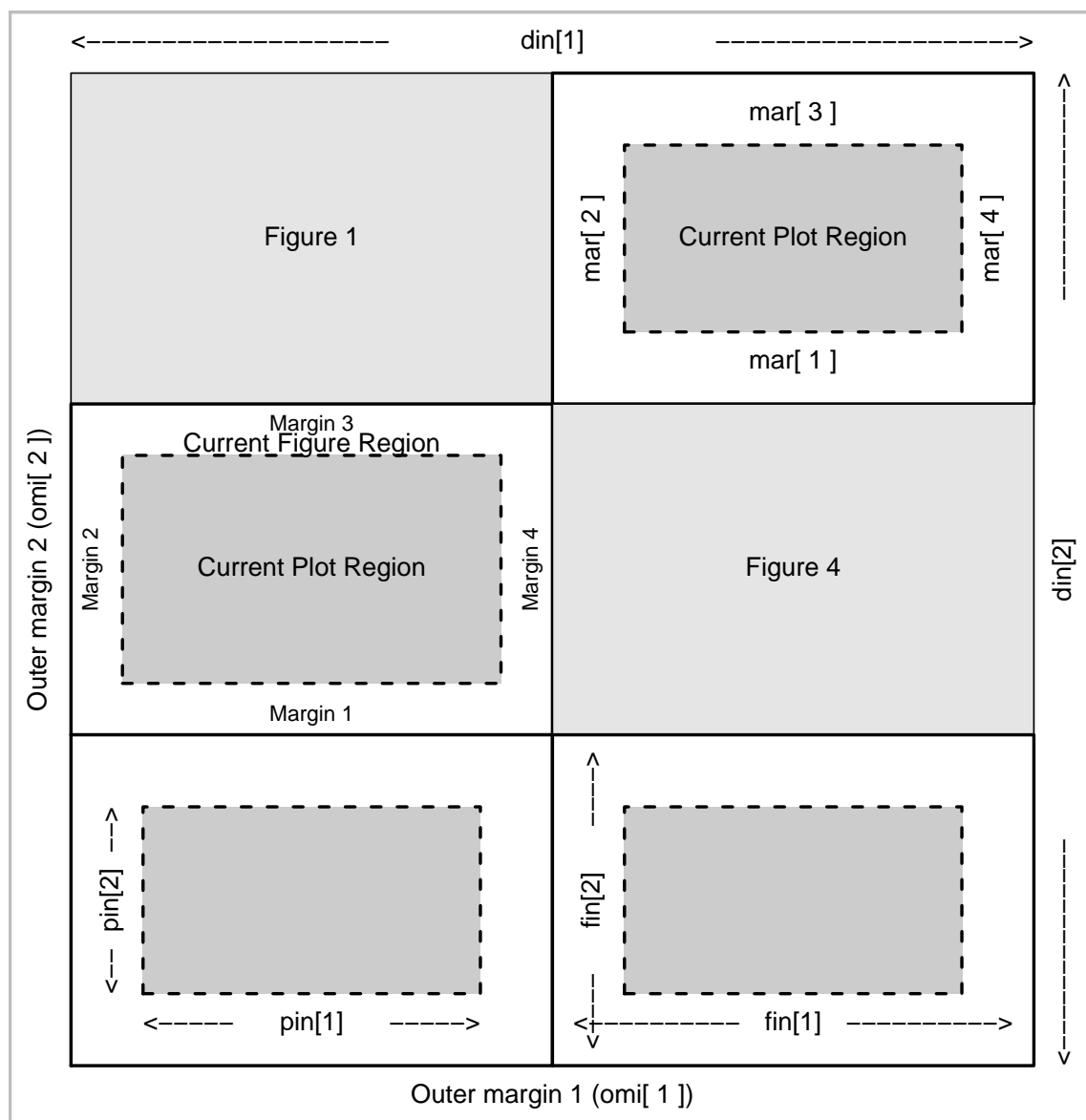



圖 6.17: R 多重圖形與圖畫邊緣解剖

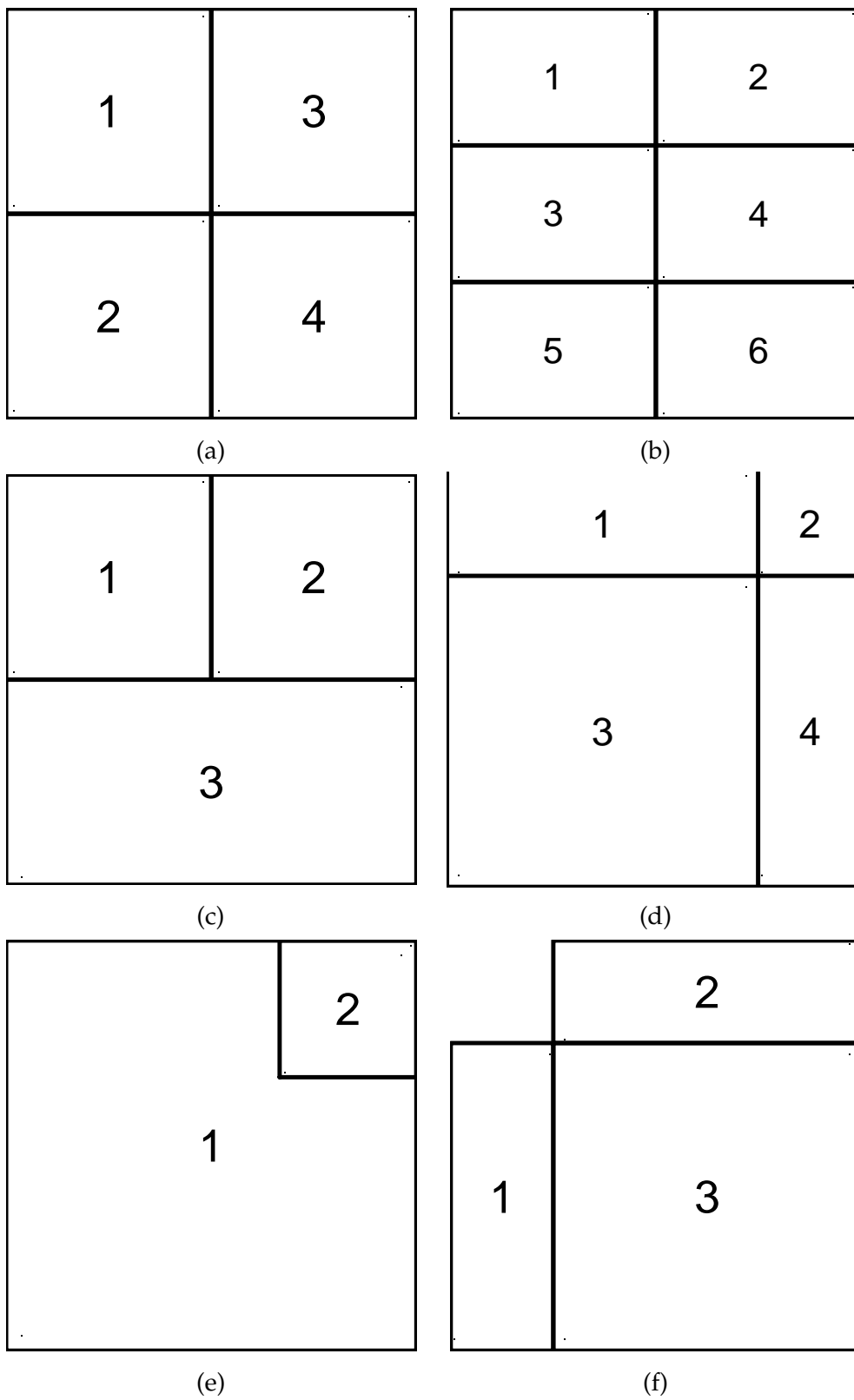


圖 6.18: 顯示多重圖形

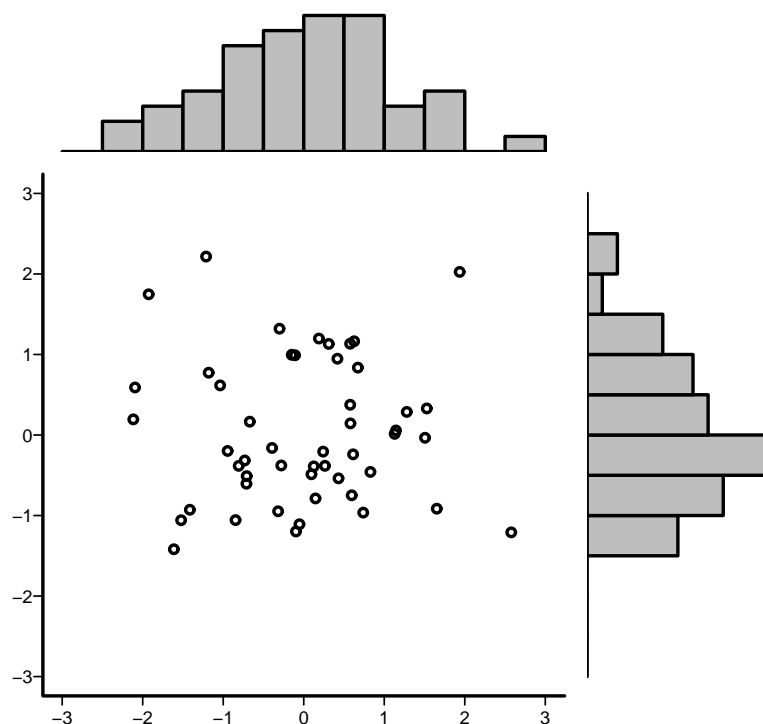


圖 6.19: R 多重圖形與圖畫邊緣使用

6.6 圖形裝置 Graphic Devices

圖形裝置 (Graphic Devices) 是指顯示器和列印設備, R 幾乎可以在任何形式的顯示器和列印設備上繪出圖片。但是在繪出圖片前, R 必須設定一種圖形裝置來處理圖片。透過啟動圖形裝置驅動器 (device driver) 來完成圖形裝置驅動器將 R 的畫圖指令, 如 `plot()`, 轉化成特定圖形裝置可以識別的指令。可以用圖形裝置函式來啟動圖形裝置。例使用指令 `postscript()` 會指令 `postscript()` 之後所有產生的圖形以 PostScript 的形式輸出。當結束使用一個特定圖形裝置時, 用 `dev.off()` 指令終止圖形裝置驅動器。利用 Windows R 的圖形視窗上方表單點選方式, 可以把視窗中的圖片複製到剪貼簿上, 或存為各種格式的圖形檔案, 包括 WMF, PostScript, PNG, BMP, JPEG。每一種圖形裝置都有對應的函數, 參見 `help(Devices)` 輔助文件。

6.7 常用圖形裝置

一些常用的圖形裝置是

`X11()`

用 UNIX 類型的系統的 X11 桌面系統。

`windows()`

用於 Windows 系統。

`quartz()`

用於 MacOS X 系統.

`postscript()`

用於 PostScript 印表機, 或者輸出 PostScript 型式檔案.

`pdf()`

用於 PDF 印表機, 或者輸出 PDF 型式檔案.

`png()`

輸出 png 點陣圖型式檔案.

`jpeg()`

輸出 jpeg 點陣圖型式檔案.

6.7.1 PostScript 與 pdf 圖形裝置

在論文發表時, 常需要高畫質之統計繪圖, 圖形裝置驅動函式

```
postscript(file="filename.ps", ...)  
pdf(file="filename.pdf", ...)
```

透過引數 `file` 可以在是定的檔名中以 PostScript 或 pdf 格式儲存圖片. 圖片以橫行 (landscape) 儲存放置. 引數 `horizontal=FALSE` 可以使得圖片縱向 (portrait) 儲存放置. 引數 `width` 和 `height` 可以設定圖形大小. 例如,

```
postscript("file.ps", horizontal=FALSE, height=3, pointsize=10)  
pdf("file.pdf", horizontal=FALSE, height=3, pointsize=10)
```

產生一個圖片高為 3 英寸, 以 PostScript 或 pdf 格式的圖片且檔案名稱為 `filename.ps` 或 `filename.pdf`, 儲存在工作空間目錄中. 如果指令中的檔案名已經存在, 原來的檔將會被覆蓋. PostScript 輸出的圖片常常用於插入其他文件檔中, 如 L^AT_EX, 此時最好用 encapsulated PostScript 格式圖片.

在 **Hmisc** 套件中, 有 `setps(filename, ...)` 與 `setpdf(filename, ...)` 函式, 其內部自動設定在輸出一般圖片上, 比 `postscript(file= , ...)`, `pdf(file= , ...)` 方便, 但有些時候, 可以比較不同方式, 選擇較適當之輸出函式.